

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
10 January 2008 (10.01.2008)

PCT

(10) International Publication Number
WO 2008/006000 A2

(51) International Patent Classification:
G06F 19/00 (2006.01)

(21) International Application Number:
PCT/US2007/072784

(22) International Filing Date: 3 July 2007 (03.07.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/806,569 5 July 2006 (05.07.2006) US

(71) Applicant (for all designated States except US): **YISSUM RESEARCH DEVELOPMENT COMPANY OF THE HEBREW UNIVERSITY OF JERUSALEM** [IL/IL]; Hi-Tech Park, Edmond J. Safra Campus, Givat-Ram, Post Office Box 39135, 91390 Jerusalem (IL).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **JAFFE, Elliot** [US/IL]; Pob 1787, 3 Bareket, 73127 Hashmonaem (IL). **DAYAN, Aviva** [IL/IL]; 6/2 Chopin Street, 91390

Jerusalem (IL). **DEKEL, Amnon** [IL/IL]; Apartment 3, 9 Michal Street, 63261 Tel Aviv (IL).

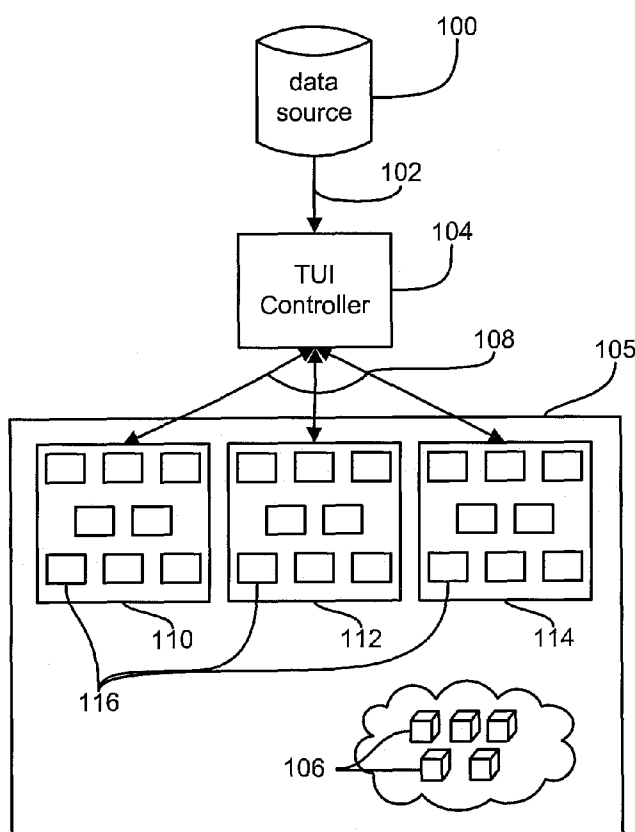
(74) Agents: **HUNTER, Paul S.** et al.; Foley & Lardner LLP, 150 East Gilman Street, Post Office Box 1497, Madison, Wisconsin 53701-1497 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: MONITORING APPARATUS FOR LARGE SCALE SYSTEMS



(57) Abstract: An apparatus monitors large-scale computer systems and exists as a physical entity separate from the computer system administrator's monitor. The apparatus includes a plurality of interface elements and a plurality of interface holders. The interface elements can be cubes equipped with a liquid crystal display (LCD) and with light emitting diodes (LEDs) which provide a user with system information. The interface holders can be locations on a tray, including a main area, a context area, and an aggregate area.



European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

- *without international search report and to be republished upon receipt of that report*

MONITORING APPARATUS FOR LARGE SCALE SYSTEMS

FIELD

[0001] The subject of the disclosure relates generally to computer management systems. More specifically, the disclosure relates to an apparatus and a system for monitoring large-scale computer systems.

BACKGROUND

[0002] Monitoring of any large-scale system requires situational awareness. In some installations, there are a team of administrators in a monitoring center tracking a large number of systems. When a problem is identified, someone other than the administration team is likely to be assigned to investigate and possibly repair it. Another complicating factor for monitoring such systems is that the systems can be so large that it is not possible to view the entire system at once or in one pass. Thus, administrators must maintain their location within the system as well as the current whole system view.

[0003] In general, the monitoring process begins with the identification of an afflicted or misbehaving system component. Monitoring could also be used to identify behavior patterns. Conventional software tools exist which display problematic events as they occur, providing a page-by-page view of the system. However, web-based systems like Nagios (an open source host, service and network monitoring program) have the disadvantage that once an administrator navigates away from a current page, the information of that page is lost. An administrator cannot select items to review later.

[0004] Yet another challenge to monitoring systems is the fact that monitoring systems generally share display space with a system administrator's operational environment. Thus, if an administrator notices that a web server has stopped responding to HTTP traffic and he opens a new window to investigate that problem server, the new window may hide (either partially or totally) the monitoring window, making it difficult or impossible to view new events or additional problems. Furthermore, a system administrator may lose interest or

attention in the monitoring program due to a limited number of alerts and, thus, may hide the monitoring display with another display.

SUMMARY

[0005] According to an exemplary embodiment, an apparatus monitors large-scale computer systems and exists as a physical entity separate from the computer system administrator's monitor. The apparatus includes a plurality of interface elements and a plurality of interface holders. The interface elements can be cubes equipped with a liquid crystal display (LCD) and with light emitting diodes (LEDs) among other elements which provide a user with system information. The interface holders can be locations on a tray, including a main area, a context area, and an aggregate area. Additional, fewer, or different areas may be utilized.

[0006] In a system in accordance with an exemplary embodiment, nested cubes are used with the monitoring apparatus to represent different levels in a hierarchy of data sources. For example, at a top level, cubes can represent racks of computer servers. At an intermediate level, cubes represent computer servers and, at a low level, cubes represent components of an individual server.

[0007] According to an exemplary embodiment, the monitoring apparatus allows the user to monitor the overall status of a computer system at a glance. When the computer system is working properly, the constituent interface elements of the apparatus glow a uniform green (or some other color or indication). When an immediate or anticipated failure event occurs in any part of the tree, a system monitor detects the failure event and informs a controller. The controller updates the interface so that the interface element associated with the highest level component that includes the affected component glows red or yellow (or some other color or indication), alerting the user to the problem. The user then interacts with the interface to explore the system tree and locate the fault. At each level, the interface associated with the element that contains the fault glows red or yellow. The user can follow this trail of red and yellow glowing interface elements by moving them to a context area one by one. At the end of this trail, the user finds the interface element associated with the faulty component, and can take steps to correct the problem. In some embodiments, the user can

remove this interface element from the tangible user interface and have it preserve its current display. It can then be physically transferred to another location, where it can either provide information about the problem, or, in some embodiments, act as a security token that temporarily gives permission to perform actions that will correct the problem. Once the problem has been corrected, the user can retrace her steps through the tree by moving the interface elements from the context area back to the main presentation area in reverse order. According to an exemplary embodiment, the interface element acts in a double capacity, both as a display token and as a data or security token.

[0008] Further objects, features and advantages will be apparent from the following detailed description when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Exemplary embodiments will hereafter be described with reference to the accompanying drawings.

[0010] FIG. 1 is a general diagram representing a tangible user interface system in accordance with an exemplary embodiment.

[0011] FIGS. 2A, 2B, and 2C are diagrams illustrating the use and operations of tangible interface element holders in accordance with exemplary embodiments.

[0012] FIG. 2D is a diagram illustrating an exemplary tangible interface element including a controller in accordance with an exemplary embodiment.

[0013] FIG. 3A is a flow diagram depicting operations performed in the operation of a tangible user interface controller in accordance with an exemplary embodiment.

[0014] FIG. 3B is a flow diagram depicting operations performed in the changed-data processing operation of FIG. 3A in accordance with an exemplary embodiment.

[0015] FIG. 3C is a flow diagram depicting operations performed in the changed-interface processing operation of FIG. 3A in accordance with an exemplary embodiment.

[0016] FIG. 3D is a flow diagram depicting operations performed in the remove-active-node operation of FIG. 3A in accordance with an exemplary embodiment.

[0017] FIG. 3E is a flow diagram depicting operations performed in the update interface operation of FIG. 3A in accordance with an exemplary embodiment.

[0018] FIG. 4 is a diagram illustrating a tray-based embodiment of a tangible user interface system for monitoring computer systems in accordance with an exemplary embodiment.

[0019] FIG. 5 is a diagram depicting a tangible user interface system configured for monitoring computer systems in accordance with an exemplary embodiment.

[0020] FIG. 6A, 6B, 6C, and 6D are diagrams depicting the use and operation of the tray-based embodiment of FIG. 4 in accordance with exemplary embodiments.

[0021] FIG. 7 is a diagram representing a tangible user interface system in accordance with another exemplary embodiment.

DETAILED DESCRIPTION

[0022] With reference to the drawings, **FIG. 1** illustrates a tangible user interface (TUI) system for exploring hierarchical data sources. This exploration enables the monitoring of data sources within the hierarchy. The exemplary embodiment shown includes a data source **100**, a TUI controller **104**, a set of tangible interface elements (TIEs) **106**, and three collections of TIE holders **116**, namely a main data presentation (MDP) collection **110**, a context collection **112**, and an aggregate collection **114**. The three collections of TIE holders **116**, along with the TIEs **106**, collectively compose tangible user interface **105**. The TUI controller **104** communicates with the data source **100** via the data-source link **102**, and with the three collections of TIE holders **116** via the interface links **108**. Each tangible interface element **106** fits into any one of the TIE holders **116** in any of the collections of TIE holders.

[0023] The TUI controller **104** obtains data from the data source **100**. The data source **100** may store the data structure of the data explicitly or implicitly. An example of explicit storage is a data source that maintains a tree representation of the structure and modifies it whenever the structure changes. An example of implicit storage is a data source that maintains an initial description of the structure together with an ordered record of all the changes it experiences. The TUI system can be used to monitor computer systems, air traffic systems, hospitals, emergency services, and a variety of other large scale systems.

[0024] **FIGS. 2A, 2B AND 2C** show exemplary embodiments of the collections of TIE holders 116. **FIG. 2A** is a block diagram of an exemplary embodiment of the MDP collection 110. A MDP collection controller 200 communicates with a plurality of TIE holders 116 via MDP links 202. Each TIE holder 116 has the ability to communicate with any TIE it might hold via a TIE link 204. Similarly, **FIG. 2B** shows an exemplary embodiment of the context collection 112, comprising a context collection controller 210, which communicates with a plurality of TIE holders 116 and in at least one embodiment with a TIE holder selector 214. The TIE holder selector 214 is a mechanism for designating one of the TIE holders in the collection as “selected.” Each TIE holder 116 is again equipped with a TIE link 204. **FIG. 2C** shows an exemplary embodiment of the aggregate collection 114 which comprises an aggregate collection controller 220, which communicates with a plurality of TIE holders 116, each equipped with a TIE link 204. The TIE links 204 may be embodied as physical links, wireless links, or any other means for transferring data.

[0025] **FIG. 2D** is a block diagram of an exemplary embodiment of a TIE 106, including a TIE controller 260, a TIE link 204, a plurality of display elements 262, and a plurality of alert presentation elements 264. The display elements 262 are capable of displaying data. For example, they may consist of one or more Liquid Crystal Display (LCD) screens that can show text and/or graphics. Additionally, they may have other components by which a user can choose which of a number of different messages she will see on the screen or screens, and scroll between these messages. The alert presentation elements 264 are capable of presenting any one of a set of alerts. For example, they may be embodied as a set of colored lights that light up differently depending on the alert being presented. In another embodiment, they might cause the whole TIE 106 to glow in different colors, the colors corresponding to the different alerts. In still another embodiment, they might produce different sounds or sets of sounds corresponding to the different alerts. The TIE controller 260 can be connected to a TIE holder 116 via the TIE link 204. When it is so connected, the TIE controller 260 receives data to display and alerts to present along the TIE link 204. The TIE controller 260 causes data to be displayed on the display elements 262 and alerts to be presented via the alert presentation elements 264. The TIE 106 may also have the capacity to hold abstract data which can be transferred to a separate device for display and/or

processing. The data could be a password or data ID needed to retrieve data from a server, or the data itself. In some embodiments, when TIE 106 is removed from the tangible user interface, the display elements 262 and alert presentation elements 264 continue to display and present the same information they did before they were removed.

[0026] Although presented as such above by way of illustration, the collection controllers and/or the TIE controllers need not be separate entities. For example, all three collection controllers could be embodied in a single controller, as in the tray-based embodiment described below. As a further example, the invention could be embodied without the TIE controllers. Some or all of the TIEs could instead be controlled directly by the collection controller of the collection of TIE holders in which they sit, or, alternatively, even by the TUI controller.

[0027] Referring again to FIG. 1, the TUI controller 104 communicates with both the data source 100 and the three collections of TIE holders. From the data source 100, it receives information about the hierarchical data structure being explored; from the three collections of TIE holders, it receives information about how the user has interacted with the tangible user interface 105.

[0028] The TUI controller 104 makes and maintains associations between TIEs 106 in the tangible user interface 105 and nodes in the hierarchical data structure. Each TIE can be associated with either (a) a single node, or (b) a set of nodes that have a common parent. TIEs of the latter type are referred to as “aggregate TIEs.” The association between TIEs 106 and nodes in the data structure can be maintained in many ways. For example, the TUI controller 104 might store an association list that records which nodes are identified with which TIEs. As another example, the TUI controller 104 might cause the associations to be stored in the TIE controllers, rather than in its own storage area.

[0029] The TUI controller 104 also implicitly or explicitly records which node or nodes in the hierarchical data structure is/are presently being explored. These records are referred to as the “active node” or the “active nodes.” The TUI controller might also implicitly or explicitly record a “context set,” which is the set of nodes associated with the TIEs in TIE holders in the context collection. In some embodiments, the TUI controller 104

might also maintain an internal representation of all or part of the data structure being explored.

[0030] The set of nodes about which information is to be presented at any given time is referred to as the “presentation set.” If a single node is active, the presentation set comprises that node’s children. If a set of nodes is active, the presentation set comprises those nodes themselves. The TUI controller **104** ensures that the tangible user interface **105** always presents information about the nodes in the presentation set.

[0031] **FIG. 3A** illustrates the operation of the TUI controller **104**. Additional, fewer, or different operations may be performed depending on the embodiment. In an operation **302**, the TUI controller checks if the data in the data source have changed. If so, the TUI controller performs a changed-data processing operation **304**. Otherwise, the TUI controller performs an operation **306**, at which the TUI controller checks if the state of the tangible user interface has changed—for instance, because the user has moved one of the TIEs. If so, the TUI controller performs a changed-interface processing operation **308**. The TUI controller then repeats this sequence of operations.

[0032] For purposes of illustration, a synchronous set of operations has been described, in which the TUI controller actively polls the data source and the tangible user interface to see if changes have occurred. However, an alternative embodiment is possible in which the TUI controller receives information about either or both of these changes asynchronously. In such an embodiment, the TUI controller does not actively poll the data source and the collections of TUI holders for changes. Instead, one or both of these components actively sends a signal to the TUI controller when changes occur. The TUI controller then runs the appropriate processing operation.

[0033] **FIG. 3B** illustrates an exemplary embodiment of the changed-data processing operation **304** of **FIG. 3A**. In an operation **322**, the TUI controller retrieves data from the data source. In different embodiments, it may retrieve the entire hierarchical data structure, only a part of it, or only a list of recent changes to it. The TUI controller then performs an update operation **324**, at which it updates its internal representation (if any) of the data structure. In an operation **326**, the TUI controller checks if any changes to the data

affect the view of the data currently being presented in the tangible user interface. If so, the TUI controller performs an update interface operation **328**.

[0034] **FIG. 3C** illustrates an exemplary embodiment of the changed-interface processing operation **308**. In an operation **340**, the TUI controller identifies the change that has taken place in the interface. At least seven types of interface changes are possible: six that reflect the ways in which the TIEs could have been moved, and one that reflects the use of the TIE holder selector in the context collection. With reference to **FIG. 1**, a TIE might have been moved: from a TIE holder in the MDP collection **110** to one in the context collection **112**; from a TIE holder in the MDP collection **110** to one in the aggregate collection **114**; from a TIE holder in the context collection **112** to one in the MDP collection **110**; from a TIE holder in the context collection **112** to one in the aggregate collection **114**; from a TIE holder in the aggregate collection **114** to one in the MDP collection **110**; or from a TIE holder in the aggregate collection **114** to one in the context collection **112**. Additionally, the TIE holder selector may be used to change which of the TIE holders in the context collection is designated as “selected.”

[0035] If the TUI controller **104** detects that a TIE has moved from a TIE holder in the MDP collection **110** to one in the context collection **112**, it performs an operation **342** (**FIG. 3C**), wherein it identifies if the recently-moved TIE is associated with a leaf node in the hierarchical data structure (i.e. one that has no descendants therein). If so, it performs an operation **344**, wherein it presents information associated with that leaf node via the tangible user interface. In some embodiments, this might include, for example, historical information for that node, showing how some characteristic or characteristics associated with it have evolved over time. In other embodiments, no information may be associated with a leaf node, and the tangible user interface might be updated to present blank TIEs. If the recently-moved TIE was not associated with a leaf node, the TUI controller performs an operation **346**. In this operation **346**, it (a) marks the currently active node(s) as no longer active, and (b) marks the node(s) associated with the recently-moved TIE as the active node(s). It then performs the update interface operation **328** whereby the tangible user interface is updated to present the appropriate information for this node or nodes.

[0036] If the TUI controller detects that a TIE has moved from a TIE holder in the MDP collection to one in the aggregate collection, it performs the update interface operation 328. This operation modifies the associations between TIEs and nodes in the data structure if and as necessary to accommodate the changed position of the recently-moved TIE. It also updates the tangible user interface accordingly.

[0037] If the TUI controller detects that a TIE was moved from a TIE holder in the context collection to one in the MDP collection, it performs the operation 347, in which it checks whether that TIE was associated with the active node(s). If so, it performs a remove-active-node(s) operation 348 (described below). The TUI controller then performs the update interface operation 328 whereby the tangible user interface is updated to present the appropriate information for the newly active node(s). If the TIE that was moved was not associated with the active node(s), the TUI controller performs an operation 345 whereby it removes the node(s) associated with the recently-moved TIE from its record of the context set, if any.

[0038] If the TUI controller detects that a TIE was moved from a TIE holder in the context collection to one in the aggregate collection, it performs the operation 347, in which it checks whether that TIE was associated with the active node(s). If so, it performs the remove-active-node(s) operation 348. The TUI controller then performs the update interface operation 328, which modifies the associations between TIEs and nodes in the data structure if and as necessary to accommodate the changed position of the recently-moved TIE, and updates the tangible user interface accordingly. If the recently-moved TIE was not associated with the active node(s), the TUI controller performs the operation 345 whereby it removes the node(s) associated with the recently-moved TIE from its record of the context set, if any.

[0039] If the TUI controller detects that a TIE was moved from a TIE holder in the aggregate collection to one in the MDP collection, it performs the update interface operation 328, which modifies the associations between TIEs and nodes in the data structure if and as necessary to accommodate the changed position of the recently-moved TIE, and updates the tangible user interface accordingly.

[0040] If the TUI controller detects that a TIE was moved from a TIE holder in the aggregate collection to one in the context collection, it performs the operation 346, in which

it (a) marks the currently active node(s) as no longer active, and (b) marks the node(s) associated with the recently-moved TIE as the active node(s). It then performs the update interface operation **328**.

[0041] If the TUI controller detects that the TIE holder selector has changed which of the TIE holders in the context collection is designated as “selected,” it performs an operation **330** in which it checks if that TIE holder is connected to a TIE. If so, it performs an operation **332** in which it (a) marks the currently active node(s) as no longer being active, and (b) marks the node(s) associated with that TIE as the active nodes. It then performs the update interface operation **328**. If the selected TIE holder does not contain a TIE, the TUI controller ignores the selection.

[0042] **FIG. 3D** illustrates an exemplary embodiment of the remove-active-node(s) operation **351**. In an operation **368**, the TUI controller checks whether there is still at least one TIE in a TIE holder in the context collection. If so, it performs a choice operation **372**, in which it chooses one such TIE to represent the new active node(s). This choice can be made in a variety of ways. In some embodiments, this choice is based on order of connection to the context collection TIE holders. For instance, the TIE that was most recently connected to a TIE holder in the context collection might be chosen. In other embodiments, this choice is based on the physical position of the TIE. For instance, the TIE that is in the TIE holder that is furthest to the right in the context collection might be chosen. Still other embodiments make this choice in still other ways. The TUI controller then performs an operation **374** in which it marks the node(s) associated with the chosen TIE as the active node(s). If there is no TIE in a TIE holder in the context collection, the TUI controller performs an operation **370** wherein it marks the root node of the hierarchical data structure as the active node.

[0043] **FIG. 3E** illustrates an exemplary embodiment of the update interface operation **328**. The TUI controller performs an aggregate TIE choice operation **350**, in which it chooses a (possibly empty) set of TIEs in holders in the MDP collection or aggregate collection to be aggregate TIEs. If there are more nodes in the presentation set than there are TIEs in TIE holders in the MDP collection, then at least one TIE is guaranteed to be chosen as an aggregate TIE. TIEs in TIE holders in the aggregate collection are always chosen first;

additional TIEs may also be chosen from those in TIE holders in the MDP collection. The TUI controller then updates each TIE in a TIE holder in the MDP collection or aggregate collection as follows. First, the TUI controller associates the TIE with a node or set of nodes from the data structure being explored. In the preferred embodiment, this is done so as to ensure that, if possible, all the nodes in the presentation set are either (a) associated directly with a TIE or (b) elements of a set that is associated with a TIE. In an exemplary embodiment, this is done as follows. The TUI controller performs an operation 354 to check if the TIE is one of those chosen to be an aggregate one. If so, it performs a set association operation 358 that associates the TIE with a set of nodes. If not, it performs a node association operation 356 that associates the TIE with a single node.

[0044] Next, the TUI controller performs a data formatting operation 360, in which it converts the data corresponding to the associated nodes in the data structure into the form in which it will be displayed on the TIE's display elements. The TUI controller then performs a display operation 362 that causes the TIE's display elements to display this information.

[0045] Next, the TUI controller performs an alert-identification operation 364. In this operation, the TUI controller identifies the alert that should be presented by the TIE. This can be done in a variety of ways. Among the possibilities are: (a) computing the alert as a function of the data corresponding to the associated node(s); (b) computing the alert as a function of the data corresponding to the children of the associated node(s); and (c) computing the alert as a recursive function of the alerts that the nodes' children would present if they were explored. An example of this last possibility may be presented in an embodiment (described in detail below) in which a hierarchy of computer system components is being explored. In that embodiment, a TIE associated with high-level node that corresponds to a rack of servers might display a high alert if any of the sub-components in that rack has failed (and thus if a TIE associated with that sub-component would display a high alert), however many levels down the corresponding node may be in the hierarchy. Having identified the appropriate alert, the TUI controller performs an alert update operation 366 that causes the TIE's alert presentation elements to present it.

[0046] The operations of the update interface operation 328 have been described in sequential fashion by way of illustration. However, these operations could be reordered and/or carried out in parallel. For example, the operations in which the data are identified and displayed (operations 360 and 362) may be performed after, rather than before, those in which the alerts are (operations 364 and 366), or else executed in parallel with them. As another example, some or all of the TIEs could be updated in parallel, rather than sequentially.

[0047] FIG. 4 illustrates a tray-based embodiment of a tangible user interface (TUI) system for exploring hierarchical data structures. The TUI controller is implemented as a computer system. The three collections of TIE holders are contained in separate areas of a tray 402. Tray 402 can be implemented using a peripheral interface controller (PIC) based microcontroller unit (MCU) infrastructure. An MDP area 410 of the tray 402 contains an MDP collection of TIE holders arranged in a rectangular matrix; a context area 412 contains a context collection, arranged in a row; and an aggregate area 414 contains an aggregate collection arranged in a column. Other elements of the system are as described with reference to FIG. 1. In alternative embodiments, a tray is not used but rather TIEs communicate wirelessly without the need for a tray interface.

[0048] FIGS. 5 and 6 illustrate an exemplary embodiment used to monitor computer systems. FIG. 5 is a block diagram illustrating a configuration of the embodiment described in FIG. 1 for monitoring computer systems. A computer system 510 is a data source being monitored by a system monitor 512. The computer system 510 is viewed as a system tree, which is a hierarchical data structure. Nodes near the root of the tree correspond to the largest system components; nodes at successive levels correspond to sub-components of the components at the level above. For instance, a large computer system might comprise many network sites, each of which in turn comprises many racks of servers, each of which in turn comprises a number of individual servers, each of which comprises processing units, memory units, disk drives, and so on. The root node of that system's tree could correspond to the entire system, the nodes at its first level to the sites; the nodes at its second level to the racks of servers; the nodes at its third level to the individual servers; and the nodes at its final

level to the components of the individual servers. The exemplary embodiment can be used to explore such a system tree, and thus can help to administer such a computer system.

[0049] The system monitor **512** provides updates about events that affect the state of the computer system **510**. Such events might be, for example, the addition or removal of a system component, the failure of such a component, or a significant change in the demand for a system resource. The system monitor **512** might also run diagnostics to identify likely future events—for instance, by predicting when a current configuration is likely to lead to future component failure or resource exhaustion. The system monitor **512** notifies the TUI controller via the data-source link about some or all of these events. In an exemplary embodiment, these notifications take the form of a stream of XML messages.

[0050] In some embodiments, each TIE has at least one display element that is capable of displaying text. This element displays information about the component(s) corresponding to the node(s) associated with the TIE. It might, for example, display the components' names, their configurations, a history of changes to their configurations, and other helpful information. Each TIE also has alert presentation elements that can present at least three distinct alerts. In an exemplary embodiment, the TIEs glow a different color to present each of these alerts: green to represent proper operation, red to represent immediate failure or resource exhaustion, and yellow to represent anticipated future failure or resource exhaustion.

[0051] If the TIE is associated with a leaf node in the system tree, it presents the alert that best represents how the corresponding component is functioning. If the TIE is not associated with a leaf node in the system tree—either because it is associated with a node that has children or because it is an aggregate TIE—it computes what alert to display as a function of the alerts that its children or constituent nodes would display, were they linked to TIEs. Under an exemplary computation, such a TIE would present the most serious alert that any of its children or constituent nodes would. Thus, it would glow green only if all the relevant child or constituent nodes would do likewise; it would glow yellow only if (a) at least one of the relevant child or constituent nodes would do the same, and (b) none of these nodes would glow red; otherwise, it would glow red. The three collections of TIE holders may be incorporated into a single tray as in the exemplary embodiment described above.

[0052] The user can use the above described configuration to monitor and administer the computer system **510**. As illustrated in **FIG. 6A**, the user can monitor the overall status of the computer system **510** at a glance. When the computer system **510** is working properly, the tangible user interface glows a uniform green. When an immediate or anticipated failure event occurs in any part of the tree, the system monitor **512** will detect the failure event and inform the TUI controller. As illustrated in **FIG. 6B**, the TUI controller updates the interface so that a TIE **602** associated with the highest level component that includes the affected component—in the example system described, this would be the site that includes the component—glows red or yellow, alerting the user to the problem. The user then interacts with the interface to explore the system tree and locate the fault. As illustrated in **FIGS. 6C** and **6D**, at each level, the TUI associated with the element that contains the faulty one glows red or yellow. The user can follow this trail of red and yellow glowing TIEs by moving them to the context area one by one. At the end of this trail, the user finds a TIE **606** associated with the faulty component, and can take steps to correct the problem. In some embodiments, the user can remove this TIE **606** from the tangible user interface and have it preserve its current display and alerts. It can then be physically transferred to another location, where it can either provide information about the problem, or, in some embodiments, act as a security token that temporarily gives permission to perform actions that will correct the problem. Once the problem has been corrected, the user can retrace her steps through the tree by moving the TIEs from the context area back to the MDP area in reverse order.

[0053] **FIG. 7** illustrates still another example embodiment. In this embodiment, another collection of TIE holders is included with the embodiment described with reference to **FIG. 1**. This additional collection is called a communication collection **717**. A plurality of TIE holders **716** in the communication collection **717** is used as a drop box for communication with other users. Each TIE holder **716** in this collection is associated with another user or set of users. Connecting a TIE to one of these TIE holders **716** notifies that user or users of the associated node. This notification may be either by an electronic data communication method, for instance email or pager, or by way of the system, via an alert or message presented on some element of those users' tangible user interfaces.

[0054] As an example, a virtual embodiment is also possible, in which some or all of the elements described are objects in a virtual display. The user might then interact with these objects in any of the ways that users interact with computer systems, including by dragging and dropping them with a mouse, touchpad, touch screen, keyboard, stylus, or other computer input mechanism. In such embodiments, the configuration and placement of the TUI holders in the collections is particularly flexible. For example, the MDP collection might be laid out to correspond to a logical layout of the current presentation set, which might change each time the presentation set does. In addition, virtual TIEs can easily be transmitted to other systems and users in any way that electronic data can be transferred, including over a network, or via physical storage media. A hybrid system including both virtual and physical aspects is also possible.

[0055] The foregoing description of embodiments of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the embodiments disclosed; modifications and variations are possible in light of the above teachings or may be acquired from practice of the invention. The embodiments were chosen to explain the principles of the invention and its practical application to enable one skilled in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. By way of example, an embodiment is possible wherein there are no TIE holders, or where the TUI controller uses other means, such as Radio-frequency Identification (RFID), to decide whether to treat each TIE as being attached to the MDP area, the context area, or the aggregate area. By way of another example, the invention may be embodied without an explicit aggregate area. In such an embodiment, the only aggregate TIEs used may be those the TUI controller chooses as such.

CLAIMS

WHAT IS CLAIMED IS:

- 1 1. An apparatus providing a tangible user interface for monitoring operations of a
2 plurality of components in a multi-component, multi-level system, the apparatus comprising:
3 a plurality of interface elements, the interface elements providing indications
4 of status for a plurality of monitored sources arranged in a hierarchy; and
5 a plurality of interface element holders, the interface element holders
6 providing for a change in hierarchy levels for which indications of status are presented by the
7 plurality of interface elements.
- 1 2. The apparatus of Claim 1 wherein the plurality of interface elements
2 comprise cubes equipped with a liquid crystal display (LCD) and with light emitting diodes
3 (LEDs).
- 1 3. The apparatus of Claim 1 wherein the plurality of interface element
2 holders comprise locations on a tray, including a main area, a context area, and an aggregate
3 area.
- 1 4. The apparatus of Claim 1 wherein indications of status are
2 implemented using color indicators.
- 1 5. The apparatus of Claim 4 wherein the color indicators comprise red,
2 yellow, and green.
- 1 6. The apparatus of Claim 1 wherein the hierarchy levels comprise a rack
2 level, a server level, and a component level.
- 1 7. The apparatus of Claim 1 wherein the plurality of interface elements
2 are removable from the plurality of interface element holders.

1 8. The apparatus of Claim 7 wherein the plurality of interface elements
2 present the indication of status even when not held by one of the plurality of interface
3 holders, whereby one or more of the plurality of interface elements can be taken to where one
4 or more of the plurality of monitored sources causing a fault status is located.

1 9. The apparatus of Claim 1 wherein the indications of status for a
2 plurality of monitored sources are based on a plurality of data feeds from the monitored
3 sources.

1 10. A system which provides monitoring of a plurality of components
2 which are part of a multi-level, multi-component system, the monitoring system comprising:
3 a plurality of data sources;
4 a server configured to receive data from the plurality of data sources;
5 a plurality of monitoring cubes providing status indications for the plurality of
6 data sources based on data communicated to the server; and
7 a tray communicatively coupled to the server and including a plurality of
8 sections on which the plurality of monitoring cubes are placed, the plurality of monitoring
9 cubes obtaining different information depending on placement in one of the plurality of
10 sections.

1 11. The system of Claim 10 wherein the plurality of sections comprise a
2 top area, a main area, and an aggregate area, wherein the top area defines a depth of a
3 hierarchical tree based on a number of monitoring cubes in the top area, wherein the main
4 area holds monitoring cubes presenting status indications for the plurality of data sources at a
5 currently selected level of the hierarchical tree, and wherein the aggregate area provides a
6 location for one or more monitoring cubes holding information for more than one of the
7 plurality of data sources.

1 12. The system of Claim 10 wherein the plurality of monitoring cubes
2 comprise red, yellow, and green light emitting diodes (LEDs) for displaying status
3 information for the plurality of data sources.

1 13. The system of Claim 10 wherein the plurality of monitoring cubes
2 comprise displays providing information about the plurality of data sources.

1 14. The system of Claim 10 wherein the plurality of monitoring cubes
2 comprise a plurality of power sources, whereby each of the plurality of monitoring cubes are
3 removable from the tray while maintaining power.

1 15. A method of monitoring a plurality of components which are part of a
2 multi-level, multi-component system, the method comprising:

- 3 (a) receiving data from a system component;
- 4 (b) directing the received data to a monitoring cube corresponding to a
5 physical location and a hierarchy tier location of the system component;
- 6 (c) presenting a status indication at the monitoring cube based on the
7 received data and corresponding to status of the system component; and
- 8 (d) displaying information about the system component.

1 16. The method of Claim 15 wherein directing the received data to a
2 monitoring cube corresponding to a physical location and a hierarchy tier location of the
3 system component comprises identifying the hierarch tier, including one of a rack tier, a
4 server tier, and a component tier.

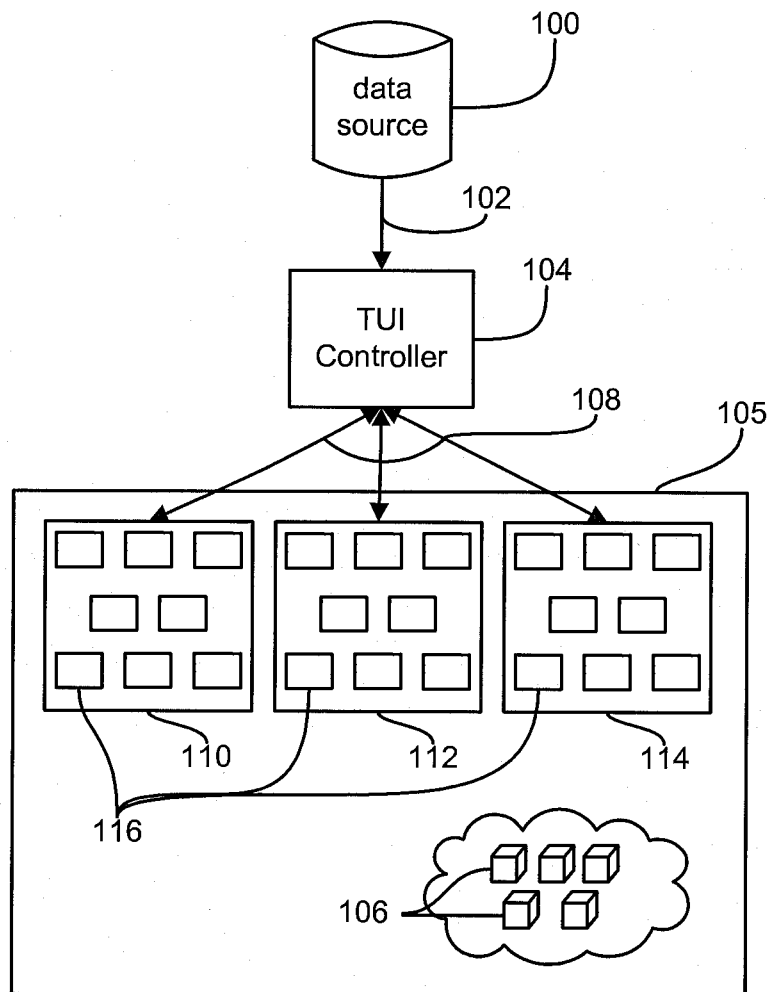
1 17. The method of Claim 15 wherein presenting a status indication at the
2 monitoring cube based on the received data and corresponding to status of the system
3 component comprises lighting one of a red, yellow, or green light emitting diode (LED).

1 18. The method of Claim 15 further including detecting location of a
2 plurality of monitoring cubes on a tray, the tray having a plurality of sections.

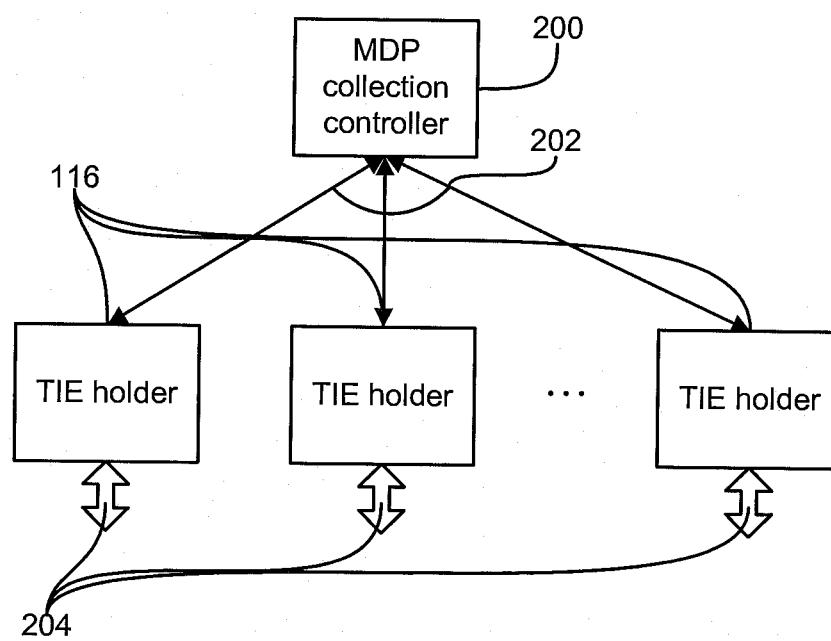
1 19. The method of Claim 17, wherein the plurality of sections comprise a
2 top area, a main area, and an aggregate area, wherein the top area defines a depth of a
3 hierarchical tree based on a number of monitoring cubes in the top area, wherein the main
4 area holds monitoring cubes presenting status indications for the plurality of data sources at a
5 currently selected level of the hierarchical tree, and wherein the aggregate area provides a
6 location for one or more monitoring cubes holding information for more than one of the
7 plurality of data sources.

1 20. The method of Claim 19 further comprising using the location of the
2 plurality of monitoring cubes on the tray in the directing the received data to the monitoring
3 cube corresponding to the physical location and the hierarchy tier location of the system
4 component.

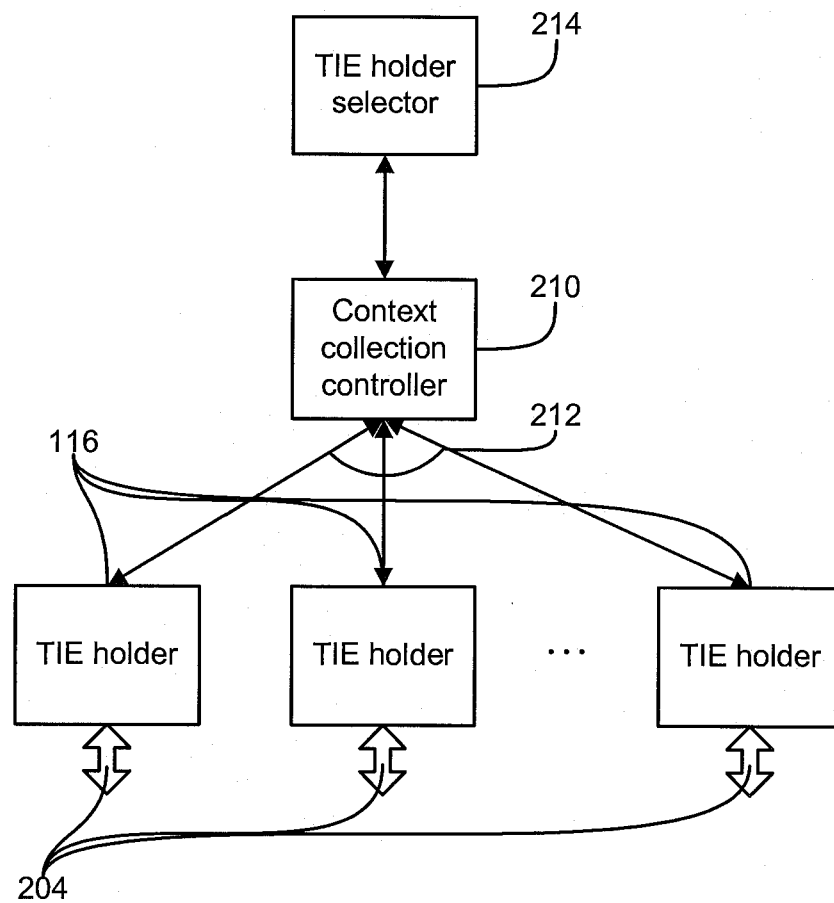
1/15

**FIG. 1**

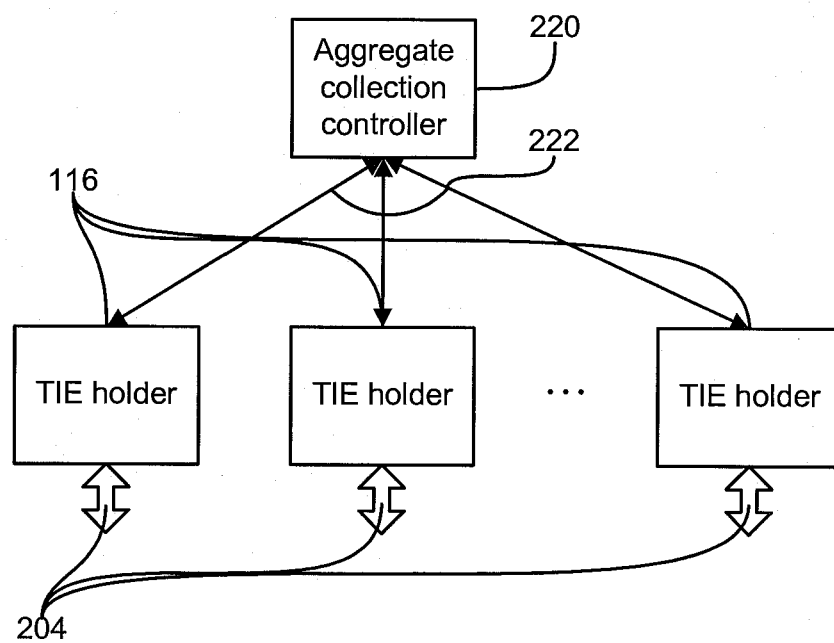
2/15

**FIG. 2A**

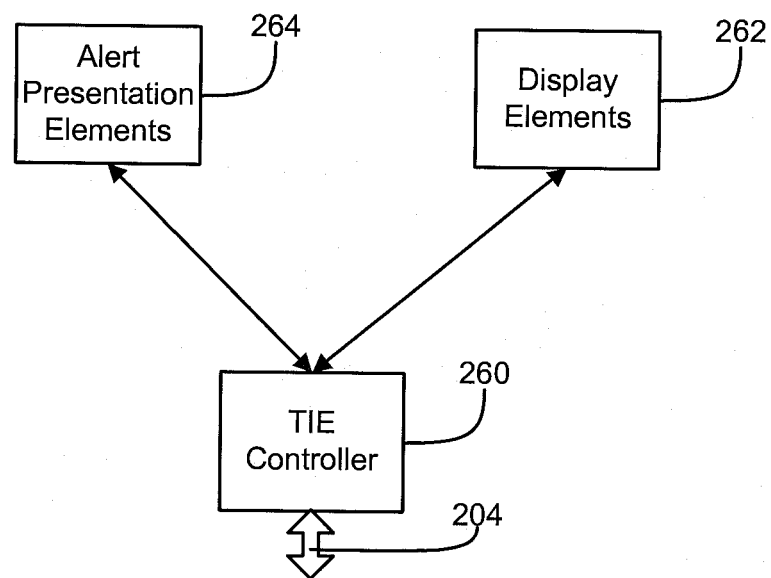
3/15

**FIG. 2B**

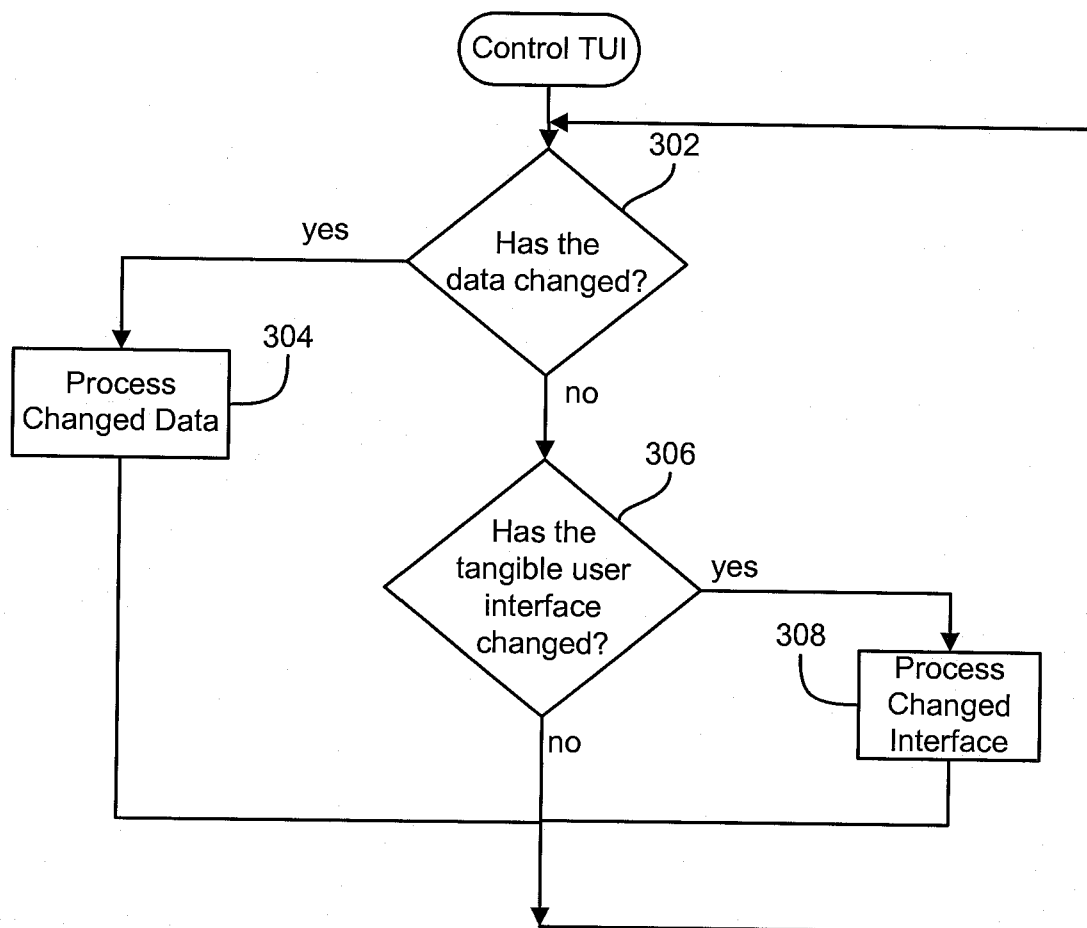
4/15

**FIG. 2C**

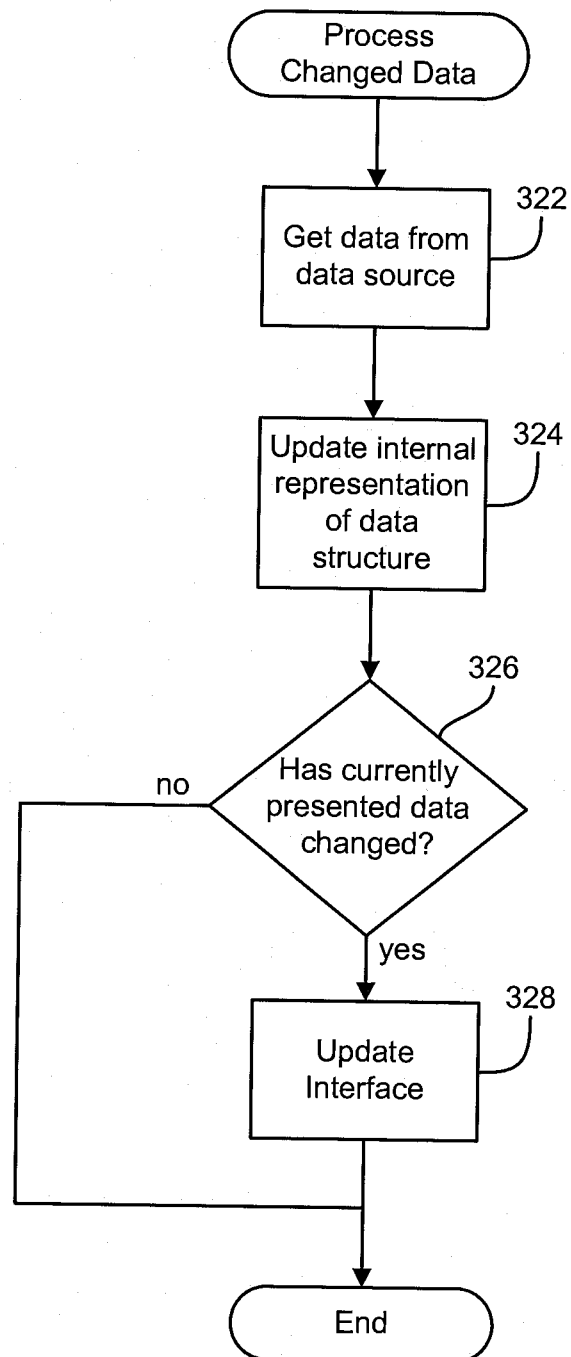
5/15

**FIG. 2D**

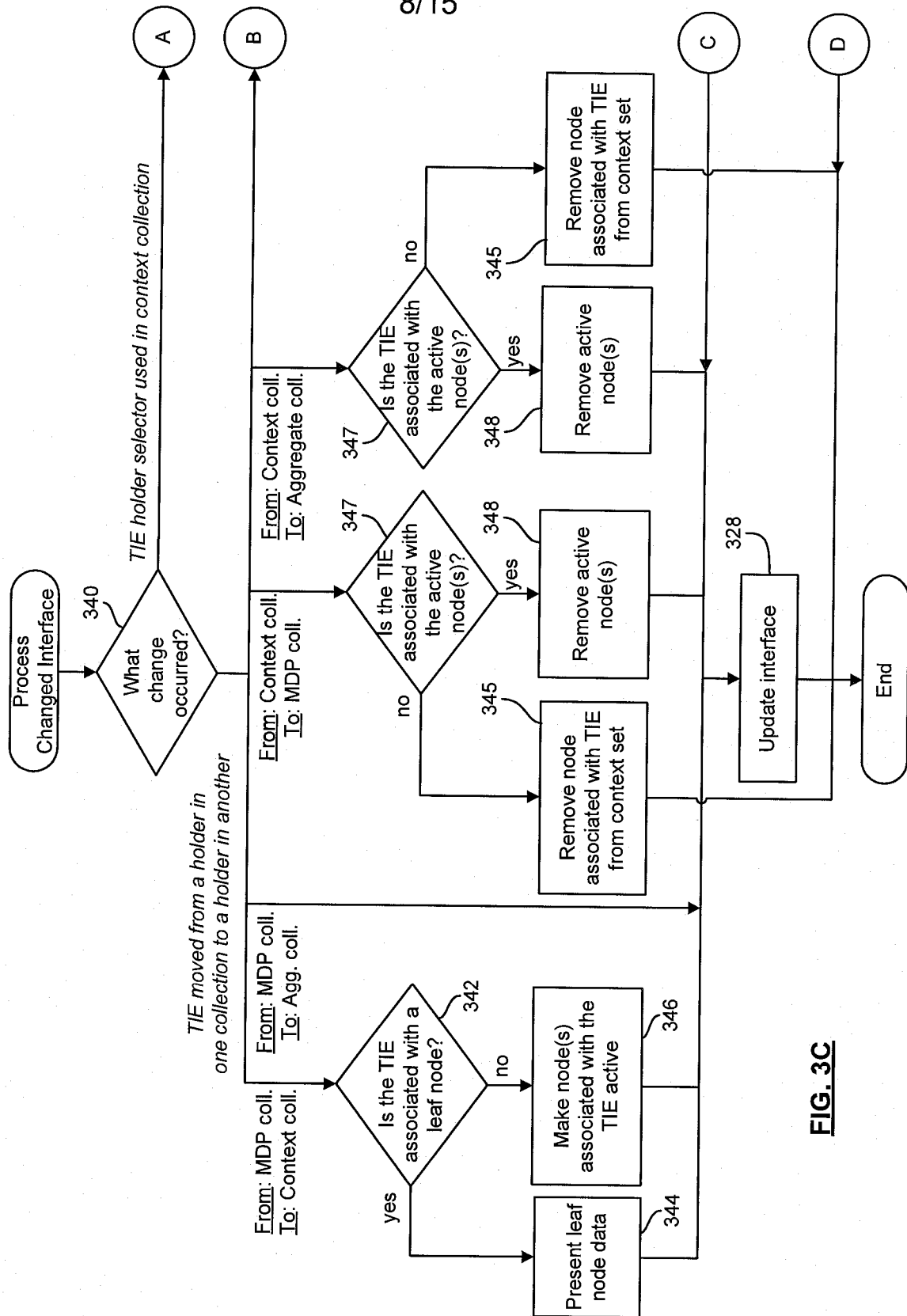
6/15

**FIG. 3A**

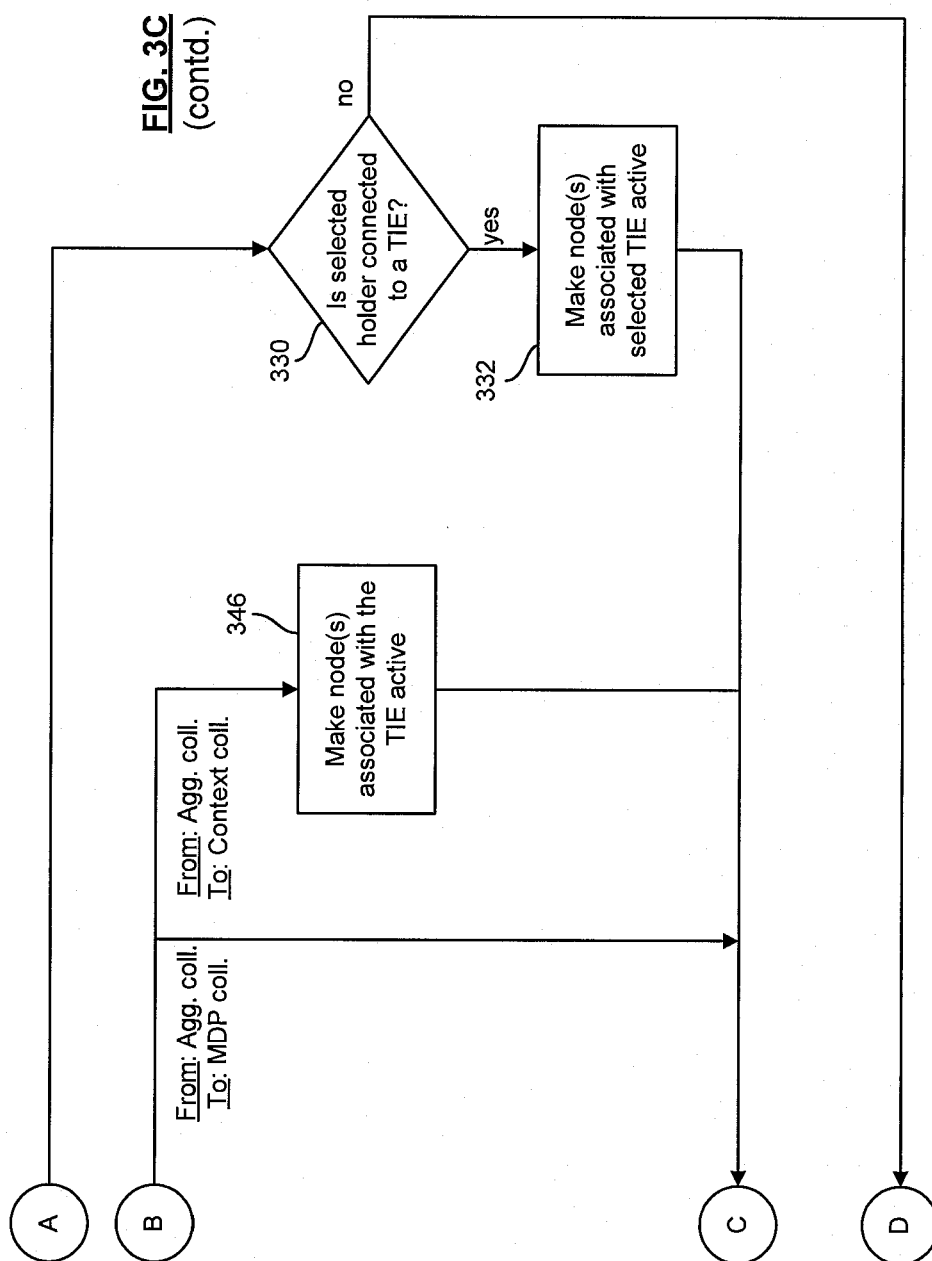
7/15

**FIG. 3B**

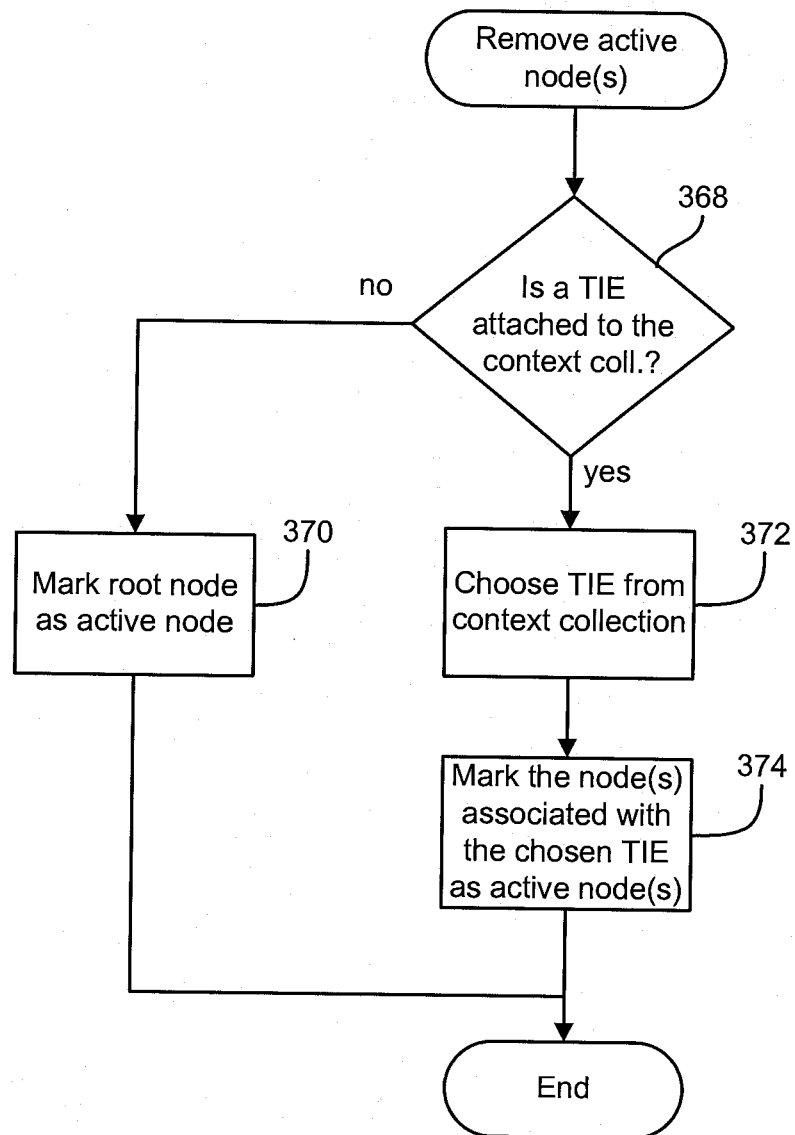
8/15



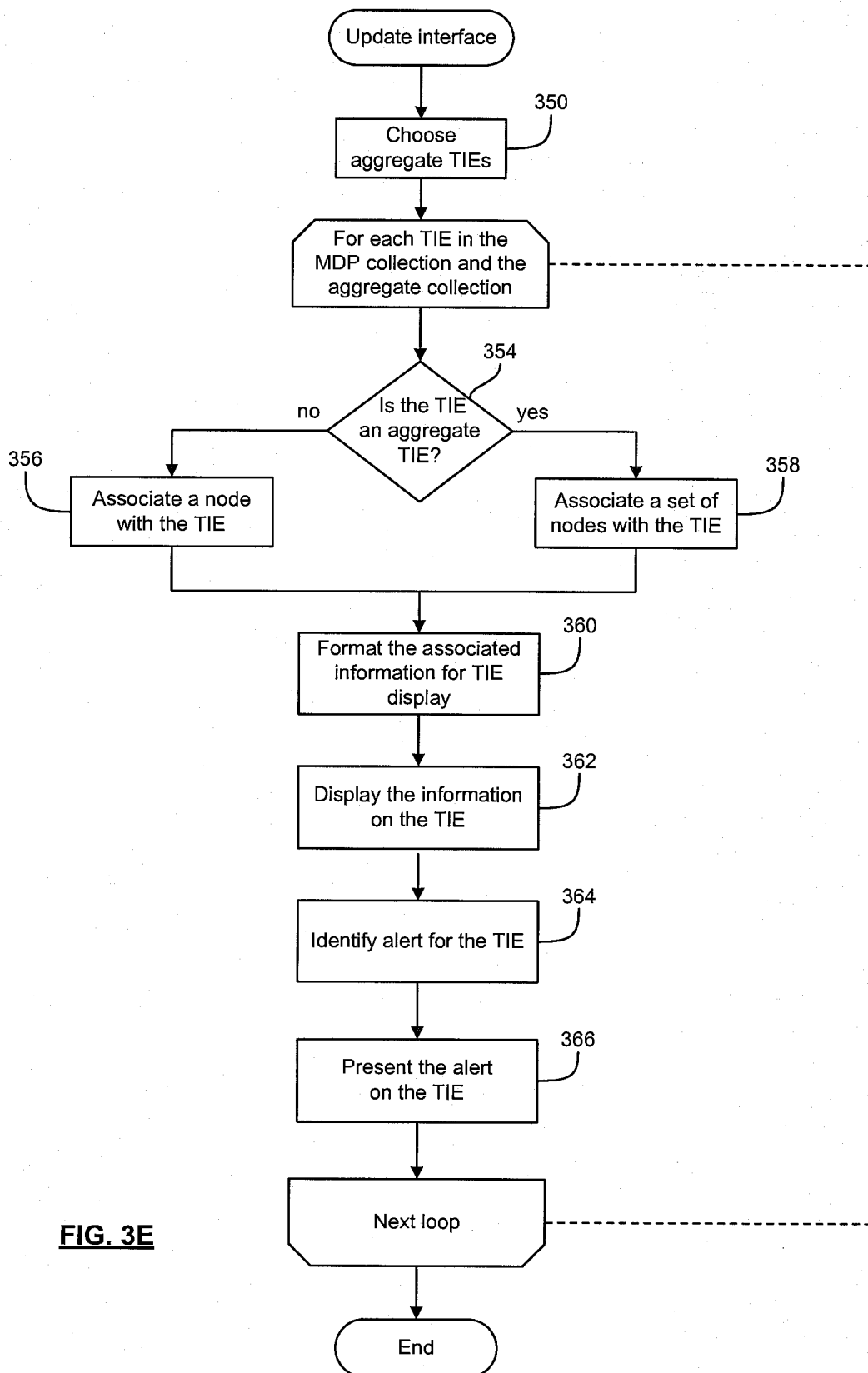
9/15



10/15

**FIG. 3D**

11/15

**FIG. 3E**

12/15

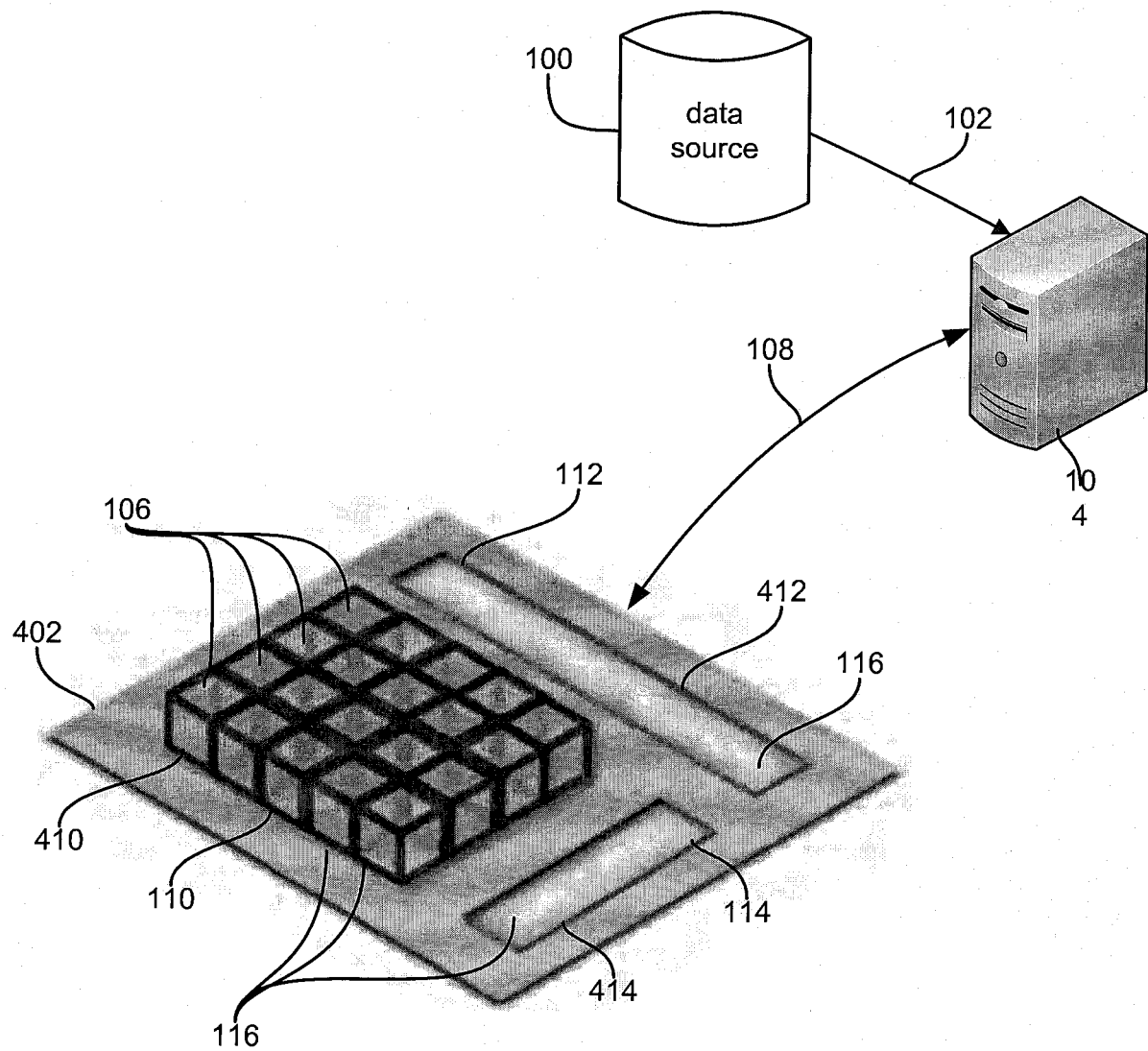
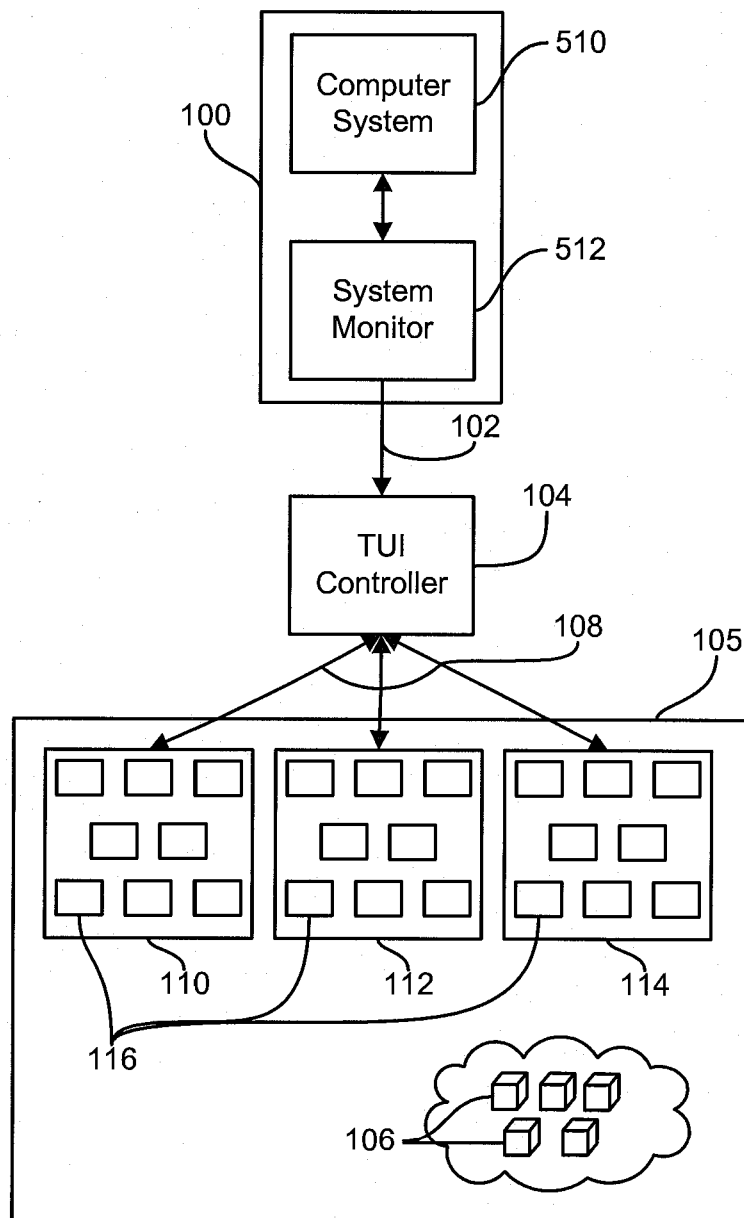


FIG. 4

13/15

**FIG. 5**

14/15

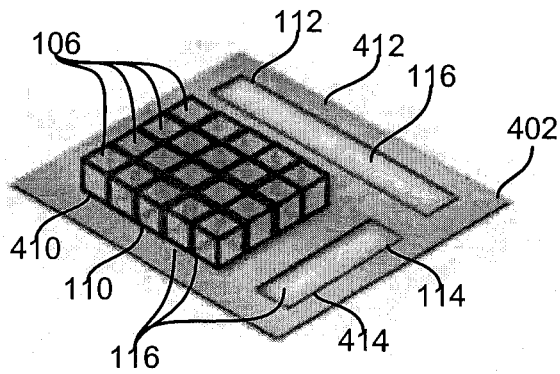


FIG. 6A

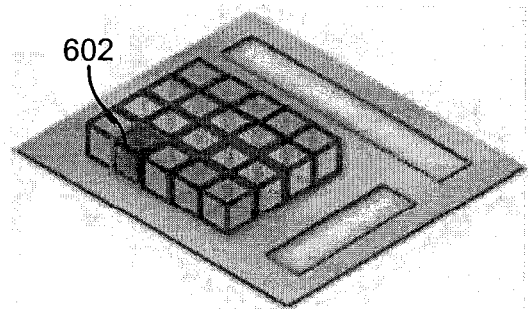


FIG. 6B

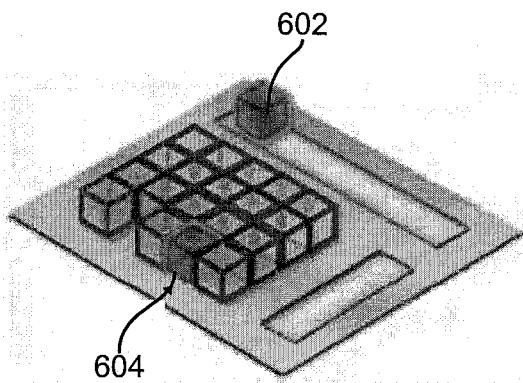


FIG. 6C

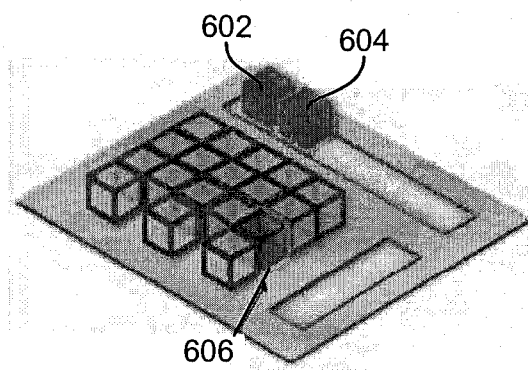
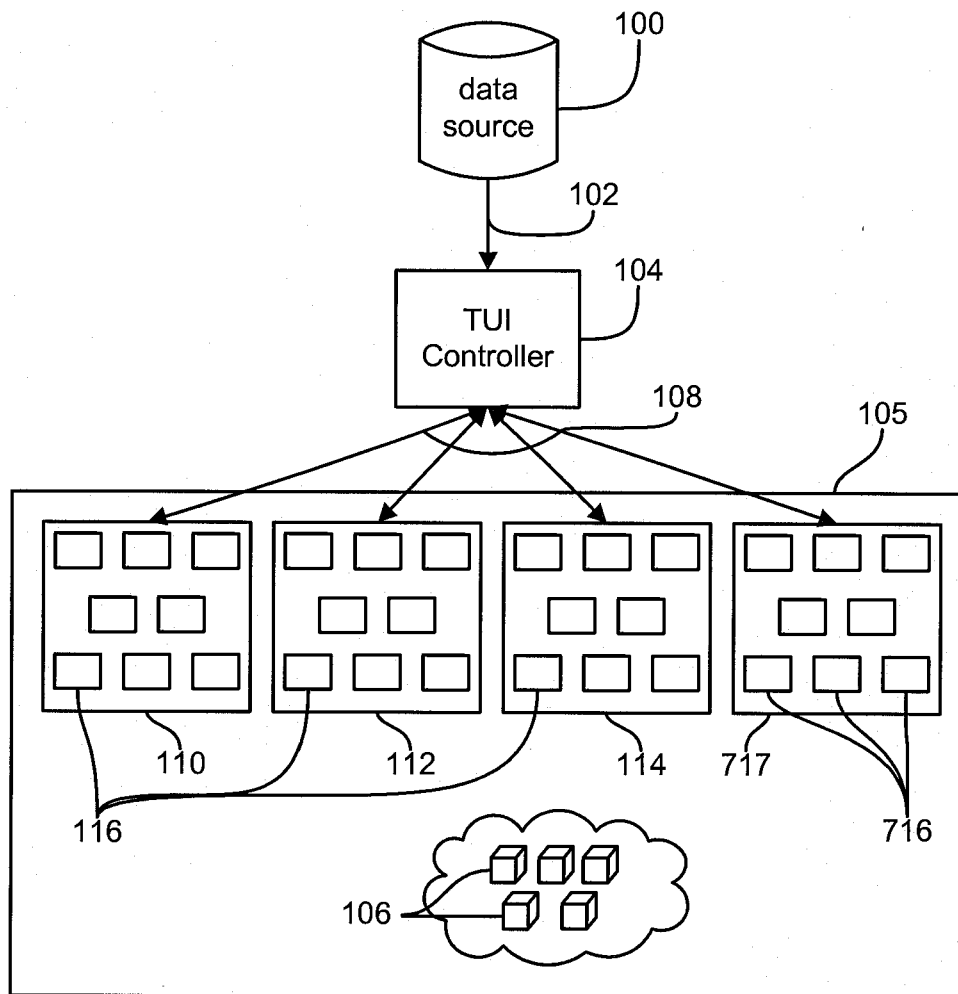


FIG. 6D

15/15

**FIG. 7**