



(22) Date de dépôt/Filing Date: 2009/10/19

(41) Mise à la disp. pub./Open to Public Insp.: 2009/12/23

(45) Date de délivrance/Issue Date: 2011/07/26

(51) Cl.Int./Int.Cl. *G06F 3/048* (2006.01)

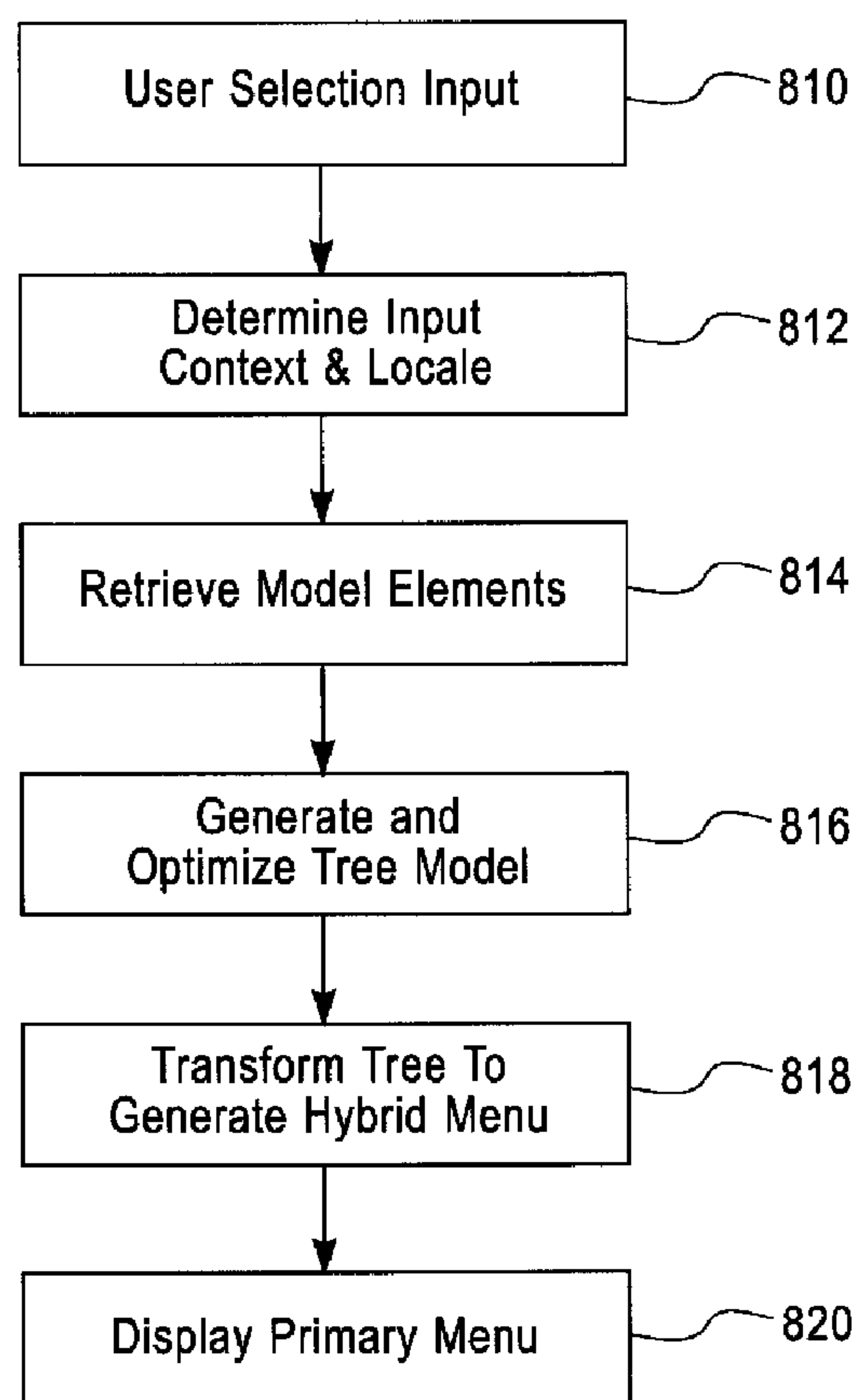
(72) Inventeur/Inventor:
GOBEIL, CHARLES MARC, CA

(73) Propriétaire/Owner:
IBM CANADA LIMITED - IBM CANADA LIMITEE, CA

(74) Agent: WANG, PETER

(54) Titre : SYSTÈME ET METHODE PERMETTANT LA GENERATION ET L'AFFICHAGE DE MENUS A CONTEXTE
HYBRIDE

(54) Title: SYSTEM AND METHOD FOR GENERATING AND DISPLAYING HYBRID CONTEXT MENUS



(57) Abrégé/Abstract:

A system and method for generating and displaying hybrid menus comprising primary and sub-menus having different geometries. Depending upon the context in which a user provides selection input for menu display, the context-appropriate model elements are



(57) **Abrégé(suite)/Abstract(continued):**

retrieved. Based on the model elements and the properties of the model elements, a tree model is generated. A layout generation component transforms the tree model into a hybrid menu for display to the user.

SYSTEM AND METHOD FOR GENERATING AND DISPLAYING HYBRID CONTEXT MENUS

ABSTRACT

A system and method for generating and displaying hybrid menus comprising primary and sub-menus having different geometries. Depending upon the context in which a user provides selection input for menu display, the context-appropriate model elements are retrieved. Based on the model elements and the properties of the model elements, a tree model is generated. A layout generation component transforms the tree model into a hybrid menu for display to the user.

SYSTEM AND METHOD FOR GENERATING AND DISPLAYING HYBRID CONTEXT MENUS

5 FIELD OF THE INVENTION

[001] The present invention relates generally to graphical user interfaces for computing devices, and relates more particularly to a method for providing dynamic generation and display of a hierarchical hybrid context menu at a graphical user interface.

BACKGROUND

10 [002] Computer applications having graphical user interfaces (GUIs) generally provide menus of selectable application features. A standard in the field is to display a drop down list of selectable menu elements, either upon user activation (e.g., a keystroke or a right mouse click) or by user inaction (i.e., after a fixed period of mouse inactivity) when the cursor or pointer hovers over a designated area of the display screen. Figure 1A
15 illustrates a context menu which appears upon right clicking anywhere on the user display when running a Graphical Modeling Framework ("GMF") tool. Traditional menu systems arrange the list of selections in a top-to-bottom linear configuration for user selection. If there is a large number of menu items, the lists will be broken into a primary list, often comprising the most frequently used menu items, and a plurality of sub-lists which are
20 selectively displayed upon user selection of a menu item in the primary list. Display of a primary list with one expanded sub-list or sub-menu is shown for the context menu for GMF in Fig. 1B. The sub-lists may be less frequently-used items or may be sub-menus with respect to particular headings found on a primary list. Upon use selection of a particular heading on the primary list, the subsequent menu list (also referred to as a sub-
25 list or a child menu) is displayed for user selection. Selection of the heading item from the primary list may be so-called "passive selection" accomplished by hovering the cursor/pointer over the heading or active user input (e.g., by mouse click or keystroke).

[003] The use of linear lists with sub-lists has some drawbacks. Frequently, the sub-list will be automatically displayed only for so long as the user maintains the exact cursor, or pointer, position with respect to the expanded menu item from the primary list or an item on the sub-list. In addition, the user needs to carefully move the cursor laterally over
5 to the sub-list/child menu list along the shared line or risk the disappearance of the list due to a newly-detected cursor position. Because of the long narrow path users must carefully follow to avoid causing the child menu to disappear, the prior art approach to display of linear linked lists performs poorly in view of the Accot-Zhai steering law, which states that the time to follow a tunnel without touching the edges is proportional to the ratio of
10 length-to-width of the tunnel. Also,, the menu items on linear lists and sub-lists often appear at unpredictable GUI locations in an attempt to display the selectable items away from screen edges and close to the current cursor location (e.g., lists above and below selected headings as shown in Figs. 1B and 1C respectively, or lists displayed on the left or right of selected headings (not illustrated). The unpredictable locations put a limit on
15 the user's ability to learn the interface by muscle memory and violate the user's expectation of consistency. It is preferable that the location for display should be fixed relative to a diagram element.

[004] Pie menus, such as the one shown in Fig. 2, have been used as a method for quick object selection. In a pie menu, a user's cursor located at the center of the pie
20 display is distance invariant (i.e., the distance to select any one menu item is the same). Pie menus typically reduce selection time. Pie slices entered on the initiation position of a mouse pointer (a.k.a., cursor) optimize those parameters, making it more efficient for users to select items than from a linear context menu. The pros and cons of pie menus over traditional linear menus have been detailed in a publication by Hopkins, P., entitled
25 "Direction Selection is Easy as Pie Menus" from Computer Graphics Workshop (Oct. 10,

1997). According to Fitts' law (discussed further below), the time required to rapidly move to a target area is a function of the distance to the target and the size of the target, making pie menus easier to navigate. U.S. Patent 6,549,219 of Selker, assigned to the present assignee, discloses a pie menu graphical user interface having concentrically-
5 arranged pie menu levels. The Selker patent positions more frequently used items, items 20-22 in Fig. 2, toward the center of the pie menu and outer menu items 23-30, representing less frequently used items that may or may not bear any relationship to the more centrally-located items, further from the center.

[005] Radial menus, which are variations on traditional pie menus, also have
10 advantages and disadvantages. It is preferable to use icons rather than text on radial pie segments in order to minimize the size of the displayed menu. The radial menu depends on having unique icons and a 1:1 aspect ratio for every item, since wide text blocks will skew the pie's geometry. In addition, the number of segments in a radial menu is effectively limited by the requirement that the display be both readable and navigable.
15 Expansion of pie segments to show sub-menus has been demonstrated in U.S. Patent 7,509,348 of Burtner, et al. The Burtner patent provides adjacent radial submenus which can be displayed based on user selection of one of the central pie segments. As shown in Fig. 3, a radial menu including eight primary segments, 31-38, is displayed to a user. When the user hovers over or actively selects one of the segments, 32, a sub-menu is
20 displayed including sub-segments 32.1, 32.2 and 32.3. However, when the adjacent radial sub-menu is displayed, user navigation of items displayed in the radial sub-menu is inhibited by the same principle (the aforementioned Accot-Zhai steering law) that limits navigation for linear sub-menus, namely that the width of the area to be navigated is narrow relative to the distance to be traveled. Moreover, user control of a pointer along an
25 arced path is more difficult than user control along the linear paths.

[006] Thus, there is a need in the art for a method and apparatus for automatically generating hierarchical menus with sub-menus of user-selectable entries that are easier and faster to use, that scale reasonably to large numbers of entries, and that use minimal display area.

5 SUMMARY OF THE INVENTION

[007] The present invention provides a system and method for generating and displaying hybrid menus comprising primary menus and sub-menus having different geometries. Depending upon the context in which a user provides selection input for menu display, the context-appropriate model elements are retrieved. Based on the model
10 elements and the properties of the model elements, a tree model is dynamically generated and optimized for display as a hybrid menu.

BRIEF DESCRIPTION OF THE DRAWINGS

[008] So that the manner in which the above recited embodiments of the invention are attained and can be understood in detail, a more particular description of the invention,
15 briefly summarized above, may be obtained by reference to the embodiments thereof which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of this invention and are therefore not to be considered limiting of its scope, for the invention may admit to other equally effective embodiments.

20 Figs. 1A-1C illustrate graphical user interfaces displaying linear menus as are well known in the art;

Fig. 2 illustrates a prior art pie menu for a graphical user interface;

Fig. 3 illustrates a radial menu with an expanded sub-menu;

Figs. 4A and 4B illustrate Fitts' law with reference to radial and linear menus;

Fig. 5 is a diagram illustrating the hybrid menu generated in accordance with the present invention;

Fig. 6 comprises Figs. 6A and 6B to illustrate a sample uncompressed menu model
5 and linear arrangement, respectively;

Figs. 7A and 7B illustrate the model of Fig. 6 optimized for a radial menu and for a linear menu, respectively;

Fig. 8 is a process flow for automatically generating a hybrid menu in accordance with the present invention;

10 Fig. 9 illustrates the logical flow for transforming a tree menu into a hybrid menu in accordance with the present invention; and

Fig. 10 illustrates a system for implementing the present invention.

DETAILED DESCRIPTION

[009] In accordance with the present invention, the advantages of radial and linear
15 menus are realized by a system and method for automatically and dynamically generating hybrid menus and sub-menus for display to a user.

[010] Fitts' law and the law of steering can be used to predict user performance without performing user trials. Given a radial menu having six (6) segments as shown in Fig. 4A and a linear menu having the same number of entries, it can be seen that user
20 navigation is greatly facilitated by the radial layout. Fitts' law is shown in Equation 1.

$$T = a + b \int \log_2 \left(\frac{D}{W} + 1 \right) \quad (\text{Equation 1})$$

[011] Where T is the average time taken to complete the movement; a represents the start/stop time of the device and b stands for the inherent speed of the device. These constants can be determined experimentally by fitting a straight line to measured data. D is the distance from the starting point to the center of the target, and W is the width of the target measured along the axis of motion. W can also be thought of as the allowed error tolerance in the final position, since the final point of motion must fall within $\pm W/2$ of the target's center.

[012] From the equation, it can be seen that there is a speed-accuracy trade-off associated with pointing, whereby targets that are smaller and/or further away require more time to acquire. Figs. 4A and 4B show how these parameters can be extracted from the radial and linear menus. The figures show the best case parameterization for the linear menu by choosing the closer edge, and one of the many equivalent parameterizations of the radial menu considering that the D/W ratio would stay approximately the same for clicks anywhere close to the center. As illustrated by Figs. 4A and 4B, in the radial menu, the D/W ratio for any particular menu item is a constant for a given number of total items in the menu; whereas, in the linear menu, the D/W ratio increases for items further down the list.

[013] The system and method for generating and displaying hybrid menus generates primary menus and sub-menus having similar or different geometries. As in the prior art, a context-appropriate hierarchy of model elements is retrieved as a function of the context in which a user provides selection input. Departing from the prior art, in which model elements were displayed using fixed display geometry, however, the inventive system

queries the model for what type of view shall be used to display it. The model is then transformed to optimize user interaction performance characteristics associated with the views, while observing constraints to preserve semantic structure of the input model. Views of similar and differing geometries are bound together using a common geometric
5 coupling constraint to form a hierarchical menu for display to the user. A layout generator then arranges items within each menu to optimize access time, while observing constraints that preserve the user's ability to build muscle memory of paths to item locations. For the special case of radially-arranged menus, it keeps item positions consistent between multiple invocation contexts using a preferred angle property on model elements, and a
10 combinatorial optimization algorithm to minimize departure from specified angles when conflict occurs. The hybrid menu is then bound with interaction logic to form a fully functional command interface. Further, the hybrid menu is provided with the logic for detecting and responding to events affecting the layout of items in the hybrid menu, thereby facilitating automatic updating.

15 [014] Fig. 5 is a diagram representatively illustrating a hybrid menu in accordance with the present invention. The hybrid menu addresses the disadvantages of the prior art radial menu by dynamically expanding radial pie segments to display modified linear sub-menus having labels. A hybrid menu may comprise any combination of radial and linear menus, as well as other geometries supported by the coupling constraint (further detailed
20 below). For ease of description, and with reference to Fig. 5, the primary menu will refer to a radial menu 500 having a linear sub-menu 502 which, in turn, has a radial sub-menu 544. Upon user selection of a radial segment, for example segment 501, the modified linear sub-menu, "Associations" sub-menu 502, is displayed. Where the prior art linear context menus formed long narrow tunnels to travel from menu to sub-menu, the hybrid
25 menu includes paths from radial menu items in segments to linear sub-menus which are

short and wide and easier to navigate. The modified linear sub-menus are positioned as close to their centers (i.e., their radial segment) as possible to cut the average distance to the target menu items. Since each sub-menu will expand in relation to a shared tangential line with its parent pie segment, user expectation is satisfied that the sub-menu for the
5 selected radial pie segment will always be displayed in the same orientation to the primary radial display and that a desired sub-menu item, for example "Create Composition To..." at 542 will always be found in its currently-displayed location relative to the radial pie display. Moreover, as further detailed below, menu items in displays which are dynamically generated for different contexts will preferably be displayed in the same
10 location relative to their primary radial displays, again to facilitate user memorization and ease of use. Upon user selection of sub-menu item 542, a sub-sub-menu 544 is displayed.

[015] In order to generate a hybrid menu, model elements are first retrieved for the context and a tree structure, or other hierarchical model representation, is generated. Fig. 6A illustrates a sample uncompressed menu model showing the relationships between
15 menu and sub-menu items. Fig. 6B illustrates generally how those items would be configured for linear display. Given the simple model illustrated, leaf items are labeled "L" and groups are labeled "G". Fig. 6A provides a tree model of the menu items and Fig. 6B illustrates how those items would be displayed using a standard linear menu. The present invention seeks to automatically optimize the menu for fast access, optimal use of
20 screen real estate, discoverability of items and any number of other factors that depend on how the user interacts with the visual representation of the menu model. The hybrid menu system of the present invention optimizes the tree model using model element properties a further detailed below.

[016] Figs. 7A and 7B illustrate the model of Fig. 6A optimized for a radial menu and for a linear menu, respectively. Since radial menus preferably have no more than 10 segments, and ideally have between 5 and 8 segments, the model of Fig. 6A is preferably evaluated to have 5-8 menu items in the primary radial display. Radial menus are ideal for handling sub-menus, providing a quick, constant distance, ease of memorization and a widened path for mouse movement toward sub-menu items. If the menu has fewer than the ideal number of items, for example 6 items, selected groups can be exploded. The “explosion” of a submenu moves all of that sub-menu’s items into the parent menu, surrounds them with separators, and removes the sub-menu. If there are too many items, all of the leaf items can be collected into a group. As shown in Fig. 7A, group G4 has been automatically exploded since it has only one child; G3 has been eliminated since it had no children, and G1 and G2 have been left intact, since exploding either or both of them would push the root’s child count past the ideal. In contrast, however, G1.1 has been exploded into G1, since it brings the item segment number closer to the ideal.

[017] Fig. 7B shows the same model from Fig. 6A optimized for a linear menu. With linear menus, a higher item limit can be set for the primary display and sub-menus are usually exploded into the top level menu with separators. Fig. 7B shows the optimization of the Fig. 6A model wherein G4 has been automatically exploded since it had only one child item, G3 was eliminated for having no child items, and both G1 and G2 have been exploded into root due to the linear menu’s higher fan-out limit. As illustrated, the fan-out limit was set at 9. Had the limit been higher, G1.1 would also have been exploded to the higher layer.

[018] Under the present invention, a new model of model elements selected for the context is created each time a user provides selection input in order to generate a hybrid

display. Once a tree or other hierarchical model is created, it is then transformed to generate the hybrid menu display. In evaluating menu items for generating the model, ordering of items is preferably undertaken. For example, ordering rules can be effected for ordering based on criteria such as historical usage, location consistency, or other factors. To optimize discovery and access times, for example, the most frequently used items should appear in a consistent position in primary radial menus or, if relegated to the linear sub-menus, should appear first in the linear sub-menus. Ideally items are ordered by the percentage of time that the items are selected by a user within a given task context. The percentage of time of user access may be predefined for a given context or may be dynamically tracked for a particular user. Tables of values representing the ordering criteria for each menu item preferably resides in one place where the menu items and associated values can be dynamically updated in relation to each other. For example, a user may enable different capabilities of an application which would cause an update to the ordering. The ordering under the present invention preferably orders groups and items within groups with the priority of groups being derived from the priority of the items within them.

[019] Menu items are represented as model elements which are stored, for example in the aforementioned tables. When user selection input is received for a location and context, the relevant model elements are retrieved. Model elements would be, for example, the features that are available to the user relative to the context (e.g., program) and location (e.g., a displayed object) at which the user selection input was detected. Each model element is stored with so-called dependency, or family, properties as well as display properties related to its ultimate display. The dependency or family properties are used for generating the hierarchical model. Model element display properties are used for transforming the hierarchical model into the hybrid display and include, but are not limited

to, a preferred angle for display, a priority weight factor, an explicit group flag and a display menu preference. The preferred angle for display refers to a consistent location relative to a primary display or sub-menu item. As noted above, it is preferable that menu items be displayed consistently regardless of the context for which the menu items are being displayed. A preferred angle property tells where that menu item is preferably situated in a display. The priority weight factor property tells the importance of the model element. In dynamically generating hybrid menus, the higher priority menu items are preferably displayed in the primary radial display. The priority weight factory property is also used to order elements in linear sub-menus where position determines access times.

The explicit group flag is used to flag an explicit group of menu items that should not be exploded during layout. Finally, the display menu preference property tells how the model element should be displayed. The display menu preference property may indicate “radial”, “linear” or any other geometry, or it may indicate “auto”. With an “auto” display menu preference property, the hybrid display system will automatically determine a preferred display menu for displaying the model element.

[020] Fig. 8 is a process flow for automatically generating a hybrid menu in accordance with the present invention. The process is to dynamically generate context-dependent hybrid menus. The hybrid menus may comprise any combination of radial and linear menus, as well as other geometries supported by the coupling constraint, further detailed below. For ease of description, as above with reference to Fig. 5, the primary menu will refer to a radial menu having a linear sub-menu which, in turn, has a radial sub-menu. The process of Fig. 8 starts with user selection input at 810. The user selection input may, for example, be a right click on a screen area, inactivity in a screen area at which a cursor/pointer is displayed, touch screen input, keystroke input, etc. The system determines where the input was received and identifies the context at step 812. Next,

context-dependent model elements are retrieved given the context and area of display at 814. Once the model elements have been retrieved, a tree model, or other hierarchical model, is generated and optimized taking into account the model element properties of display menu preference, explicit group flag and priority at 816. The tree model is then transformed to generate the hybrid menu, at step 818. Transforming the optimized tree model comprises creating radial and linear menus from the optimized model, consistent with the display menu preference (i.e., radial/linear view type) followed by the layout process of arranging the items within each menu, taking into account the priority and preferred angle model element display properties. Each of the menu items is associated with interaction functionality. The generated hybrid menu includes the primary and all sub-menus. Initially only the primary menu is displayed at step 820. Thereafter, if the user selects one of the primary menu segments, the interaction will cause a sub-menu to be displayed.

[021] Further detail of the transformation of the optimized tree menu into a hybrid menu are depicted in Figure 9. A Menu Controller 948 creates radial and linear menus from the optimized tree model to generate a menu figure 980. The figure controller creates instances of the Menu 980 and its child menus, a Layout Input Binding 940 which listens for event data, such as user interaction input or Menu Controller input that may affect the menu layout, a Layout Generator algorithm 950 which receives the input from the Layout Input Binding and executes to generate the layout of the hybrid menu and a Layout Output Binding 960 which applies the changes to the menu figure 980. The Menu Controller then “wires everything up” by adding the children to the main Menu figure 980, setting the Layout Input Binding to listen for events on the main Menu figure, setting the main Menu figure’s input binding to the one it created, setting the input binding’s layout

algorithm to the one it created, setting the Layout Algorithm's output binding to the one it created, and setting the Layout Output Binding's target to the main Menu figure.

[022] When an event that changes layout occurs in the main figure (e.g. child added), the Layout Input Binding assesses whether the event affects parameters that the Layout
 5 Algorithm is interested in. If the event does affect parameters the layout is interested in, the Layout Input Binding instructs the Layout Algorithm to run. The Layout Algorithm pulls parameters of interest from the figure through the Layout Input Binding (which may provide some translation). During execution, the layout algorithm sends results to the Layout Output Binding, which applies them to the main Menu figure and its children
 10 (possibly with some translation) by either applying the results/changes immediately or buffering the results and applying all changes at once when complete. The Layout Output Binding may also buffer the results and pass them to an optional Animation Controller (not shown) which will apply the changes over time.

[023] In ordering the menu items for generating the tree model, the system optimally
 15 selects a number of primary menu items that is fewer than, for example, 8 items when the model elements have a radial menu property indicated. If the identified model elements are fewer than 8, for example 6 primary model elements, the generated radial menu layout may include 8 segments, wherein 6 have icons and/or labels and 2 are non-selectable blank segments. Alternatively, the display will show only 6 segments. Dynamic model
 20 adjustment may be required if, for example, more than 8 model elements have similar priority weight properties and have radial display properties. If the identified primary menu items exceed 8 in number, another iteration of menu items ordering (i.e., tree model optimization) is done, possibly including re-grouping of menu items, to achieve the goal of 8 or fewer primary menu item model elements, taking into account the explicit group flag

property of the associated model element. In the instance where fewer than 5 model elements are identified for radial display in a primary menu, the dynamic model adjustment can determine if any identified sub-menu items have one child, in which case the sub-menu item(s) may be elevated to a primary menu display item. It will be understood that 5-8 segments have been chosen as optimal for display on an average user display device, including computer monitor, PDA, phone, etc. but that a larger or smaller number of segments can be defined as optimal. It is also contemplated that the hybrid menu display system can ascertain the user output display device, automatically or via user prompting, and dynamically tailor the primary and sub-menu displays for the current user output display device.

[024] After the primary menu items have been determined, the linear sub-menu items are ordered for each primary menu item and the layout of the sub-menus is determined. As noted above with reference to Fig. 5, when a sub-menu is being displayed relative to a radial menu, the sub-menu display is optimally displayed in tangential relationship to its associated primary segment using the preferred angle display property. Additional identified menu items may fall into sub-menus (referred to as sub-sub-menus) of a given sub-menu. Depending upon the identified relationships among the menu items, the layout generation component will also automatically generate menu navigation features for display, including such artifacts as arrows, accelerator keys underlines, highlighting, etc. Display of the hybrid menu may include aspects that further facilitate user review and selection. Given the limited segment real estate, icons are preferably displayed in a primary radial menu. Clicking on or hovering the cursor/pointer at the empty center of the radial menu for a predetermined elapsed time can cause display of the labels for all of the segments, ideally displayed outside of the periphery of the radial menu for maximum label word lengths. Clicking on or hovering over a single segment for a predetermined period

will cause display of the label for the single segment's menu item. For radial menus, pressing the space bar could cause all label callouts to protrude from the menu at once, complete with accelerator key underlines. User selection could alternatively be done using the space bar, enter key or other inputs, as will be understood by one having skill in the art. Arrows are preferably provided to cycle through elements in the menu, and to expand or collapse sub-menus. For example, for linear menu navigation, the up and down arrows go to next/prev, while the left and right arrows go to expand/collapse. For radial menus, the system checks the angle of the currently selected item to see if it's in a top, left, right, or bottom quadrant. If in the top, left and right arrows do next/prev, an up arrow does expansion, and a down arrow is for collapse. If in bottom, left and right arrows do prev/next, an up arrow collapses and a down arrow expands. If to the right, up and down arrows do next/prev, a left arrow collapses and a right arrow expands, etc. Further, if the last navigation was a prev/next, and occurred within a short time of the current and identical key press (a repeat of the last action), then the last action will be repeated regardless of the current quadrant. The ellipsis queues would preferably remain present in the label callout, to indicate whether an item is a direct action or a link to cancellable dialog. The foregoing represents logic that has been implemented for user convenience, but clearly can be modified without departing from the spirit and scope of the present invention.

20 [025] Fig. 10 illustrates a representative computer system for implementing the hybrid menu display system of the present invention. The hybrid menu display system may be located at a user device or remote from the user device (e.g., at a LAN server or at a remote server in a cloud computing environment). In FIG. 10, the exemplary processing system 1000 includes a processing device 1010 for executing software components, including the Model Generator Component 1040, Model Optimizer Component 1044,

25

Menu Controller 1048 and Layout Generator Component 1050. The system further has remote (not shown) or local memory 1020 coupled to the processor, as well as input/output (I/O) circuitry 1030 operative to interface with the processor and with a user graphical display device. The processor 1010 may be configured to load and execute

5 features of the Model Generator Component 1040, Model Optimizer Component 1044, Menu Controller 1048 and Layout Generator Component 1050 to perform at least a portion of the methodologies of the present invention, illustrative embodiments of which are described herein. Display 1012 may be provided at the processing system 1000 for display to the user. Alternatively, the processing system 1000 will send the display

10 information to a remotely-located graphical user interface 1016 at a user location 1014.

[026] The hybrid menu is dynamic and scalable to accommodate new application features, etc. without requiring complete overhaul of the menu structure and its attendant disruption to user expectations. Moreover, a hybrid menu can automatically update itself

The current hybrid menu implementation is already initialized from a simple interface

15 based menu model and can be extended to listen to that model for changes to the menu structure. Menu elements are arranged by a loosely-coupled and configurable layout algorithm that can handle a wide array of configurations. Ideally, a layer is provided in between existing menu contributions and the menu model that will intelligently optimize groupings to optimize hybrid menus. By taking into account menu groups, sub-menus, the

20 presence of icons, the number of children in a group, and any other appropriate metrics, the menu model will resolve any conflicts between linear and hybrid menu groupings to automatically reconfigure existing menus for display as hybrid menus.

[027] The methodologies of embodiments of the invention may be particularly well-suited for use in an electronic device or alternative system. Accordingly, the present

invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “circuit,” “module” or “system.” Furthermore, the present invention may take the form of
5 a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

[028] Any combination of one or more computer usable or computer readable medium(s) may be utilized. The computer-usable or computer-readable medium may be a computer readable signal medium or a computer readable storage medium. A computer
10 readable storage medium may be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer-readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random
15 access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain or store a program for use
20 by or in connection with an instruction execution system, apparatus or device.

[029] A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as a part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electromagnetic, optical, or any suitable combination thereof. A
25 computer readable signal medium may be any computer readable medium that is not a

computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

[030] Program code embodied on a computer readable medium may be transmitted
5 using any appropriate medium, including but not limited to wireless, wire-line, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

[031] Computer program code for carrying out operations of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and
10 conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's
15 computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

[032] The present invention is described above with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program
20 products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions.

[033] These computer program instructions may be stored in a computer-readable
25 medium that can direct a computer or other programmable data processing apparatus to

function in a particular manner, such that the instructions stored in the computer-readable medium produce an article of manufacture including instruction means which implement the function/act specified in the flowchart and/or block diagram block or blocks.

[034] The computer program instructions may be stored in a computer readable
5 medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks.

10 [035] For example, FIG. 9 is a block diagram depicting an exemplary processing system 900 formed in accordance with an aspect of the invention. System 900 may include a processor 910, memory 920 coupled to the processor (e.g., via a bus 940 or alternative connection means), as well as input/output (I/O) circuitry 930 operative to interface with the processor. The processor 910 may be configured to perform at least a
15 portion of the methodologies of the present invention, illustrative embodiments of which are shown in the above figures and described herein.

[036] It is to be appreciated that the term “processor” as used herein is intended to include any processing device, such as, for example, one that includes a central processing unit (CPU) and/or other processing circuitry (e.g., digital signal processor (DSP),
20 microprocessor, etc.). Additionally, it is to be understood that the term “processor” may refer to more than one processing device, and that various elements associated with a processing device may be shared by other processing devices. The term “memory” as used herein is intended to include memory and other computer-readable media associated with a processor or CPU, such as, for example, random access memory (RAM), read only
25 memory (ROM), fixed storage media (e.g., a hard drive), removable storage media (e.g., a

diskette), flash memory, etc. Furthermore, the term “I/O circuitry” as used herein is intended to include, for example, one or more input devices (e.g., keyboard, mouse, etc.) for entering data to the processor, and/or one or more output devices (e.g., printer, monitor, etc.) for presenting the results associated with the processor.

5 [037] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for
10 implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each
15 block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[038] Implementation details may modified without departing from the spirit and
20 scope of the present invention. For example, the above description describes implementation details for radial menus, including pressing the alt key to cause all label callouts to protrude from the menu at once, complete with accelerator key underlines, as well as left and right arrows to cycle through elements in the menu, the up arrow used to expand sub-menus, and the down arrow to collapse them. Clearly display alternatives may
25 be implemented yet still be covered by the present invention.

[039] While foregoing is directed to the preferred embodiment of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

SYSTEM AND METHOD FOR GENERATING AND DISPLAYING HYBRID CONTEXT MENUS

FIELD OF THE INVENTION

The present invention relates generally to graphical user interfaces for computing devices, and relates more particularly to a method for providing dynamic generation and display of a hierarchical hybrid context menu at a graphical user interface.

BACKGROUND

Computer applications having graphical user interfaces (GUIs) generally provide menus of selectable application features. A standard in the field is to display a drop down list of selectable menu elements, either upon user activation (e.g., a keystroke or a right mouse click) or by user inaction (i.e., after a fixed period of mouse inactivity) when the cursor or pointer hovers over a designated area of the display screen. Figure 1A illustrates a context menu which appears upon right clicking anywhere on the user display when running a Graphical Modeling Framework (“GMF”) tool. Traditional menu systems arrange the list of selections in a top-to-bottom linear configuration for user selection. If there is a large number of menu items, the lists will be broken into a primary list, often comprising the most frequently used menu items, and a plurality of sub-lists which are selectively displayed upon user selection of a menu item in the primary list. Display of a primary list with one expanded sub-list or sub-menu is shown for the context menu for GMF in Fig. 1B. The sub-lists may be less frequently-used items or may be sub-menus with respect to particular headings found on a primary list. Upon use selection of a particular heading on the primary list, the subsequent menu list (also referred to as a sub-list or a child menu) is displayed for user selection. Selection of the heading item from the

primary list may be so-called “passive selection” accomplished by hovering the cursor/pointer over the heading or active user input (e.g., by mouse click or keystroke).

The use of linear lists with sub-lists has some drawbacks. Frequently, the sub-list will be automatically displayed only for so long as the user maintains the exact cursor, or pointer, position with respect to the expanded menu item from the primary list or an item on the sub-list. In addition, the user needs to carefully move the cursor laterally over to the sub-list/child menu list along the shared line or risk the disappearance of the list due to a newly-detected cursor position. Because of the long narrow path users must carefully follow to avoid causing the child menu to disappear, the prior art approach to display of linear linked lists performs poorly in view of the Accot-Zhai steering law, which states that the time to follow a tunnel without touching the edges is proportional to the ratio of length-to-width of the tunnel. Also,, the menu items on linear lists and sub-lists often appear at unpredictable GUI locations in an attempt to display the selectable items away from screen edges and close to the current cursor location (e.g., lists above and below selected headings as shown in Figs. 1B and 1C respectively, or lists displayed on the left or right of selected headings (not illustrated). The unpredictable locations put a limit on the user’s ability to learn the interface by muscle memory and violate the user’s expectation of consistency. It is preferable that the location for display should be fixed relative to a diagram element.

Pie menus, such as the one shown in Fig. 2, have been used as a method for quick object selection. In a pie menu, a user’s cursor located at the center of the pie display is distance invariant (i.e., the distance to select any one menu item is the same). Pie menus typically reduce selection time. Pie slices entered on the initiation position of a mouse pointer (a.k.a., cursor) optimize those parameters, making it more efficient for users to

select items than from a linear context menu. The pros and cons of pie menus over traditional linear menus have been detailed in a publication by Hopkins, P., entitled "Direction Selection is Easy as Pie Menus" from Computer Graphics Workshop (Oct. 10, 1997). According to Fitts' law (discussed further below), the time required to rapidly move to a target area is a function of the distance to the target and the size of the target, making pie menus easier to navigate. U.S. Patent 6,549,219 of Selker, assigned to the present assignee, discloses a pie menu graphical user interface having concentrically-arranged pie menu levels. The Selker patent, the contents of which are incorporated by reference herein, positions more frequently used items, items 20-22 in Fig. 2, toward the center of the pie menu and outer menu items 23-30, representing less frequently used items that may or may not bear any relationship to the more centrally-located items, further from the center.

Radial menus, which are variations on traditional pie menus, also have advantages and disadvantages. It is preferable to use icons rather than text on radial pie segments in order to minimize the size of the displayed menu. The radial menu depends on having unique icons and a 1:1 aspect ratio for every item, since wide text blocks will skew the pie's geometry. In addition, the number of segments in a radial menu is effectively limited by the requirement that the display be both readable and navigable. Expansion of pie segments to show sub-menus has been demonstrated in U.S. Patent 7,509,348 of Burtner, et al. The Burtner patent provides adjacent radial submenus which can be displayed based on user selection of one of the central pie segments. As shown in Fig. 3, a radial menu including eight primary segments, 31-38, is displayed to a user. When the user hovers over or actively selects one of the segments, 32, a sub-menu is displayed including sub-segments 32.1, 32.2 and 32.3. However, when the adjacent radial sub-menu is displayed, user navigation of items displayed in the radial sub-menu is inhibited by the same

principle (the aforementioned Accot-Zhai steering law) that limits navigation for linear sub-menus, namely that the width of the area to be navigated is narrow relative to the distance to be traveled. Moreover, user control of a pointer along an arced path is more difficult than user control along the linear paths.

Thus, there is a need in the art for a method and apparatus for automatically generating hierarchical menus with sub-menus of user-selectable entries that are easier and faster to use, that scale reasonably to large numbers of entries, and that use minimal display area.

SUMMARY OF THE INVENTION

The present invention provides a system and method for generating and displaying hybrid menus comprising primary menus and sub-menus having different geometries. Depending upon the context in which a user provides selection input for menu display, the context-appropriate model elements are retrieved. Based on the model elements and the properties of the model elements, a tree model is dynamically generated and optimized for display as a hybrid menu..

BRIEF DESCRIPTION OF THE DRAWINGS

So that the manner in which the above recited embodiments of the invention are attained and can be understood in detail, a more particular description of the invention, briefly summarized above, may be obtained by reference to the embodiments thereof which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of this invention and are therefore not to be considered limiting of its scope, for the invention may admit to other equally effective embodiments.

Figs. 1A-1C illustrate graphical user interfaces displaying linear menus as are well known in the art;

Fig. 2 illustrates a prior art pie menu for a graphical user interface;

Fig. 3 illustrates a radial menu with an expanded sub-menu;

Figs. 4A and 4B illustrate Fitts' law with reference to radial and linear menus;

Fig. 5 is a diagram illustrating the hybrid menu generated in accordance with the present invention;

Fig. 6 comprises Figs. 6A and 6B to illustrate a sample uncompressed menu model and linear arrangement, respectively;

Figs. 7A and 7B illustrate the model of Fig. 6 optimized for a radial menu and for a linear menu, respectively;

Fig. 8 is a process flow for automatically generating a hybrid menu in accordance with the present invention;

Fig. 9 illustrates the logical flow for transforming a tree menu into a hybrid menu in accordance with the present invention; and

Fig. 10 illustrates a system for implementing the present invention.

DETAILED DESCRIPTION

In accordance with the present invention, the advantages of radial and linear menus are realized by a system and method for automatically and dynamically generating hybrid menus and sub-menus for display to a user.

Fitts' law and the law of steering can be used to predict user performance without performing user trials. Given a radial menu having six (6) segments as shown in Fig. 4A and a linear menu having the same number of entries, it can be seen that user navigation is greatly facilitated by the radial layout. Fitts' law is shown in Equation 1.

$$T = a + b \int \log_2 \left(\frac{D}{W} + 1 \right) \quad (\text{Equation 1})$$

Where T is the average time taken to complete the movement; a represents the start/stop time of the device and b stands for the inherent speed of the device. These constants can be determined experimentally by fitting a straight line to measured data. D is the distance from the starting point to the center of the target, and W is the width of the target measured along the axis of motion. W can also be thought of as the allowed error tolerance in the final position, since the final point of motion must fall within $\pm W/2$ of the target's center.

From the equation, it can be seen that there is a speed-accuracy trade-off associated with pointing, whereby targets that are smaller and/or further away require more time to acquire. Figs. 4A and 4B show how these parameters can be extracted from the radial and linear menus. The figures show the best case parameterization for the linear menu by choosing the closer edge, and one of the many equivalent parameterizations of the radial menu considering that the D/W ratio would stay approximately the same for clicks anywhere close to the center. As illustrated by Figs. 4A and 4B, in the radial menu, the D/W ratio for any particular menu item is a constant for a given number of total items in the menu; whereas, in the linear menu, the D/W ratio increases for items further down the list.

The system and method for generating and displaying hybrid menus generates primary menus and sub-menus having similar or different geometries. As in the prior art, a context-appropriate hierarchy of model elements is retrieved as a function of the context in which a user provides selection input. Departing from the prior art, in which model elements were displayed using fixed display geometry, however, the inventive system queries the model for what type of view shall be used to display it. The model is then transformed to optimize user interaction performance characteristics associated with the views, while observing constraints to preserve semantic structure of the input model. Views of similar and differing geometries are bound together using a common geometric coupling constraint to form a hierarchical menu for display to the user. A layout generator then arranges items within each menu to optimize access time, while observing constraints that preserve the user's ability to build muscle memory of paths to item locations. For the special case of radially-arranged menus, it keeps item positions consistent between multiple invocation contexts using a preferred angle property on model elements, and a combinatorial optimization algorithm to minimize departure from specified angles when conflict occurs. The hybrid menu is then bound with interaction logic to form a fully functional command interface. Further, the hybrid menu is provided with the logic for detecting and responding to events affecting the layout of items in the hybrid menu, thereby facilitating automatic updating.

Fig. 5 is a diagram representatively illustrating a hybrid menu in accordance with the present invention. The hybrid menu addresses the disadvantages of the prior art radial menu by dynamically expanding radial pie segments to display modified linear sub-menus having labels. A hybrid menu may comprise any combination of radial and linear menus, as well as other geometries supported by the coupling constraint (further detailed below). For ease of description, and with reference to Fig. 5, the primary menu will refer to a

radial menu 500 having a linear sub-menu 502 which, in turn, has a radial sub-menu 544. Upon user selection of a radial segment, for example segment 501, the modified linear sub-menu, “Associations” sub-menu 502, is displayed. Where the prior art linear context menus formed long narrow tunnels to travel from menu to sub-menu, the hybrid menu includes paths from radial menu items in segments to linear sub-menus which are short and wide and easier to navigate. The modified linear sub-menus are positioned as close to their centers (i.e., their radial segment) as possible to cut the average distance to the target menu items. Since each sub-menu will expand in relation to a shared tangential line with its parent pie segment, user expectation is satisfied that the sub-menu for the selected radial pie segment will always be displayed in the same orientation to the primary radial display and that a desired sub-menu item, for example “Create Composition To...” at 542 will always be found in its currently-displayed location relative to the radial pie display. Moreover, as further detailed below, menu items in displays which are dynamically generated for different contexts will preferably be displayed in the same location relative to their primary radial displays, again to facilitate user memorization and ease of use. Upon user selection of sub-menu item 542, a sub-sub-menu 544 is displayed.

In order to generate a hybrid menu, model elements are first retrieved for the context and a tree structure, or other hierarchical model representation, is generated. Fig. 6A illustrates a sample uncompressed menu model showing the relationships between menu and sub-menu items. Fig. 6B illustrates generally how those items would be configured for linear display. Given the simple model illustrated, leaf items are labeled “L” and groups are labeled “G”. Fig. 6A provides a tree model of the menu items and Fig. 6B illustrates how those items would be displayed using a standard linear menu. The present invention seeks to automatically optimize the menu for fast access, optimal use of screen real estate, discoverability of items and any number of other factors that depend on

how the user interacts with the visual representation of the menu model. The hybrid menu system of the present invention optimizes the tree model using model element properties a further detailed below.

Figs. 7A and 7B illustrate the model of Fig. 6A optimized for a radial menu and for a linear menu, respectively. Since radial menus preferably have no more than 10 segments, and ideally have between 5 and 8 segments, the model of Fig. 6A is preferably evaluated to have 5-8 menu items in the primary radial display. Radial menus are ideal for handling sub-menus, providing a quick, constant distance, ease of memorization and a widened path for mouse movement toward sub-menu items. If the menu has fewer than the ideal number of items, for example 6 items, selected groups can be exploded. The “explosion” of a submenu moves all of that sub-menu’s items into the parent menu, surrounds them with separators, and removes the sub-menu. If there are too many items, all of the leaf items can be collected into a group. As shown in Fig. 7A, group G4 has been automatically exploded since it has only one child; G3 has been eliminated since it had no children, and G1 and G2 have been left intact, since exploding either or both of them would push the root’s child count past the ideal. In contrast, however, G1.1 has been exploded into G1, since it brings the item segment number closer to the ideal.

Fig. 7B shows the same model from Fig. 6A optimized for a linear menu. With linear menus, a higher item limit can be set for the primary display and sub-menus are usually exploded into the top level menu with separators. Fig. 7B shows the optimization of the Fig. 6A model wherein G4 has been automatically exploded since it had only one child item, G3 was eliminated for having no child items, and both G1 and G2 have been exploded into root due to the linear menu’s higher fan-out limit. As illustrated, the fan-out

limit was set at 9. Had the limit been higher, G1.1 would also have been exploded to the higher layer.

Under the present invention, a new model of model elements selected for the context is created each time a user provides selection input in order to generate a hybrid display. Once a tree or other hierarchical model is created, it is then transformed to generate the hybrid menu display. In evaluating menu items for generating the model, ordering of items is preferably undertaken. For example, ordering rules can be effected for ordering based on criteria such as historical usage, location consistency, or other factors. To optimize discovery and access times, for example, the most frequently used items should appear in a consistent position in primary radial menus or, if relegated to the linear sub-menus, should appear first in the linear sub-menus. Ideally items are ordered by the percentage of time that the items are selected by a user within a given task context. The percentage of time of user access may be predefined for a given context or may be dynamically tracked for a particular user. Tables of values representing the ordering criteria for each menu item preferably resides in one place where the menu items and associated values can be dynamically updated in relation to each other. For example, a user may enable different capabilities of an application which would cause an update to the ordering. The ordering under the present invention preferably orders groups and items within groups with the priority of groups being derived from the priority of the items within them.

Menu items are represented as model elements which are stored, for example in the aforementioned tables. When user selection input is received for a location and context, the relevant model elements are retrieved. Model elements would be, for example, the features that are available to the user relative to the context (e.g., program) and location

(e.g., a displayed object) at which the user selection input was detected. Each model element is stored with so-called dependency, or family, properties as well as display properties related to its ultimate display. The dependency or family properties are used for generating the hierarchical model. Model element display properties are used for transforming the hierarchical model into the hybrid display and include, but are not limited to, a preferred angle for display, a priority weight factor, an explicit group flag and a display menu preference. The preferred angle for display refers to a consistent location relative to a primary display or sub-menu item. As noted above, it is preferable that menu items be displayed consistently regardless of the context for which the menu items are being displayed. A preferred angle property tells where that menu item is preferably situated in a display. The priority weight factor property tells the importance of the model element. In dynamically generating hybrid menus, the higher priority menu items are preferably displayed in the primary radial display. The priority weight factory property is also used to order elements in linear sub-menus where position determines access times. The explicit group flag is used to flag an explicit group of menu items that should not be exploded during layout. Finally, the display menu preference property tells how the model element should be displayed. The display menu preference property may indicate “radial”, “linear” or any other geometry, or it may indicate “auto”. With an “auto” display menu preference property, the hybrid display system will automatically determine a preferred display menu for displaying the model element.

Fig. 8 is a process flow for automatically generating a hybrid menu in accordance with the present invention. The process is to dynamically generate context-dependent hybrid menus. The hybrid menus may comprise any combination of radial and linear menus, as well as other geometries supported by the coupling constraint, further detailed below. For ease of description, as above with reference to Fig. 5, the primary menu will

refer to a radial menu having a linear sub-menu which, in turn, has a radial sub-menu. The process of Fig. 8 starts with user selection input at 810. The user selection input may, for example, be a right click on a screen area, inactivity in a screen area at which a cursor/pointer is displayed, touch screen input, keystroke input, etc. The system determines where the input was received and identifies the context at step 812. Next, context-dependent model elements are retrieved given the context and area of display at 814. Once the model elements have been retrieved, a tree model, or other hierarchical model, is generated and optimized taking into account the model element properties of display menu preference, explicit group flag and priority at 816. The tree model is then transformed to generate the hybrid menu, at step 818. Transforming the optimized tree model comprises creating radial and linear menus from the optimized model, consistent with the display menu preference (i.e., radial/linear view type) followed by the layout process of arranging the items within each menu, taking into account the priority and preferred angle model element display properties. Each of the menu items is associated with interaction functionality. The generated hybrid menu includes the primary and all sub-menus. Initially only the primary menu is displayed at step 820. Thereafter, if the user selects one of the primary menu segments, the interaction will cause a sub-menu to be displayed.

Further detail of the transformation of the optimized tree menu into a hybrid menu are depicted in Figure 9. A Menu Controller 948 creates radial and linear menus from the optimized tree model to generate a menu figure 980. The figure controller creates instances of the Menu 980 and its child menus, a Layout Input Binding 940 which listens for event data, such as user interaction input or Menu Controller input that may affect the menu layout, a Layout Generator algorithm 950 which receives the input from the Layout Input Binding and executes to generate the layout of the hybrid menu and a Layout Output

Binding 960 which applies the changes to the menu figure 980. The Menu Controller then “wires everything up” by adding the children to the main Menu figure 980, setting the Layout Input Binding to listen for events on the main Menu figure, setting the main Menu figure’s input binding to the one it created, setting the input binding’s layout algorithm to the one it created, setting the Layout Algorithm’s output binding to the one it created, and setting the Layout Output Binding’s target to the main Menu figure.

When an event that changes layout occurs in the main figure (e.g. child added), the Layout Input Binding assesses whether the event affects parameters that the Layout Algorithm is interested in. If the event does affect parameters the layout is interested in, the Layout Input Binding instructs the Layout Algorithm to run. The Layout Algorithm pulls parameters of interest from the figure through the Layout Input Binding (which may provide some translation). During execution, the layout algorithm sends results to the Layout Output Binding, which applies them to the main Menu figure and its children (possibly with some translation) by either applying the results/changes immediately or buffering the results and applying all changes at once when complete. The Layout Output Binding may also buffer the results and pass them to an optional Animation Controller (not shown) which will apply the changes over time.

In ordering the menu items for generating the tree model, the system optimally selects a number of primary menu items that is fewer than, for example, 8 items when the model elements have a radial menu property indicated. If the identified model elements are fewer than 8, for example 6 primary model elements, the generated radial menu layout may include 8 segments, wherein 6 have icons and/or labels and 2 are non-selectable blank segments. Alternatively, the display will show only 6 segments. Dynamic model adjustment may be required if, for example, more than 8 model elements have similar

priority weight properties and have radial display properties. If the identified primary menu items exceed 8 in number, another iteration of menu items ordering (i.e., tree model optimization) is done, possibly including re-grouping of menu items, to achieve the goal of 8 or fewer primary menu item model elements, taking into account the explicit group flag property of the associated model element. In the instance where fewer than 5 model elements are identified for radial display in a primary menu, the dynamic model adjustment can determine if any identified sub-menu items have one child, in which case the sub-menu item(s) may be elevated to a primary menu display item. It will be understood that 5-8 segments have been chosen as optimal for display on an average user display device, including computer monitor, PDA, phone, etc. but that a larger or smaller number of segments can be defined as optimal. It is also contemplated that the hybrid menu display system can ascertain the user output display device, automatically or via user prompting, and dynamically tailor the primary and sub-menu displays for the current user output display device.

After the primary menu items have been determined, the linear sub-menu items are ordered for each primary menu item and the layout of the sub-menus is determined. As noted above with reference to Fig. 5, when a sub-menu is being displayed relative to a radial menu, the sub-menu display is optimally displayed in tangential relationship to its associated primary segment using the preferred angle display property. Additional identified menu items may fall into sub-menus (referred to as sub-sub-menus) of a given sub-menu. Depending upon the identified relationships among the menu items, the layout generation component will also automatically generate menu navigation features for display, including such artifacts as arrows, accelerator keys underlines, highlighting, etc. Display of the hybrid menu may include aspects that further facilitate user review and selection. Given the limited segment real estate, icons are preferably displayed in a

primary radial menu. Clicking on or hovering the cursor/pointer at the empty center of the radial menu for a predetermined elapsed time can cause display of the labels for all of the segments, ideally displayed outside of the periphery of the radial menu for maximum label word lengths. Clicking on or hovering over a single segment for a predetermined period will cause display of the label for the single segment's menu item. For radial menus, pressing the space bar could cause all label callouts to protrude from the menu at once, complete with accelerator key underlines. User selection could alternatively be done using the space bar, enter key or other inputs, as will be understood by one having skill in the art. Arrows are preferably provided to cycle through elements in the menu, and to expand or collapse sub-menus. For example, for linear menu navigation, the up and down arrows go to next/prev, while the left and right arrows go to expand/collapse. For radial menus, the system checks the angle of the currently selected item to see if it's in a top, left, right, or bottom quadrant. If in the top, left and right arrows do next/prev, an up arrow does expansion, and a down arrow is for collapse. If in bottom, left and right arrows do prev/next, an up arrow collapses and a down arrow expands. If to the right, up and down arrows do next/prev, a left arrow collapses and a right arrow expands, etc. Further, if the last navigation was a prev/next, and occurred within a short time of the current and identical key press (a repeat of the last action), then the last action will be repeated regardless of the current quadrant. The ellipsis queues would preferably remain present in the label callout, to indicate whether an item is a direct action or a link to cancellable dialog. The foregoing represents logic that has been implemented for user convenience, but clearly can be modified without departing from the spirit and scope of the present invention.

Fig. 10 illustrates a representative computer system for implementing the hybrid menu display system of the present invention. The hybrid menu display system may be

located at a user device or remote from the user device (e.g., at a LAN server or at a remote server in a cloud computing environment). In FIG. 10, the exemplary processing system 1000 includes a processing device 1010 for executing software components, including the Model Generator Component 1040, Model Optimizer Component 1044, Menu Controller 1048 and Layout Generator Component 1050. The system further has remote (not shown) or local memory 1020 coupled to the processor, as well as input/output (I/O) circuitry 1030 operative to interface with the processor and with a user graphical display device. The processor 1010 may be configured to load and execute features of the Model Generator Component 1040, Model Optimizer Component 1044, Menu Controller 1048 and Layout Generator Component 1050 to perform at least a portion of the methodologies of the present invention, illustrative embodiments of which are described herein. Display 1012 may be provided at the processing system 1000 for display to the user. Alternatively, the processing system 1000 will send the display information to a remotely-located graphical user interface 1016 at a user location 1014.

The hybrid menu is dynamic and scalable to accommodate new application features, etc. without requiring complete overhaul of the menu structure and its attendant disruption to user expectations. Moreover, a hybrid menu can automatically update itself. The current hybrid menu implementation is already initialized from a simple interface based menu model and can be extended to listen to that model for changes to the menu structure. Menu elements are arranged by a loosely-coupled and configurable layout algorithm that can handle a wide array of configurations. Ideally, a layer is provided in between existing menu contributions and the menu model that will intelligently optimize groupings to optimize hybrid menus. By taking into account menu groups, sub-menus, the presence of icons, the number of children in a group, and any other appropriate metrics,

the menu model will resolve any conflicts between linear and hybrid menu groupings to automatically reconfigure existing menus for display as hybrid menus.

The methodologies of embodiments of the invention may be particularly well-suited for use in an electronic device or alternative system. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “circuit,” “module” or “system.” Furthermore, the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

Any combination of one or more computer usable or computer readable medium(s) may be utilized. The computer-usable or computer-readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer-readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain or store a program for use by or in connection with an instruction execution system, apparatus or device.

A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as a part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electromagnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wire-line, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

Computer program code for carrying out operations of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

The present invention is described above with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the invention. It will be understood that each block of the

flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions.

These computer program instructions may be stored in a computer-readable medium that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable medium produce an article of manufacture including instruction means which implement the function/act specified in the flowchart and/or block diagram block or blocks.

The computer program instructions may be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks.

For example, FIG. 9 is a block diagram depicting an exemplary processing system 900 formed in accordance with an aspect of the invention. System 900 may include a processor 910, memory 920 coupled to the processor (e.g., via a bus 940 or alternative connection means), as well as input/output (I/O) circuitry 930 operative to interface with the processor. The processor 910 may be configured to perform at least a portion of the methodologies of the present invention, illustrative embodiments of which are shown in the above figures and described herein.

It is to be appreciated that the term “processor” as used herein is intended to include any processing device, such as, for example, one that includes a central processing unit (CPU) and/or other processing circuitry (e.g., digital signal processor (DSP), microprocessor, etc.). Additionally, it is to be understood that the term “processor” may

refer to more than one processing device, and that various elements associated with a processing device may be shared by other processing devices. The term “memory” as used herein is intended to include memory and other computer-readable media associated with a processor or CPU, such as, for example, random access memory (RAM), read only memory (ROM), fixed storage media (e.g., a hard drive), removable storage media (e.g., a diskette), flash memory, etc. Furthermore, the term “I/O circuitry” as used herein is intended to include, for example, one or more input devices (e.g., keyboard, mouse, etc.) for entering data to the processor, and/or one or more output devices (e.g., printer, monitor, etc.) for presenting the results associated with the processor.

The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

Implementation details may be modified without departing from the spirit and scope of the present invention. For example, the above description describes implementation

details for radial menus, including pressing the alt key to cause all label callouts to protrude from the menu at once, complete with accelerator key underlines, as well as left and right arrows to cycle through elements in the menu, the up arrow used to expand sub-menus, and the down arrow to collapse them. Clearly display alternatives may be implemented yet still be covered by the present invention.

While foregoing is directed to the preferred embodiment of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

CLAIMS

1. A computer-implemented method for dynamically generating context-dependent hybrid menus for display at graphical user interfaces comprising the steps of:

5 interpreting user input to the graphical user interface to determine input location and context;

selecting model elements representing menu items for said determined context;

generating a hierarchical model comprising the selected model elements; and

10 transforming the hierarchical model to generate a hybrid menu comprising at least one primary menu coupled to at least one sub-menu, at least one of the primary and at least one sub-menu comprising a radial menu;

wherein each of said model elements comprise model element properties and wherein the step of generating the hierarchical model comprises determining model element positioning in the hierarchical model based on the model element properties.

15 2. The method of Claim 1 further comprising the step of displaying at least the one primary menu at the graphical user interface.

3. The method of Claim 2 wherein the at least one primary menu comprises at least one indicator for user selection for display of at least one sub-menu and wherein the method further comprises the steps of:

receiving user selection input at said at least one indicator; and

20 displaying the at least one sub-menu associated with the selected indicator.

4. The method of Claim 1 wherein the model element properties comprise at least one of a preferred menu display geometry property, a preferred angle property, an explicit group property and a priority weight factor property.

25 5. The method of Claim 4 wherein said preferred menu display geometry property comprises at least one of radial and linear display.

6. The method of Claim 4 wherein the preferred menu display geometry property indicates no preference whereby said transforming comprises automatic menu display geometry determination.
7. The method of Claim 1 wherein said generating a hierarchical model comprises
5 creating a tree model.
8. The method of Claim 2 wherein said displaying comprises displaying a radial primary menu having segments and wherein said at least one sub-menu comprises at least one of a linear and radial sub-menu for selective display upon user input to at least one segment of the radial primary menu.
- 10 9. The method of Claim 1 wherein said generating a hierarchical model comprises ordering model elements based on a frequency of use value for said model elements in the determined context.
10. The method of Claim 9 wherein the frequency of use value comprises one of a predefined frequency value and a tracked frequency value for a given user.
- 15 11. The method of Claim 1 wherein at least one menu comprises a radial menu having segments and wherein said transforming comprises applying a coupling constraint to couple at least one sub-menu along a shared tangent with a corresponding radial menu segment.
12. The method of Claim 8 wherein said displaying a radial primary menu comprises
20 displaying a number of segments having icons.
13. The method of Claim 12 further comprising selectively displaying labels for said icons based on user input.
14. The method of Claim 1 further comprising at least one of adding and updating model elements based on newly added or newly enabled context features.
- 25 15. The method of Claim 1 wherein the primary menu comprises a radial menu and further comprising applying dynamic model adjustment when the number of model elements for said primary menu exceeds or falls below a predefined optimal number of radial segments.

16. The method of Claim 1 wherein said transforming the hierarchical model to generate a hybrid menu comprises steps of:

optimizing the hierarchical model based on the preferred menu display geometry property, the explicit group property and the priority weight factor property for each menu
5 item in the hierarchical model;

creating radial and linear menus of menu items from the optimized model, based on the preferred menu display geometry property of each menu item; and

arranging menu items within each menu using the preferred angle property and priority weight factor property for each menu item.

10 17. A system for dynamically generating context-dependent hybrid menus for display at graphical user interfaces, said system comprising at least one processing device for executing components and further comprising:

means for interpreting user input to the graphical user interface to determine input location and context;

15 means for selecting model elements representing menu items for said determined context;

means for generating a hierarchical model comprising the selected model elements; and

20 means for transforming the hierarchical model to generate a hybrid menu comprising at least one primary menu and at least one sub-menu of differing geometry;

wherein the means for transforming the hierarchical model further comprises a model optimizer for optimizing the hierarchical model based on model element properties of said model elements.

18. The system of Claim 17 further comprising at least one means for displaying at
25 least the one primary menu at the graphical user interface.

19. The system of Claim 17 further comprising at least one storage location for storing said model elements.

20. The system of Claim 17 wherein said model optimizer means optimizes the hierarchical model based on a preferred menu display geometry property, an explicit group property and a priority weight factor property for each menu item in the hierarchical model.

5 21. The system of Claim 18 wherein said means for transforming the hierarchical model comprises:

a menu controller for creating radial and linear menus of menu items from the optimized model, based on a preferred menu display geometry property of each menu item; and

10 a layout generator for arranging menu items within each menu using a preferred angle property and a priority weight factor property for each menu item.

22. A computer readable memory having recorded thereon statements and instruments for execution by a computer to carry out the method of any one of claims 1 to 16.

15

20

Add UML	▶
Add Java	▶
Add	▶
Make Default Diagram	

Fig. 1A

Add UML	▶	
Add Java	▶	
Add	▶	Note
		Text
Make Default Diagram		

Fig. 1B

Add UML	▶	
Add Java	▶	Text
Add	▶	Note
Make Default Diagram		

Fig. 1C

2/9
CA920090028CA1

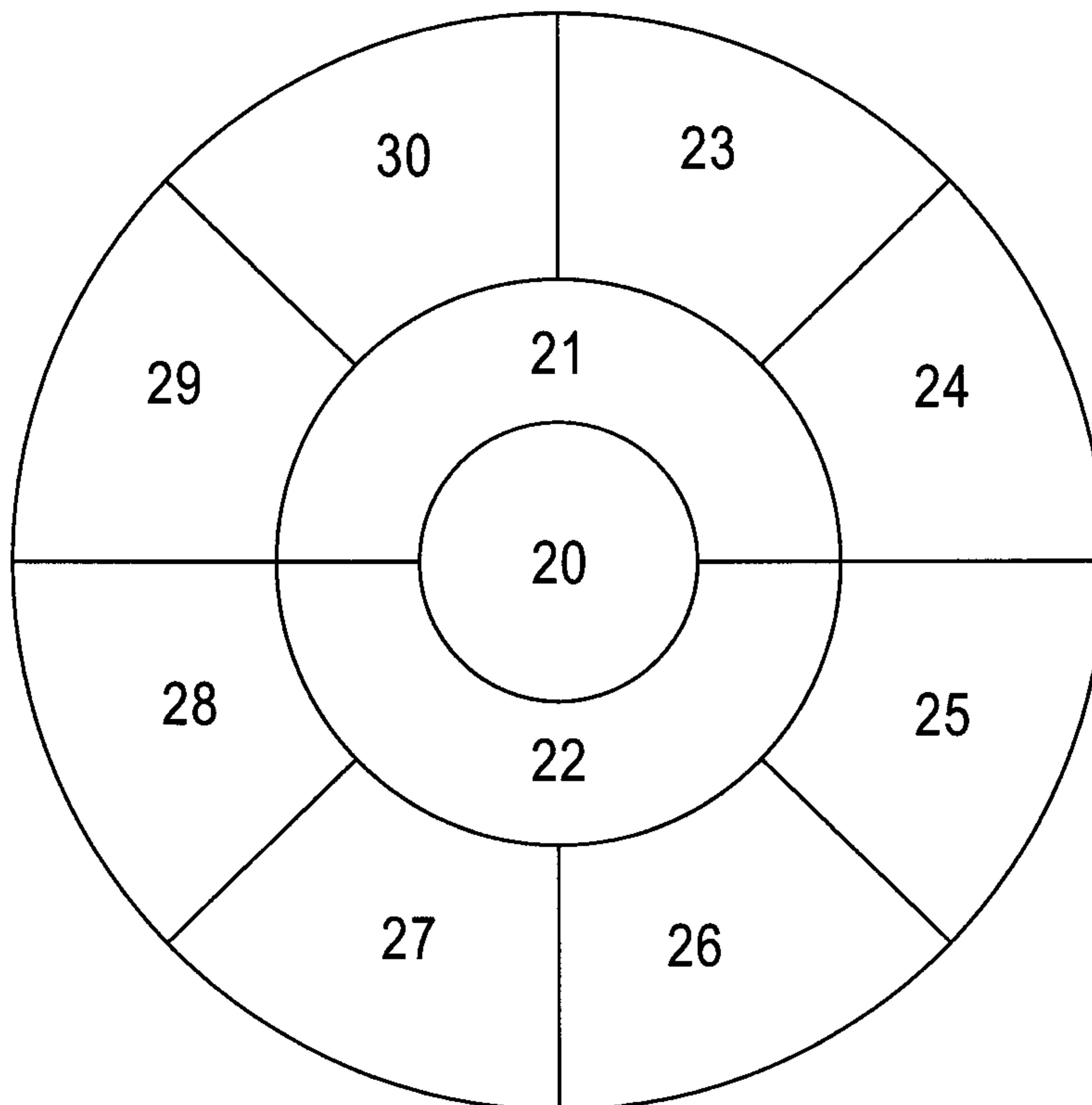


Fig. 2

3/9
CA920090028CA1

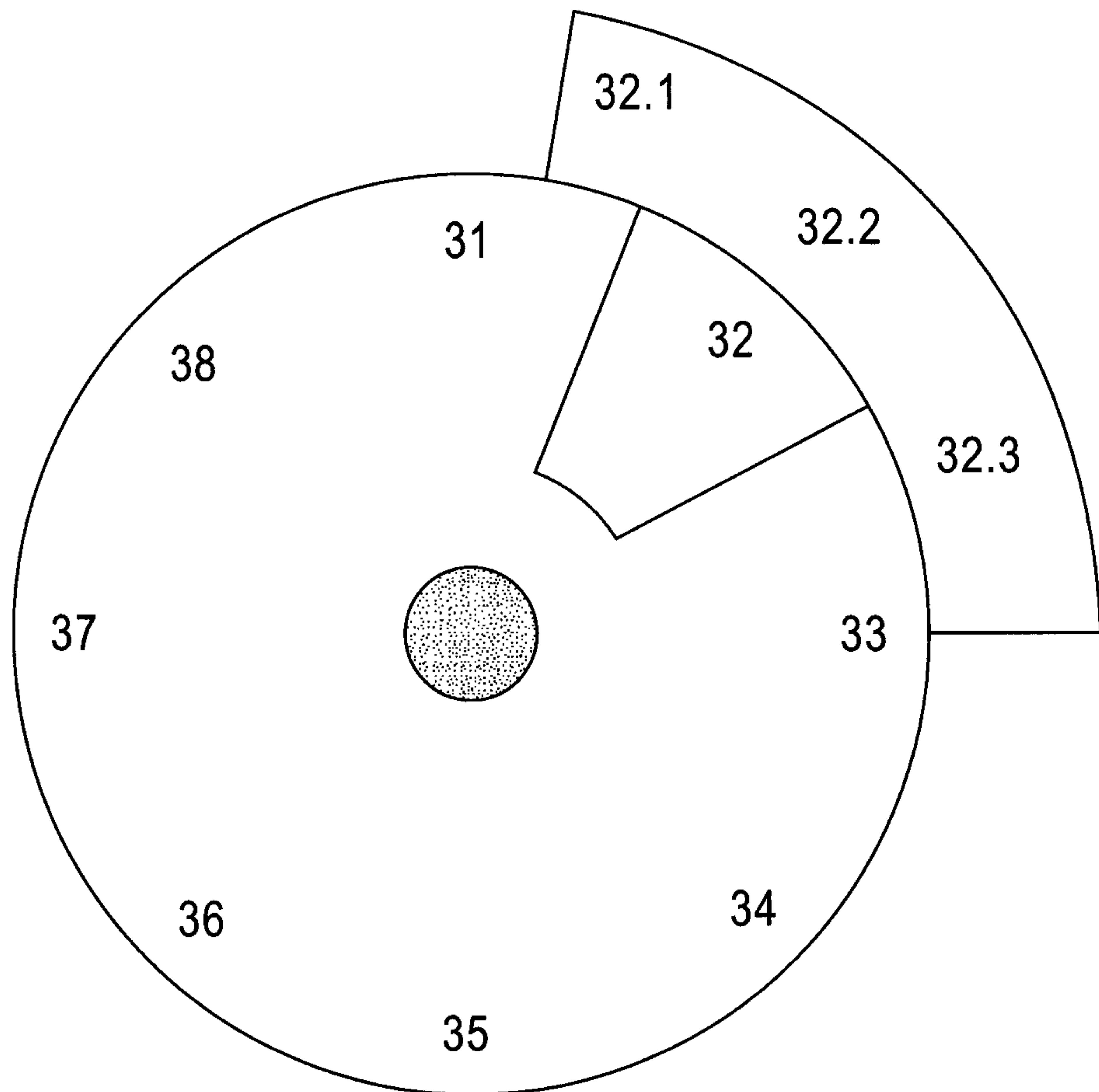


Fig. 3

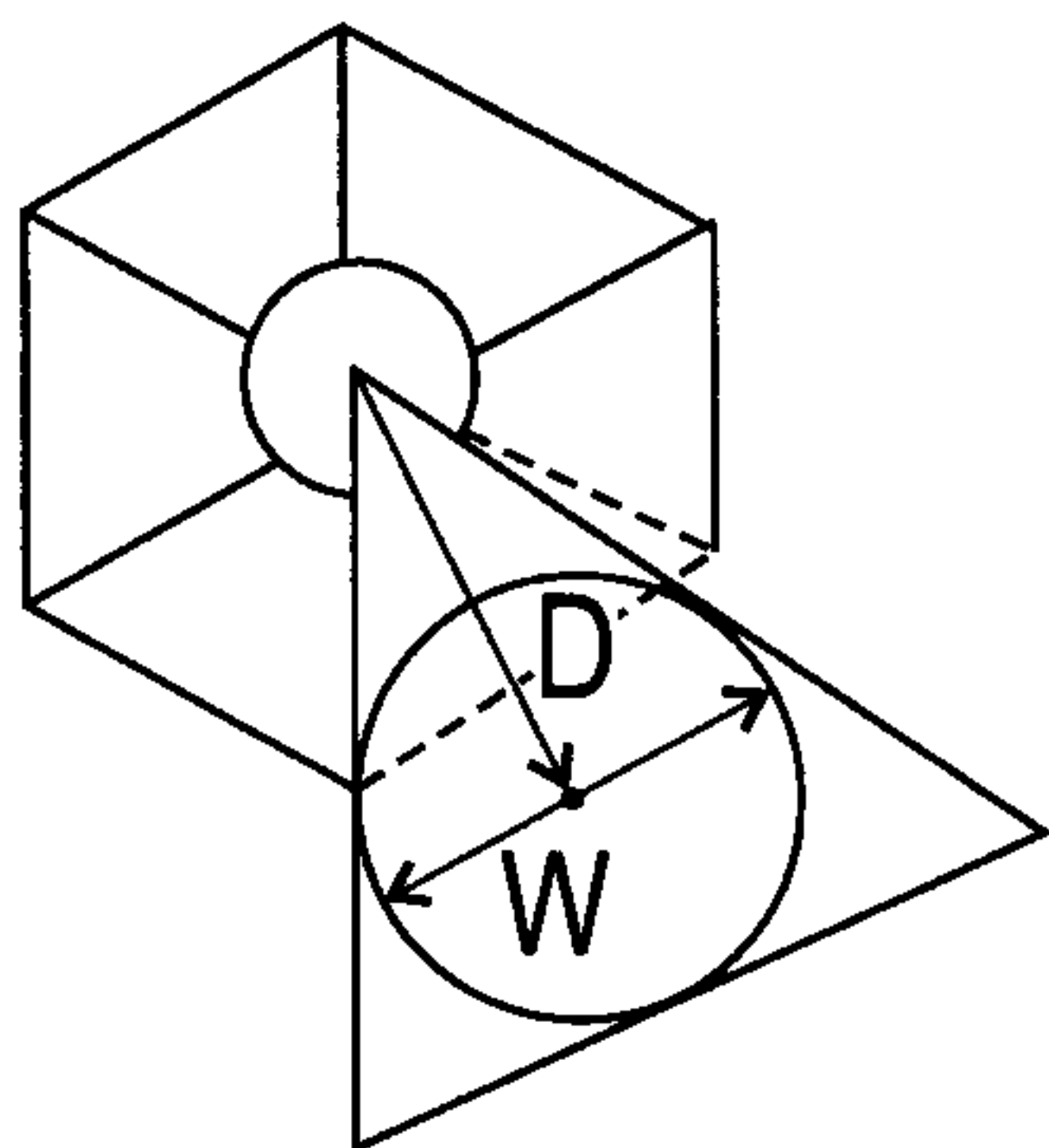


Fig. 4A

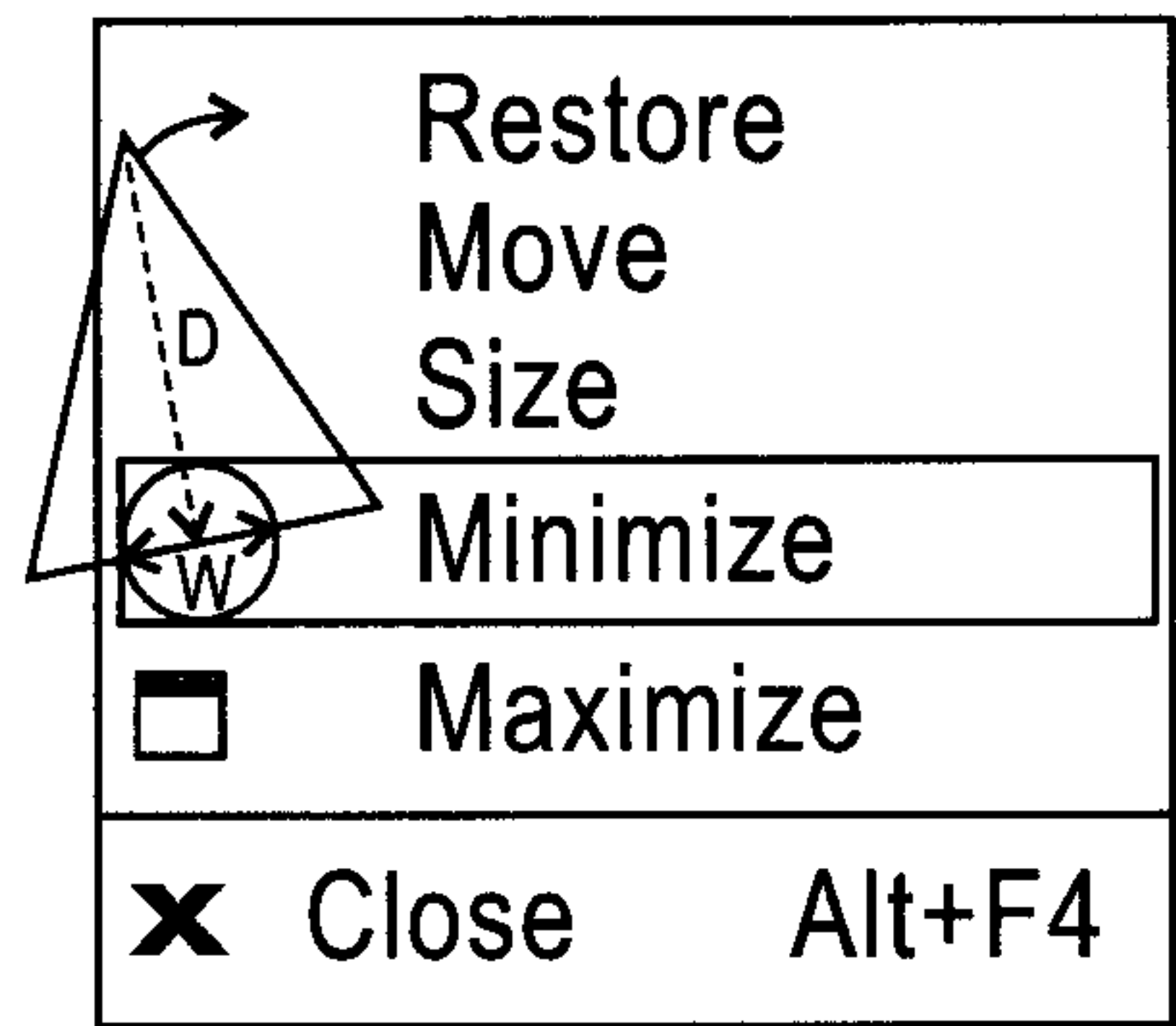


Fig. 4B

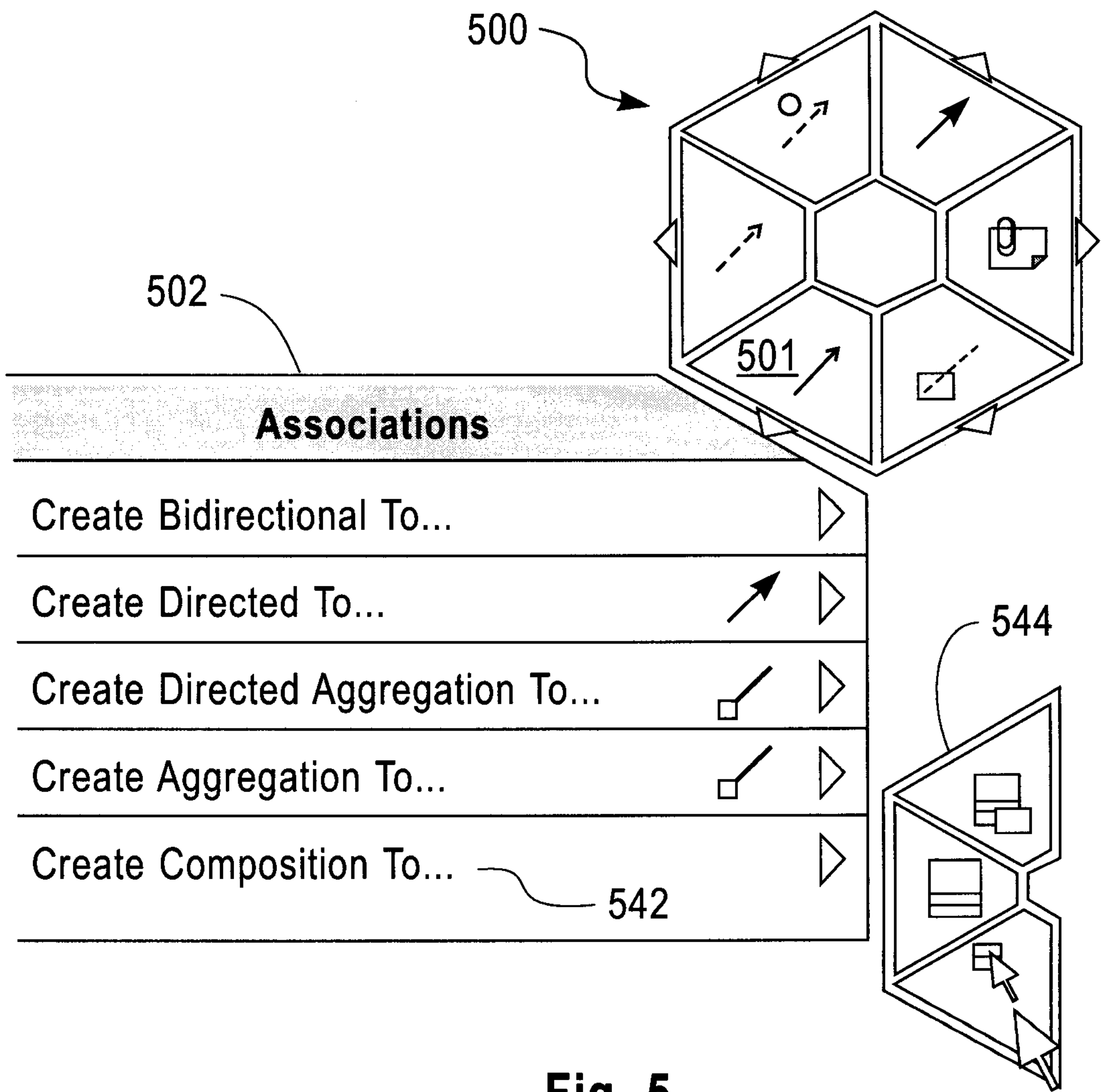


Fig. 5

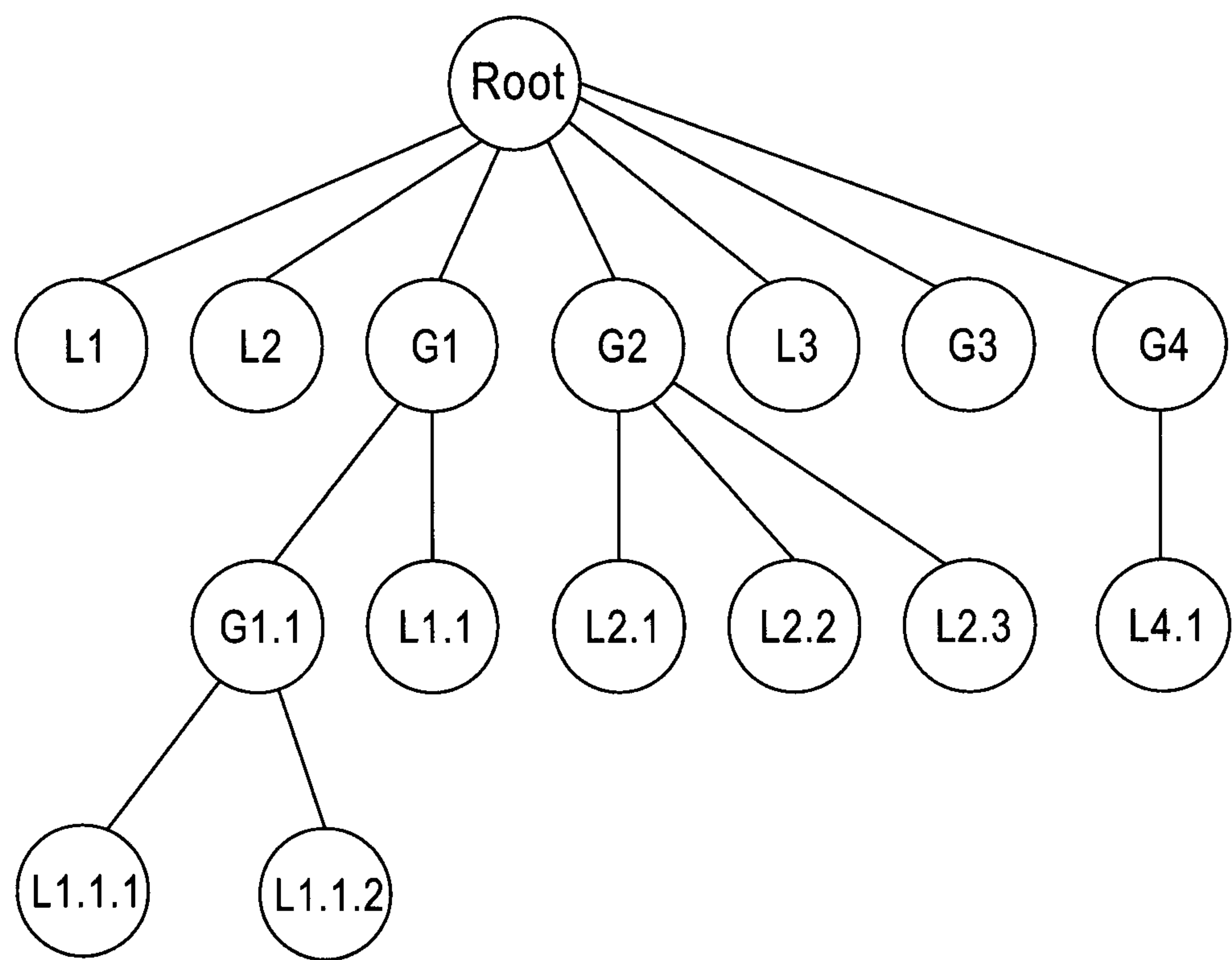


FIG. 6A

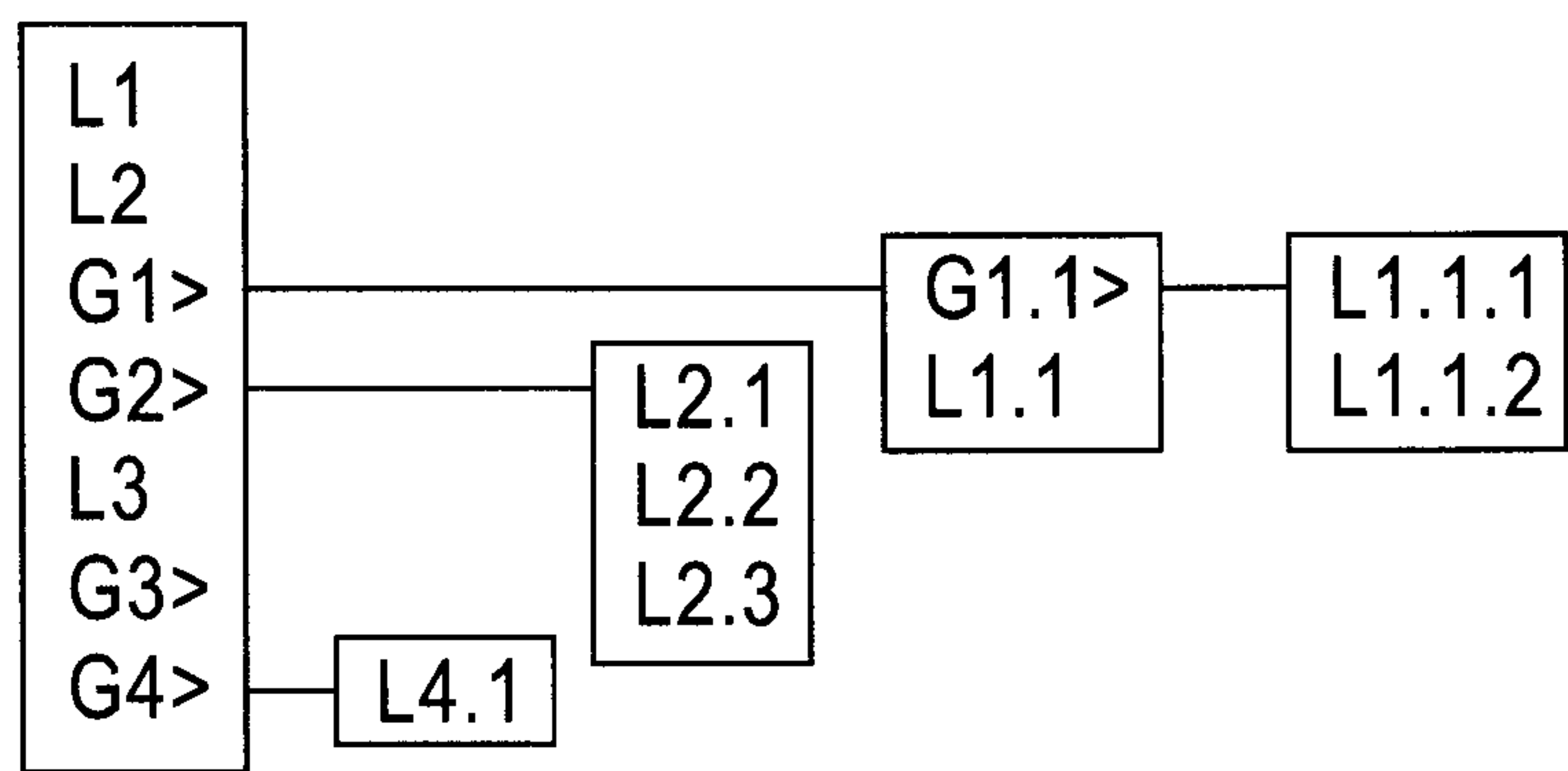


FIG. 6B

6/9
CA920090028CA1

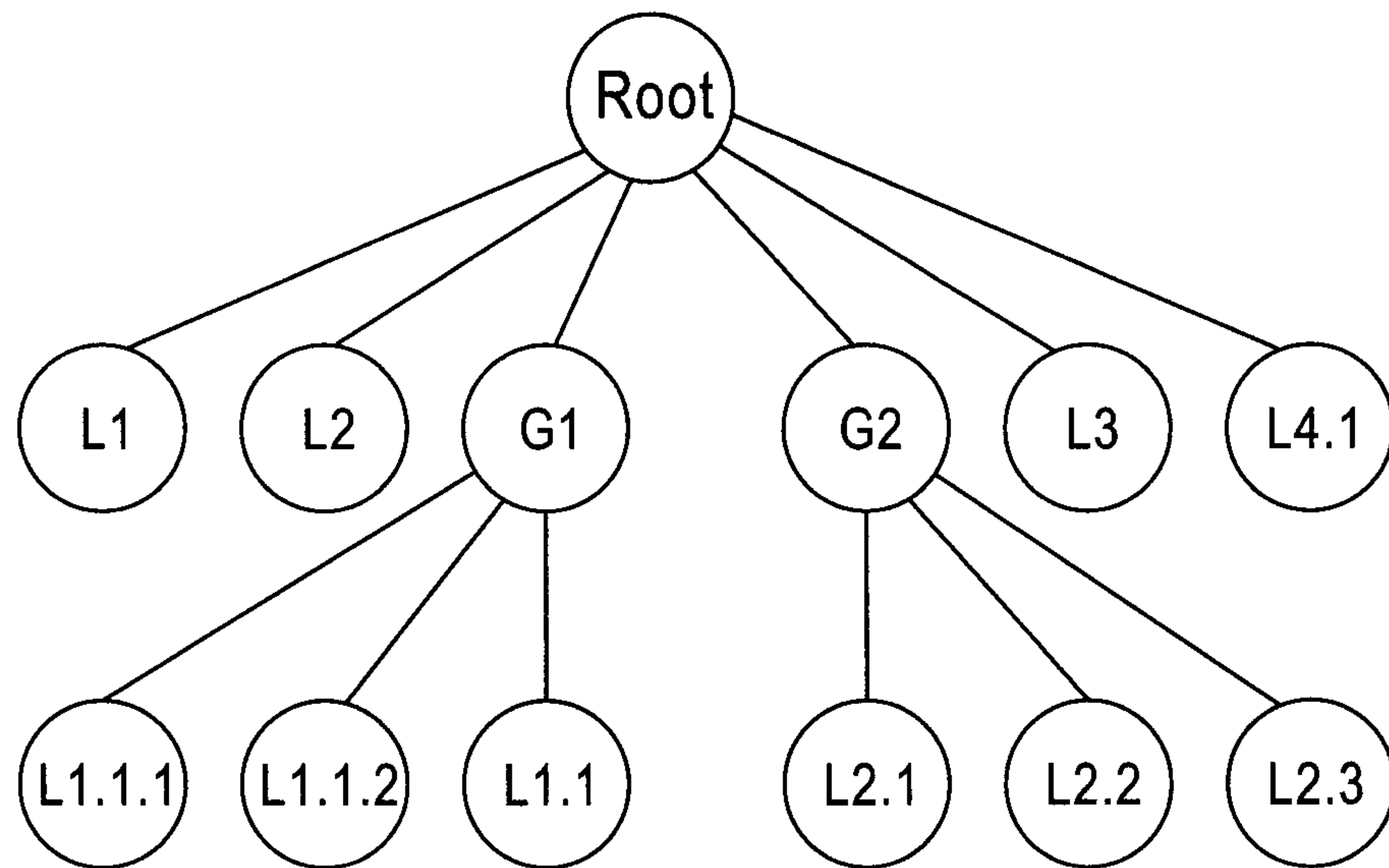


FIG. 7A

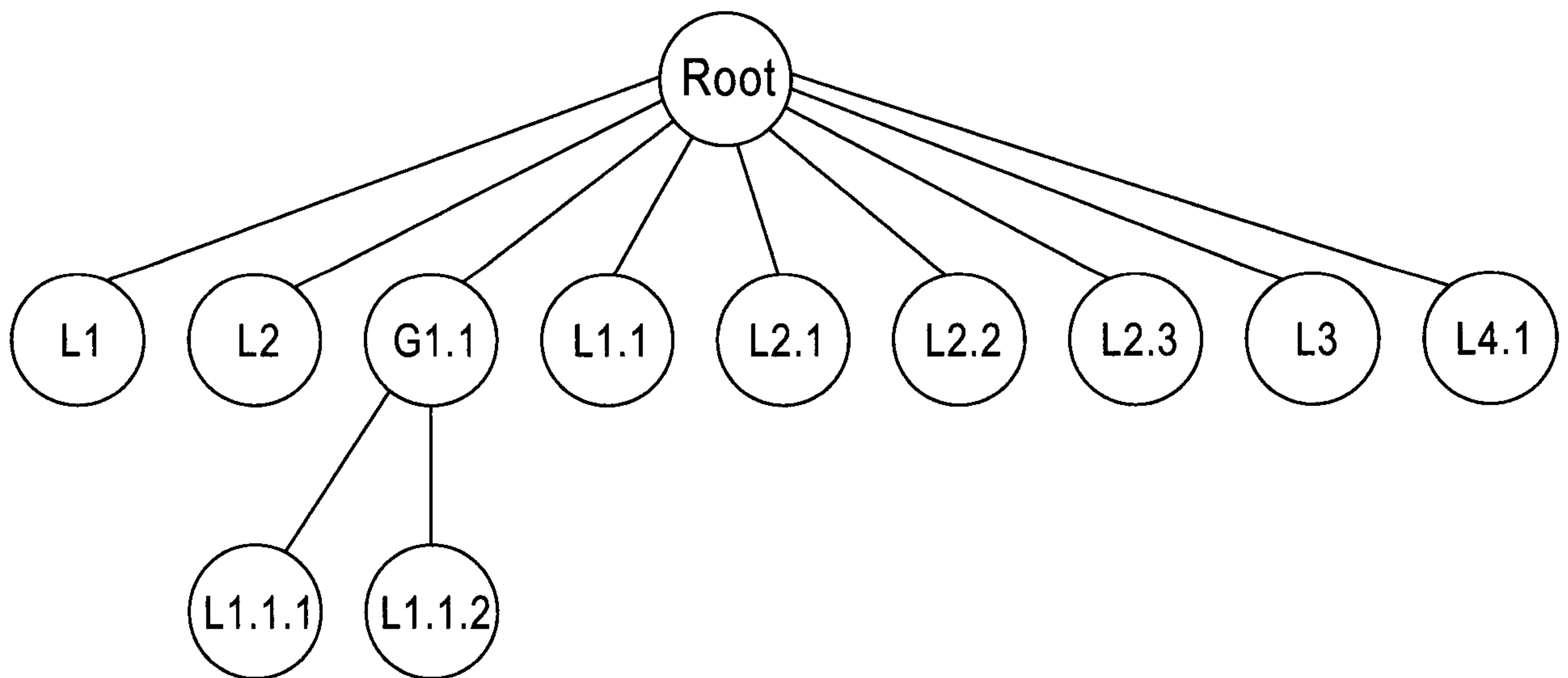


FIG. 7B

7/9
CA920090028CA

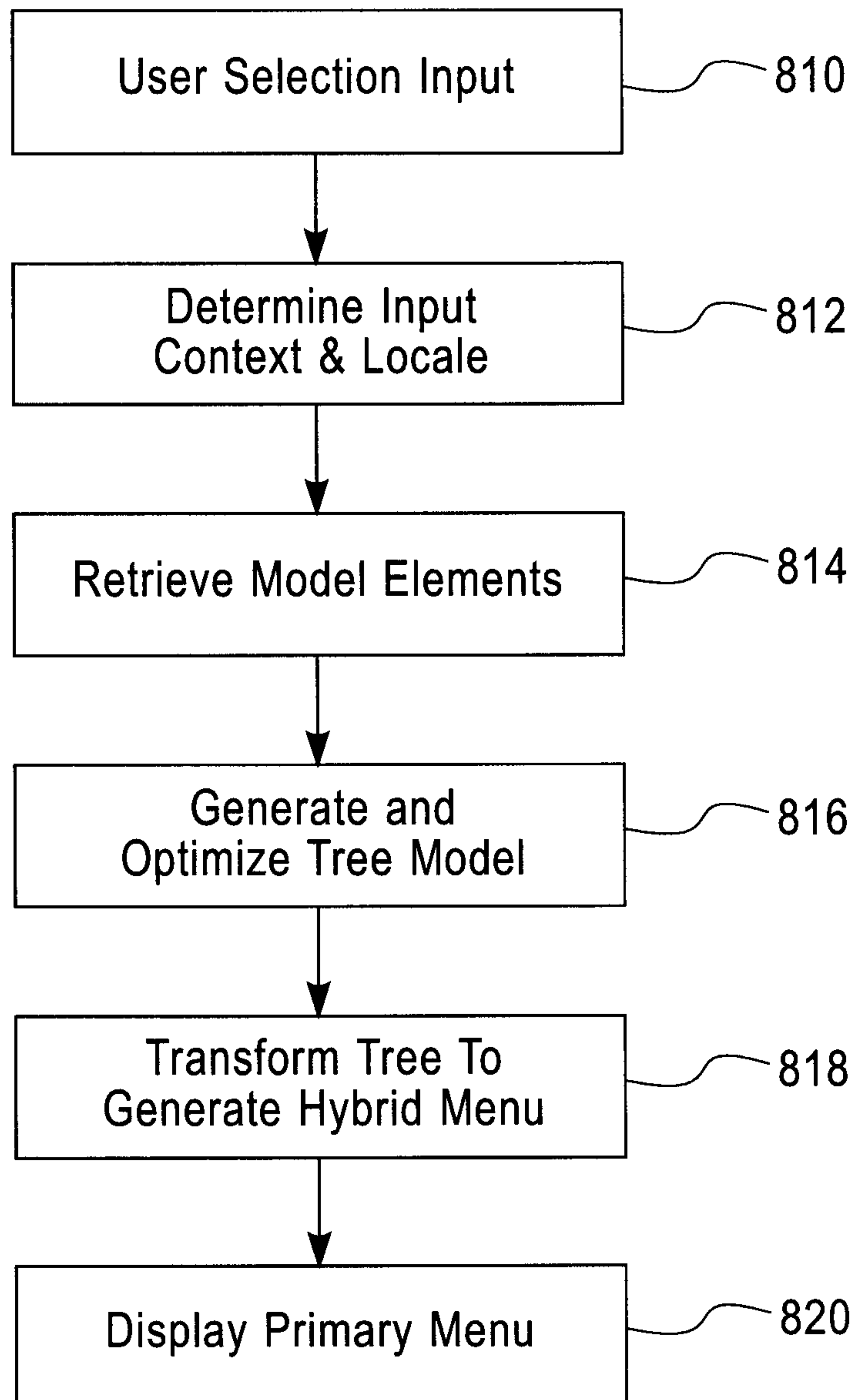


Fig. 8

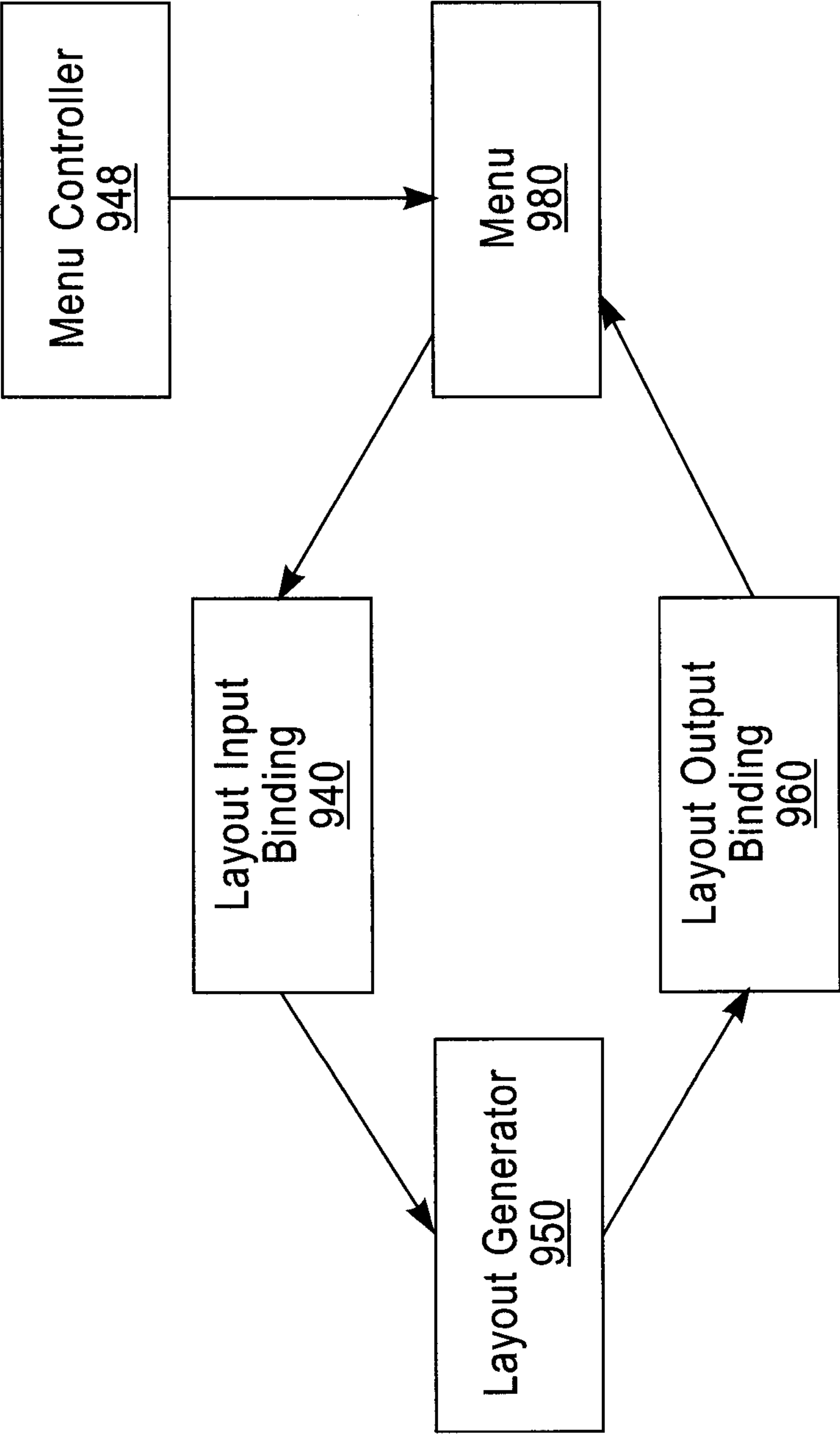


FIG. 9

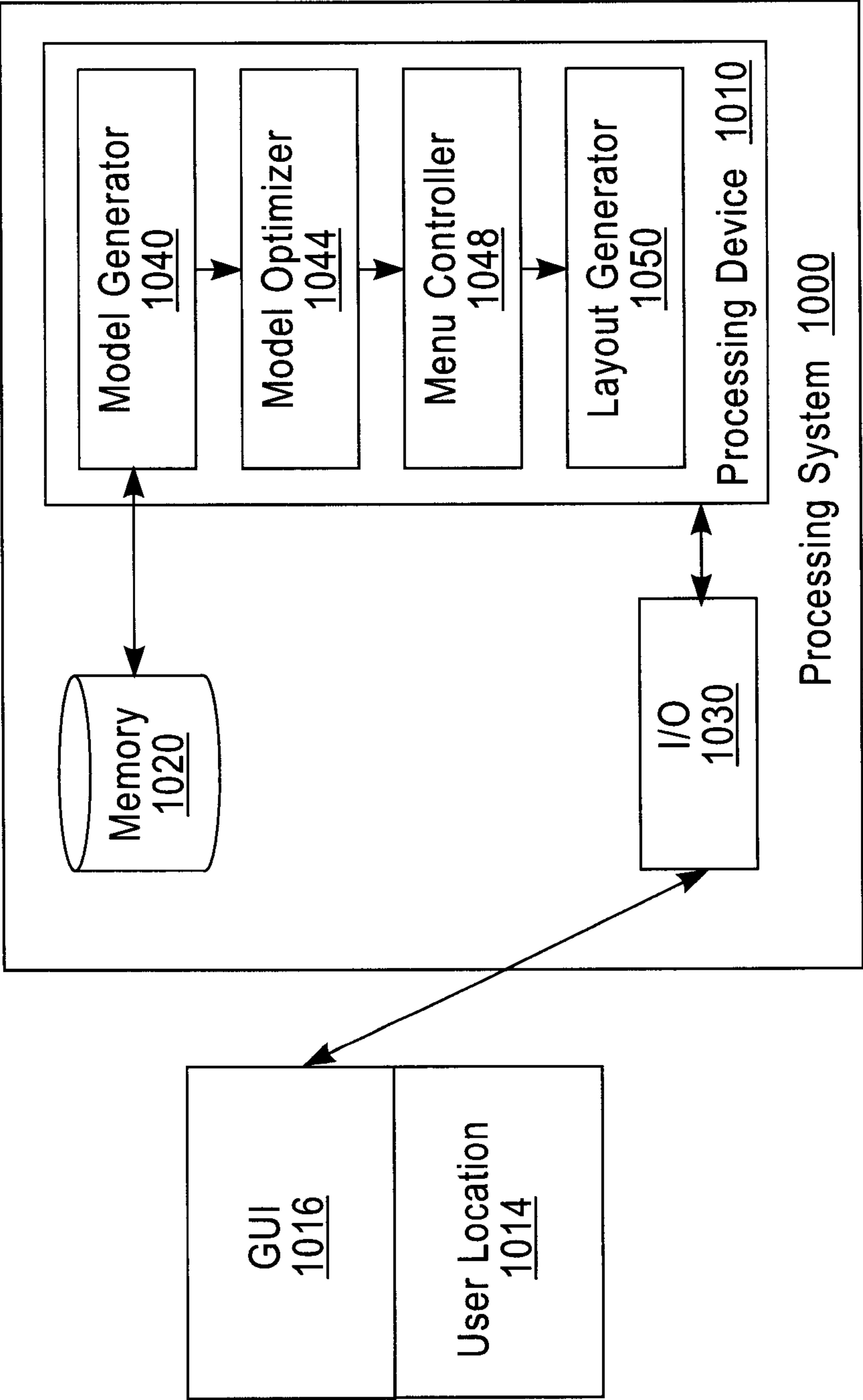


Fig. 10

User Selection Input

810

**Determine Input
Context & Locale**

812

Retrieve Model Elements

814

**Generate and
Optimize Tree Model**

816

**Transform Tree To
Generate Hybrid Menu**

818

Display Primary Menu

820

