(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 August 2016 (04.08.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/122676 A1

(51) International Patent Classification:
G06F 11/32 (2006.01)

(21) International Application Number:
PCT/US2015/014009

(22) International Filing Date:
31 January 2015 (31.01.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventor: **WILKINSON, Lyle Eric**; 1501 Page Mill Rd., Palo Alto, California 94304 (US).

(74) Agents: **ADEKUNLE, Olaolu** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79 , Fort Collins, CO 80528 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, 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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

[Continued on next page]

(54) Title: DETERMINATION OF STATUS BASED ON STATUS DOMAIN

(57) Abstract: An example method includes receiving signals indicative of a status value of each of a plurality of components in a computer system; determining a status of the computer system based on the status values of selected ones of the components, the selected ones of the components being selected based on a specified status domain and a topological map of the computer system, the topological map being indicative of inter-relationships among the plurality of components; and presenting the status of the computer system to a user.

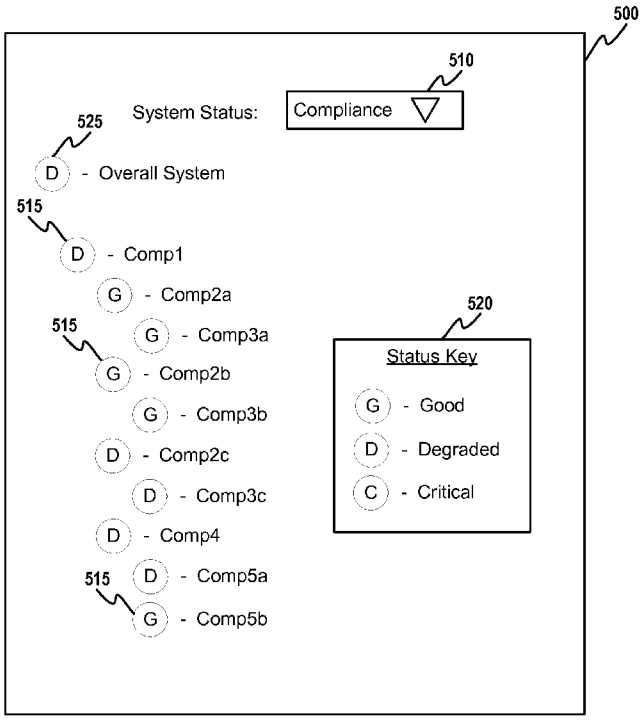


FIG. 5

WO 2016/122676 A1

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

DETERMINATION OF STATUS BASED ON STATUS DOMAIN

BACKGROUND

[0001] Status of a system can often be a critical feature to monitor. For example, an administrator may wish to know the status of a system before a major operation is to be run, such as a system upgrade. In other examples, an administrator may wish to regularly or continuously monitor the status of the system for health or usage of the system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] For a more complete understanding of various examples, reference is now made to the following description taken in connection with the accompanying drawings in which:

[0003] Figure 1 illustrates an example topological map of a computer system;

[0004] Figure 2 illustrates another example topological map of the computer system;

[0005] Figure 3 illustrates an example computing device that may utilize an example system status tracker;

[0006] Figure 4 illustrates an example flow diagram for an example process for determining status of a computer system;

[0007] Figure 5 illustrates an example status display for a compliance status domain of various components of a computer system; and

[0008] Figure 6 illustrates an example status display for a health status domain for various components of a computer system.

DETAILED DESCRIPTION

[0009] Example systems and methods described herein may utilize a system configuration template to obtain an overall system status measure by bubbling up status values from lower-level components. A system configuration template may include several topologies for describing different aspects, or status domains, of the system. These topologies may describe one or more system characteristics such as, for example, component dependencies, component redundancies, and other inter-relationships. Embedded within each of the topologies may be approaches to bubbling up the status in a particular domain (e.g. system health, system compliance, performance, or an administrator-defined domain). This generalized approach

enables presentation of status of components at any desired level in each status domain for which there is a topology, as determined by the system configuration template. In addition, overall system level status may be displayed for each status domain. Inquiry-based drill-down within each status domain, using the corresponding topology, may allow for status of any component to be displayed, along with a reason for less than perfect status.

[0010] Traditional methods for determining computing system status typically consider just a single domain and utilize a single, fixed algorithm, e.g. status of a health domain utilizing a worst of child components' status values. However, using a single status domain (e.g., health status) with a single algorithm (e.g., worst of child components) can result in difficulty in detecting a component failure or degradation. In addition, if and when the failure or degradation is identified, determining any remedy and recovering from the failure may also be difficult and take an indeterminate amount of time due to the uncertainty of identifying the failure and then applying the remedy.

[0011] Thus, it may be difficult to adequately represent the status of an overall system using a single overall status domain measure, especially for a complex computing system. However, status measures from multiple different perspectives, or status domains, may make the problem tractable. In addition, by allowing a system administrator to select views for different status domains, which may display status values at a top level or any lower level in a status domain-specific hierarchy, presents more meaningful information to that administrator. Allowing different algorithms to be used to derive the status measures for different status domains ensures that more meaningful and potentially actionable status indicators can be presented to the administrator.

[0012] Topological maps may be represented via textual, e.g. JSON or XML, means, to define a system of multiple components (e.g., servers, storage, networking, firmware, software) as described herein. The topological maps may include references from one component to one or more other components for multiple different status domain-specific hierarchies. Specific algorithms for bubbling-up of status values from lower level components allow for status values to be determined for the overall system as well as for lower level components such that status values may be obtained at any level. The algorithms may be embedded in the management system or may be external/user-defined algorithms. Status of individual components may be gathered by the management software as input for these algorithms in order to bubble-up the

status to upper layers and finally to the overall system. Status at any level in these multiple hierarchies, including the overall system status for a particular status domain, may be represented to a system administrator to allow for a more detailed identification of a problem and to allow for a faster identification of a remedy.

[0013] Referring now to the figures, Figure 1 illustrates an example topological map 100 of a computer system including dependencies, redundancies and other inter-relationships among a plurality of components for determining status of the computer system in a first status domain. By way of example only, the topological map 100 will be described as pertaining to a compliance domain. The status value of components, and thus the overall system, in the compliance domain may depend on factors such as whether the version of software or firmware of the various components are up to date, i.e., comply, with the most recent requirements. Hardware compliance may be based on a range of model numbers or serial numbers that satisfy requirements.

[0014] The topological map 100 of the first compliance domain may be stored in memory including descriptive parameters such as, by way of example, an identifier of a component, such as a unique identifier, name, serial number, etc.; functioning or type of that component, including a purpose of that component within the system; dependency of each component relative to other components in the system; whether the component in the system is redundant to another component in the system; and/or mechanism in which to perform an upgrade to the identified component, such as a link to download upgrades to software, etc.

[0015] For example, a memory storing data indicative of the topological map 100 of a computer system may include topological map data indicative of device types of a plurality of components (COMP1, COMP2A, COMP2B, COMP2C, COMP3A, COMP3B, COMP3C, COMP4, COMP5A and COMP5B in the example topological map 100) of the computer system and inter-relationships among the plurality of components. In Figure 1, the inter-relationships include first dependencies 110 between the top level component COMP1 and four second level components COMP2A, COMP2B, COMP2C and COMP4, and second dependencies 115 between the second level components and third level components COMP3A, COMP3B, COMP3C, COMP5A and COMP5B.

[0016] In addition to the first and second dependencies 110 and 115, the topological map interrelationship data further includes redundancy information identifying components that are

capable of performing the same function and have the same dependency relative to another computing device. For example, dashed line 120 indicates that the components COMP2A, COMP2B and COMP2C are redundant components in relationship to COMP1 in the compliance domain. Redundant components may be treated differently than non-redundant components in the way that their individual status values affect the status of components with which they have a dependency relationship. For example, if one or two of the COMP2A-COMP2C components has a failed status value in the compliance domain, the status value of COMP1 may only be degraded since a third redundant component that is in compliance may perform the necessary functions. As will be discussed below, the way in which status values of lower level components are bubbled-up to upper level components, as determined in part by the dependencies and redundancies of the topological map data, is controlled by one of various algorithms.

[0017] Figure 2 illustrates another example topological map 200 of the same computer system as Figure 1, but including different dependencies, redundancies and other inter-relationships among the plurality of components than those illustrated in Figure 1 for determining status of the computer system in a second status domain. By way of example only, the second topological map 200 will be described as pertaining to a health domain. The status value of components, and thus the overall system, in the health domain may depend on whether the various components are operating in a fully up state, in a degraded state, in a failed state, in a scheduled down state (e.g., for scheduled upgrades), etc.

[0018] The topological map 200 of the health domain may be stored in memory including descriptive parameters such as, by way of example, an identifier of a component, such as a unique identifier, name, serial number, etc.; functioning or type of that component, including a purpose of that component within the system; dependency of each component relative to other components in the system; whether the component in the system is redundant to another component in the system; and/or mechanism in which to perform an upgrade to the identified component, such as a link to download upgrades to software, etc.

[0019] As seen in Figures 1 and 2, the components are the same. However, the interrelationships describing the dependencies, and possibly the inter-relationships describing redundancies, are different. In Figure 2, the inter-relationships of the COMP1 component include first dependencies 210 between the top level component COMP1 and one second level component COMP4, and three third level components COMP3A, COMP3B and COMP3C. The

inter-relationships also include second dependencies 215 between the second level components COMP2A, COMP2B, COMP2C and COMP4 and third level components COMP3A, COMP3B, COMP3C, COMP5A and COMP5B. As can be seen, the health domain topological map 200 differs from the compliance domain topological map 100 in that the COMP1 component is not directly dependent on the COMP2A, COMP2B and COMP2C components. For this reason, the status values of the lower level components may bubble-up differently to the COMP1 component and thus affect the status value of the overall system in a different way. As will be discussed below, the way in which status values of lower level components are bubbled-up to upper level components, as determined in part by the dependencies and redundancies of the topological map data, is controlled by one of various algorithms.

[0020] Referring now to Figure 3, a computing device 300 that may utilize an example system status tracker 330 for determining computer system status in the first and second status domains of Figures 1 and 2, as well as other status domains such as a performance domain or administrator-defined status domains, is illustrated. The example computing device 300 may include embedded firmware and hardware components in order to continually collect operational and event data from the components from a computer system such as the components illustrated in Figures 1 and 2. The example computing device 300 may be a standalone server such as a blade server, a storage server or a switch, for example. The example computing device 100 may include a server CPU (central processing unit) 310, at least one memory device 320, and a power supply 340. The power supply 340 is coupled to an electrical interface 345 that is coupled to an external power supply such as an AC power supply 350.

[0021] The computing device 300 also includes an operating system component 355 including, for example, an operating system driver component and a pre-boot BIOS (Basic Input/Output System) component stored in ROM (read only memory), and coupled to the CPU 120. In various examples, the CPU 120 may have a non-transitory memory device 320. In various examples, the memory device 320 may be integrally formed with the CPU 120 or may be an external memory device. The memory device 320 may include program code that may be executed by the CPU 120. For example, one or more processes may be performed to execute a user control interface 375 and/or software applications 380. The memory device 320 also stores data 322 indicative of topological maps, data 324 indicative of status algorithms and other parameter data 326.

[0022] The topological map data 322 includes data indicative of topological maps of the computer system that the computing device 300 is a part of or is monitoring. The topological map data 322 may be indicative of device types of a plurality of components in the computer system and inter-relationships among the plurality of components. As described above, the inter-relationships include at least first dependencies of the computing device with one or more first components of the plurality of components, and second dependencies of first components with one or more second component of the plurality of components. For example, in order to represent the first topological map 100 of Figure 1, the topological map data 322 would include data indicative of device types of the components COMP1, COMP2A, COMP2B, COMP2C, COMP3A, COMP3B, COMP3C, COMP4, COMP5A and COMP5B and data indicative of the inter-relationships including the first dependencies 110 between the top level component COMP1 and four second level components COMP2A, COMP2B, COMP2C and COMP4, and the second dependencies 115 between the second level components and third level components COMP3A, COMP3B, COMP3C, COMP5A and COMP5B. Although the example topological map 100 includes only three component layers and first and second dependencies between components, other topological maps may include many more layers of components and many more numbers of dependencies. In some examples, topological map data 322 may be defined by a user.

[0023] The topological map data 322 may also include redundancy information identifying components of the plurality of components each capable of performing the same function and each having the same dependency relative to the computing device.

[0024] The topology map data 322 may include topology map code. An example of topology code that may be used to model the compliance domain topology map 100 of Figure 1 and the health domain topological map 200 of Figure 2 is presented below:

Sample Topology Code

```
Topology : "Compl";
Component  : "Compl"
{
    Status Domain "Compliance"
    {
            Algorithm "Worst";
            Children : "Comp2a", "Comp2b", "Comp2c", "Comp4";
```

```
    }
    Status Domain "Health"
    {
        Algorithm "Average";
        Children : "Comp3a", "Comp3b", "Comp3c", "Comp4";
    }
}
Component : "Comp4"
{
    Status Domain : “Compliance”
    {
        Algorithm : “UserDefined1”
        {
            URI : "http://example.com:8088/rest/userdef1";
        }
        Children : "Comp3a", "Comp3b", "Comp3c", "Comp4";
    }
    Status Domain "Health";
    {
        Algorithm "Inherent";
    }
}
Component : "Comp3a"
{
    Status Domain : "Compliance"
    {
        Algorithm : "Inherent";
    }
    Status Domain : "Health";
    {
        Algorithm "Inherent";
    }
}
```

[0025] In this example code, the status of the COMP1 component in the compliance domain, and hence the overall system in this example, is determined using a worst of child algorithm that depends on the status values of the children COMP2A, COMP2B, COMP2C and COMP4. The status of the COMP1 component in the health domain, and hence the overall system in this example, is determined using an averaging algorithm that depends on the status values of the children COMP3A, COMP3B, COMP3C and COMP4. The status of the COMP2A, COMP2B, COMP2C, COMP3A, COMP3B and COMP3C components may be an inherent status algorithm

(as shown for COMP3A only in the above code), or another type of algorithm that may also take into account the status of other underlying components, not shown.

[0026] The status algorithm data 324 includes data indicative of algorithms used by the system status tracker 330 for determining status of the computer system, and components of the computer system, based on status of components in all levels of the computer system. The status of lower layer components may also depend on algorithms dependent on other lower level components. The lower level algorithms may be implemented in the same status algorithm data 324 of the computing device 300, or may be implemented in the lower level components themselves, where the lower level components simply communicate status values, along with supporting data such as device type identifier, reason for any abnormal status values, etc., to the computing device 300.

[0027] The status algorithm for an underlying component may include a status inherent to the underlying component plus the status of other underlying components within that particular topological map. An inherent status may be determined via some programmatic means from the system of the underlying component. The status algorithms that are stored in the status algorithm data 324 may include worst of underlying components algorithms, average of underlying components algorithms, highest of underlying components algorithms, user-defined algorithms and time varying algorithms (e.g., dependent on the time of day, day of week, etc.). The user-defined algorithms may be obtained via the user control interface 375 configured to receive input representative of a user specified algorithm with which to determine the status value of the computer system.

[0028] The other parameter data 326 may include status values that have been determined by the system status tracker 330 for the overall computer system and the individual components. The example system status tracker 330 may perform acquisition, logging, file management, time-stamping, and storing of status value data in the other parameter data 326. In order to optimize the amount of actual data stored, the system status tracker 330 may apply filter, hash, tokenization, and delta functions on the data acquired prior to storing the information to the non-volatile memory 320.

[0029] The system status tracker 330 may be implemented in software, firmware and/or hardware. The system status tracker 330 receives control signals from a plurality of components in the computer system, the control signals being associated with a status value of respective

ones of the plurality of components with respect to at least one status domain. The status domains may include, but are not limited to, a health domain representative of availability level of a component, a compliance domain representative of whether versions of software and or firmware of a component comply with current requirements, a performance domain representative of a level at which a component is functioning relative to a normal level, and an administrator-defined domain.

[0030] Upon receiving the control signals associated with the status values of the underlying components, the system status tracker 330 determines a status value of the computer system (and the status values of underlying components) based on algorithms stored in the status algorithm data 324, the inter-relationships of the topological maps stored in the topological map data 322, and further based on the received control signals associated with the status values of the computer system components. Upon determining the status values, the system status tracker 330 stores data indicative of the status values of the respective ones of the plurality of components and data indicative of the determined status value of the computer system in the memory 320, e.g., in the other parameters data 326.

[0031] The control signals received by the system status tracker 330 may include the actual status values of the components of the computer system, or data that may enable the system status tracker 330 to utilize algorithms to calculate the status values of the components of the computer system. For example, the system status tracker 330 may determine the status value of computing device 300 (this being the overall computer system status in the example topological maps 100 and 200 of Figures 1 and 2) based on an algorithm associated with component COMP4 and inter-relationships, e.g., dependencies, of the components COMP5A and COMP5B, both having a dependency with the component COMP4. The system status tracker combines this derived status value of COMP4 with other status values of other components in the computer system in order to determine the status value of the overall computer system.

[0032] The computing device 300 may also include a display 360, a network interface 365 and other hardware 370 known to those skilled in the art. The network interface 365 may be coupled to a network such as an intranet, a local area network (LAN), a wireless local area network (WLAN), the Internet, etc., where the other underlying components are a part of the network or at least coupled to the network

[0033] The display 360 may be used to display the determined status values of the overall computer system as well as the status values of the individual underlying components to an administrator or other user such that problems may be easily identified and remedied. For example, the user control interface 375 may be configured to receive commands from a user or administrator inquiring about the status of specified ones of the plurality of components. In response to the commands, the system status tracker 330 retrieves data stored in the memory 320, e.g., in the other parameters data 326, indicative of the status of the other components and communicates the retrieved data to be displayed on the display 360.

[0034] Figure 4 illustrates an example flow diagram for a process 400 for determining status of a computer system. The process 400 is exemplary only and may be modified. The example process 400 of Figure 4 will now be described with further references to Figures 1, 2 and 3.

[0035] In various examples, the system status tracker 330 may receive, via the user control interface 375, input representative of a user specified algorithm with which to determine the status value of the computer system. The user specified algorithm may be in the form of programming code such as that described above. Receiving the user specified algorithm may be optional, as pre-defined algorithms may be used in various examples. The system status tracker may retrieve one or more topological maps from the topological map data 322 stored in the memory 320. The retrieved data is indicative of a topological map of the computer system that the computing device 300 is tracking. The retrieved topological map data may be indicative of device types of a plurality of components in the computer system and inter-relationships among the plurality of components. The inter-relationships include at least one first dependency of the computing device 300 on at least one first component of the plurality of components, and at least one second dependency of the at least one first component of the plurality of components on at least one second component of the plurality of components. Also, the system status tracker may retrieve data from the status algorithm data 324 that is indicative of at least one algorithm for determining a status of the computer system based on a status of at least the first component and a status of the second component.

[0036] Referring now to Figure 4, the system status tracker 330 may receive control signals from a plurality of components in the computer system, the control signals being associated with a status value of respective ones of the plurality of components with respect to at least one status domain (block 410). For example, the control signals may be status values in one or more

domains such as the compliance domain, the health domain, the performance domain and/or an administrator-defined domain. The control signals received by the system status tracker 330 may include the actual status values of the components of the computer system, or data that may enable the system status tracker 330 to utilize algorithms to calculate the status values of the components of the computer system.

[0037] At block 420, the system status tracker 330 determines a status of the computer system based on the status values of selected components. The selected components may be selected from the plurality of components based on a specified status domain and a topological map associated with the status domain, for example. The determination of the status of the computer system may also take into account the inter-relationships, which may be incorporated into the topological map. In some examples, the system status tracker 330 may determine the status value of a lower level component based on an algorithm associated with the lower-level component and inter-relationships of the at least one second component and at least one third component with a dependency with the at least one second component, and further based on the status values of the at least one first component, the at least one second component and the at least one third component.

[0038] The system status tracker may store data indicative of the status values of the respective ones of the plurality of components and data indicative of the determined status value of the computer system in the memory 320, e.g., in the other parameters data 326. At block 430, the system status tracker 330 displays information indicative of the status of the computer system to the user. In some examples, the system status tracker 330 may also display information indicative of the status values of the respective ones of the plurality of components.

[0039] Figure 5 illustrates an example status display 500 for the compliance status domain including the components depending from COMP1 as illustrated in the first topological map 100 of Figure 1. Figure 6 illustrates an example status display 600 for the health status domain including the components depending from COMP1 as illustrated in the second topological map 200 of Figure 2. The displays 500 and 600 include a status domain menu 510 which allows selection by a user of which of a plurality of status domains is displayed. In Figure 5, the compliance domain is selected and displayed while in Figure 6 the health domain is selected and displayed.

[0040] Each of the status displays 500 and 600 display component level status value indicators 515 that are selected from a status key 520. In these example displays 500 and 600, the status key 520 includes three status values, a "G" for a good or normal status, a "D" for a degraded status, and a "C" for a critical status. Other status values may be used such as, for example, numerical values such as 1 for good, 2 for degraded and 3 for critical. Other numerical scales may include status values from 1 to 10 with 1 being the best status and 10 being the most critical status. By associating numerical values with the status (e.g., status including good, degraded and critical), mathematical algorithms such as an averaging algorithm may be used. In other example displays, the component level status indicators 515 and the overall system status indicator 525 may be color coded. For example, green may signify a good status, yellow may signify a degraded status and red may signify a critical status.

[0041] In addition to displaying the component level status values, the status displays 500 and 600 each display an overall system status value indicator 525. In these examples, the overall system status value corresponds to the status value of the uppermost component COMP1. However, the system status tracker 330 may be implemented on a lower level component, or in a component external to the computer system being tracked, and in these cases, the status values of the computing device 300 where the system status tracker 330 is implemented may be different than the value of the overall system status indicatory 525.

[0042] In some examples, the system status tracker 330 may receive, via the user control interface 375, commands inquiring about the status of specified ones of the plurality of components. This allows a user or administrator an ability to drill down into the computer system to identify the root cause or causes of the overall system status. In response to receiving the inquiry commands, the system status tracker 330 may retrieve the stored data indicative of the status of the other components that were inquired about and communicate the retrieved data to be displayed on the display 360. The information displayed may include a device type, the status value(s) and the cause of the status value.

[0043] Various examples described herein are described in the general context of method steps or processes, which may be implemented in one example by a software program product or component, embodied in a machine-readable medium, including executable instructions, such as program code, executed by entities in networked environments. Generally, program modules may include routines, programs, objects, components, data structures, etc. which may be

designed to perform particular tasks or implement particular abstract data types. Executable instructions, associated data structures, and program modules represent examples of program code for executing steps of the methods disclosed herein. The particular sequence of such executable instructions or associated data structures represents examples of corresponding acts for implementing the functions described in such steps or processes.

[0044] Software implementations of various examples can be accomplished with standard programming techniques with rule-based logic and other logic to accomplish various database searching steps or processes, correlation steps or processes, comparison steps or processes and decision steps or processes.

[0045] The foregoing description of various examples has been presented for purposes of illustration and description. The foregoing description is not intended to be exhaustive or limiting to the examples disclosed, and modifications and variations are possible in light of the above teachings or may be acquired from practice of various examples. The examples discussed herein were chosen and described in order to explain the principles and the nature of various examples of the present disclosure and its practical application to enable one skilled in the art to utilize the present disclosure in various examples and with various modifications as are suited to the particular use contemplated. The features of the examples described herein may be combined in all possible combinations of methods, apparatus, modules, systems, and computer program products.

[0046] It is also noted herein that while the above describes examples, these descriptions should not be viewed in a limiting sense. Rather, there are several variations and modifications which may be made without departing from the scope as defined in the appended claims.

WHAT IS CLAIMED IS:

1. A device, comprising:

a memory storing data indicative of a topological map of a computer system, the topological map data being indicative of inter-relationships among a plurality of components of the computer system; and

a processor for executing a system status tracker, the system status tracker to:

receive signals from the plurality of components in the computer system, the signals being associated with a status value of respective ones of the plurality of components with respect to at least one status domain; and

determine a status value of the computer system based on the inter-relationships among the plurality of components and the status values of selected ones of the plurality of components, the selected ones being selected based on a selected status domain.

2. The device of claim 1, further comprising a display device to display information indicative of the status of the computer system.

3. The computing device of claim 1, wherein the received signals are indicative of status values for a plurality of status domains selected from a group consisting of a health domain representative of availability level of a component, a compliance domain representative of whether versions of software and or firmware of a component comply with current requirements, and a performance domain representative of a level at which a component is functioning relative to a normal level.

4. The computing device of claim 1, further comprising a user control interface configured to receive a user specified algorithm for determining the status of the computer system.

5. The computing device of claim 1, wherein, the inter-relationships further include redundancy information identifying two or more components of the plurality of components capable of performing a same function and having a same dependency relative to the computing device.

6. The computing device of claim 1, further comprising a user control interface configured to receive commands inquiring about a status of specified ones of the plurality of components, wherein the system status tracker provides the status of the other components.

7. The computing device of claim 1, wherein an algorithm used to determine the status of the computer system comprises at least one of:

a worst of underlying components algorithm;

an average of underlying components algorithm;

a highest of underlying components algorithm;

an inherent status measurement of at least one of the plurality of components;

a user-defined algorithm; or

a time varying algorithm.

8. The computing device of claim 1, wherein the plurality of components include at least one blade server and at least one blade residing within the at least one blade server.

9. A method, comprising:

receiving signals indicative of a status value of each of a plurality of components in a computer system;

determining a status of the computer system based on the status values of selected ones of the components, the selected ones of the components being selected based on a specified status domain and a topological map of the computer system, the topological map being indicative of inter-relationships among the plurality of components; and

presenting the status of the computer system to a user.

10. The method of claim 9, wherein presenting the status comprises displaying information indicative of the status values of at least one of the plurality of components.

11. The method of claim 9, wherein determining the status of the computer system comprises:

receiving a user-specified algorithm for determining the status of the computer system;

and

determining the status of the computer system based on the user-specified algorithm.

12. The method of claim 9, wherein the inter-relationships include redundancy information identifying two or more components of the plurality of components capable of performing a same function and having a same dependency, and wherein the status of the computer system is based on the redundancy information; and

wherein an algorithm used to determine the status of the computer system comprises at least one of:

a worst of underlying components algorithm;

an average of underlying components algorithm;

a highest of underlying components algorithm;

an inherent status measurement of at least one of the plurality of components;

a user-defined algorithm; or

a time varying algorithm.

13. A computer program product, embodied on a non-transitory computer-readable medium, comprising:

computer code for receiving signals indicative of a status value of each of a plurality of components in a computer system;

computer code for determining a status of the computer system based on the status values of selected ones of the components, the selected ones of the components being selected based on a specified status domain and a topological map of the computer system, the topological map being indicative of inter-relationships among the plurality of components; and

computer code for presenting the status of the computer system to a user.

14. The computer program product of claim 13, wherein the computer code for presenting the status comprises computer code for displaying information indicative of the status values of at least one of the plurality of components.

15. The computer program product of claim 13, wherein the computer code for determining the status of the computer system comprises:

computer code for receiving a user-specified algorithm for determining the status of the computer system; and

computer code for determining the status of the computer system based on the user-specified algorithm.

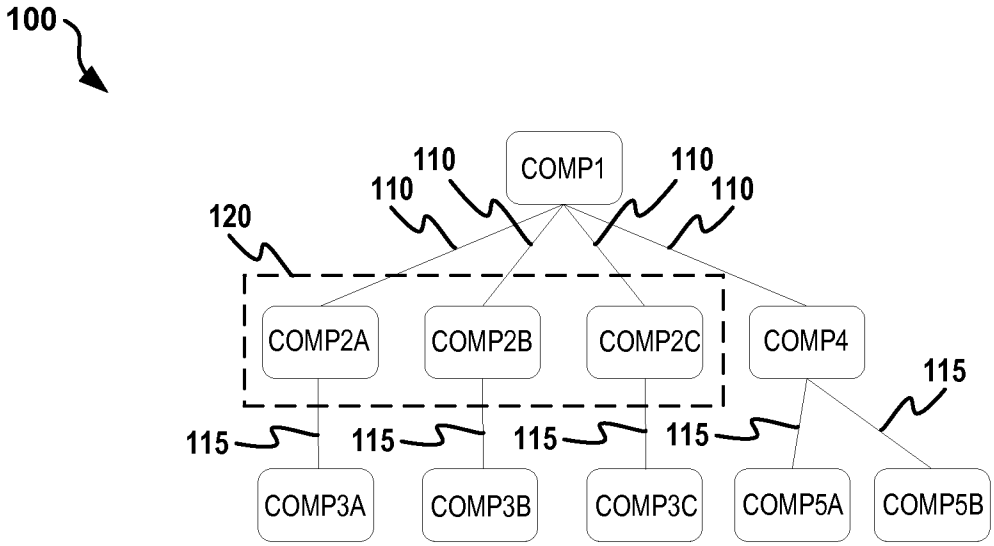

100
110
110
COMP1
110
110
120
COMP2A
COMP2B
COMP2C
COMP4
115
115
115
115
115
COMP3A
COMP3B
COMP3C
COMP5A
COMP5B

FIG. 1

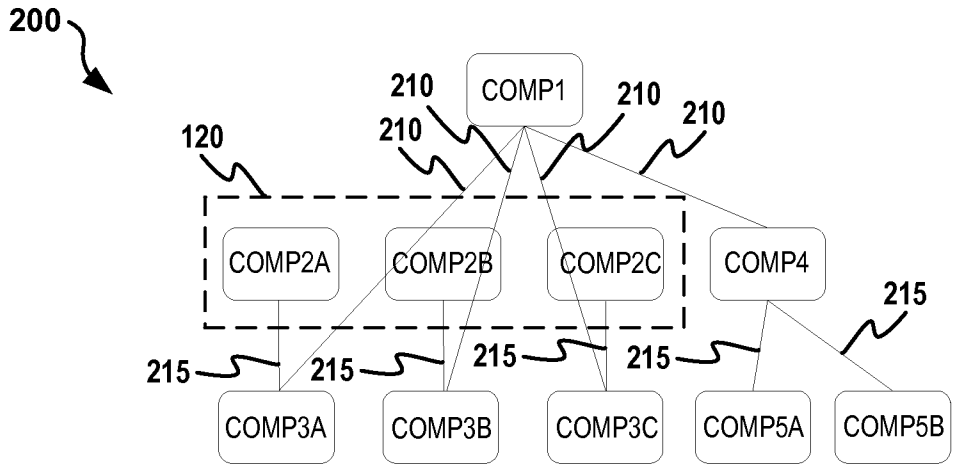

200
210
COMP1
210
210
210
120
COMP2A
COMP2B
COMP2C
COMP4
215
215
215
215
215
COMP3A
COMP3B
COMP3C
COMP5A
COMP5B

FIG. 2

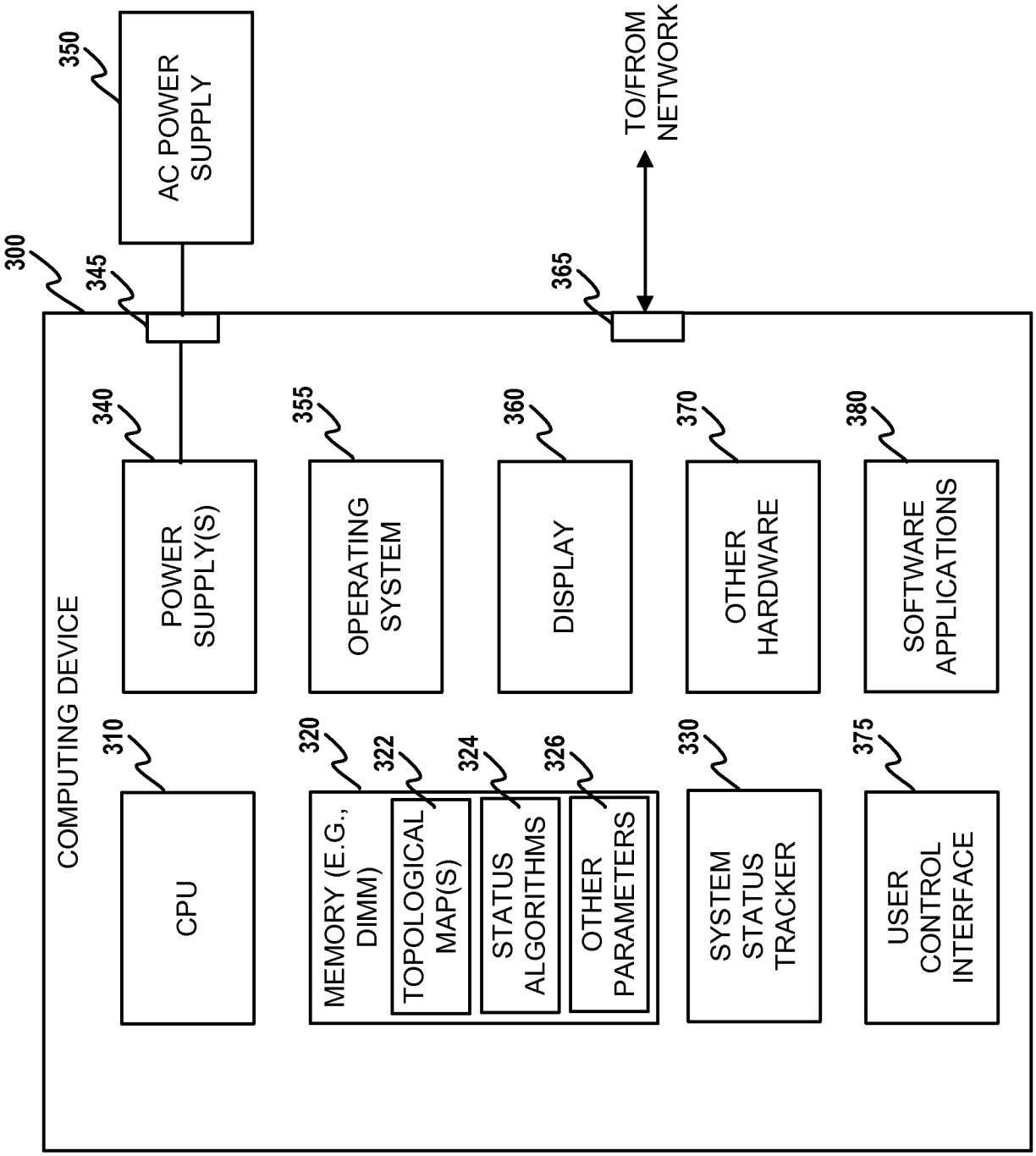

300
345
365
350
AC POWER SUPPLY
TO/FROM NETWORK
COMPUTING DEVICE
340
POWER SUPPLY(S)
355
OPERATING SYSTEM
360
DISPLAY
370
OTHER HARDWARE
380
SOFTWARE APPLICATIONS
310
CPU
320
MEMORY (E.G., DIMM)
322
TOPOLOGICAL MAP(S)
324
STATUS ALGORITHMS
326
OTHER PARAMETERS
330
SYSTEM STATUS TRACKER
375
USER CONTROL INTERFACE

FIG. 3

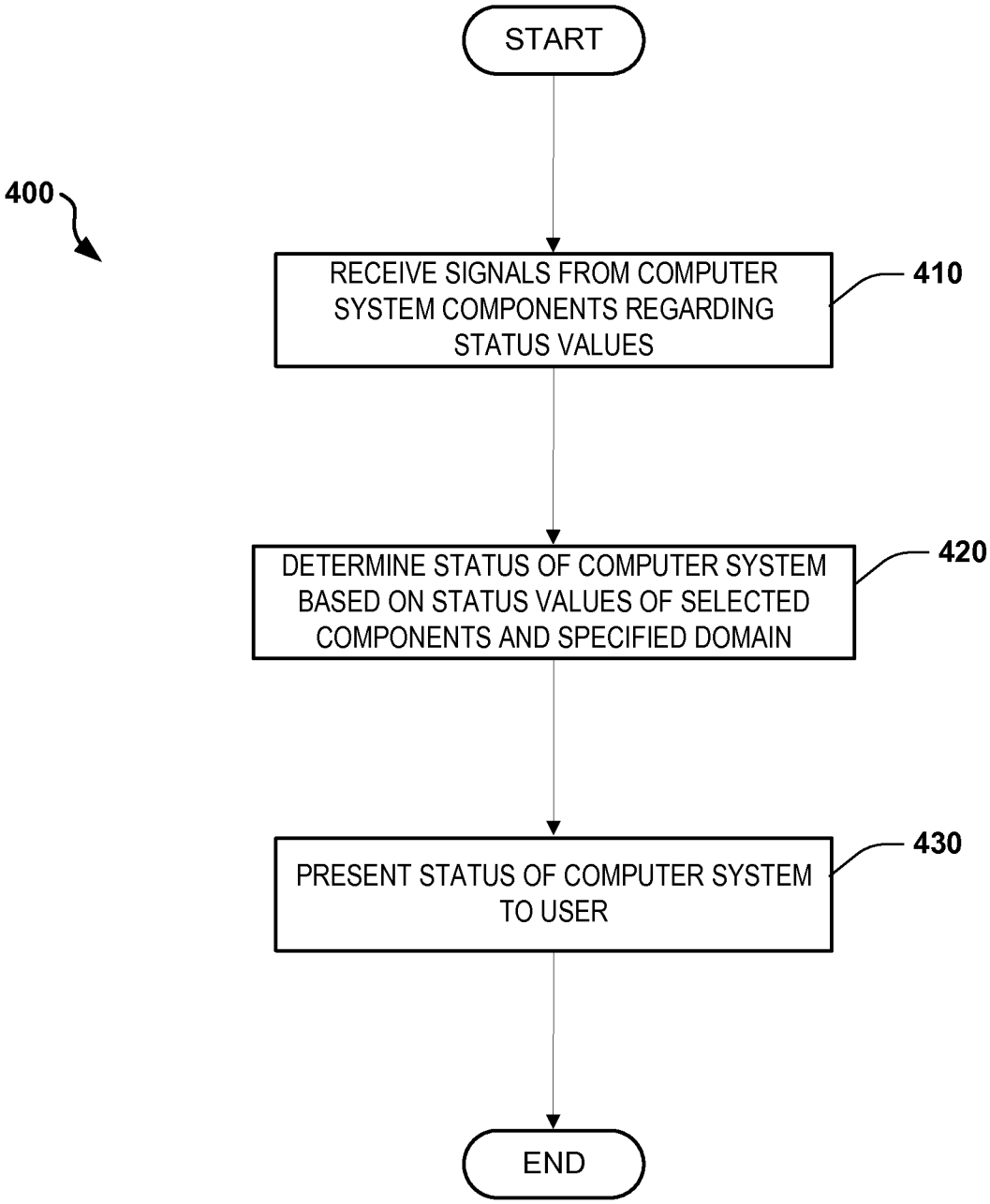

400
START
RECEIVE SIGNALS FROM COMPUTER SYSTEM COMPONENTS REGARDING STATUS VALUES
410
DETERMINE STATUS OF COMPUTER SYSTEM BASED ON STATUS VALUES OF SELECTED COMPONENTS AND SPECIFIED DOMAIN
420
PRESENT STATUS OF COMPUTER SYSTEM TO USER
430
END

FIG. 4

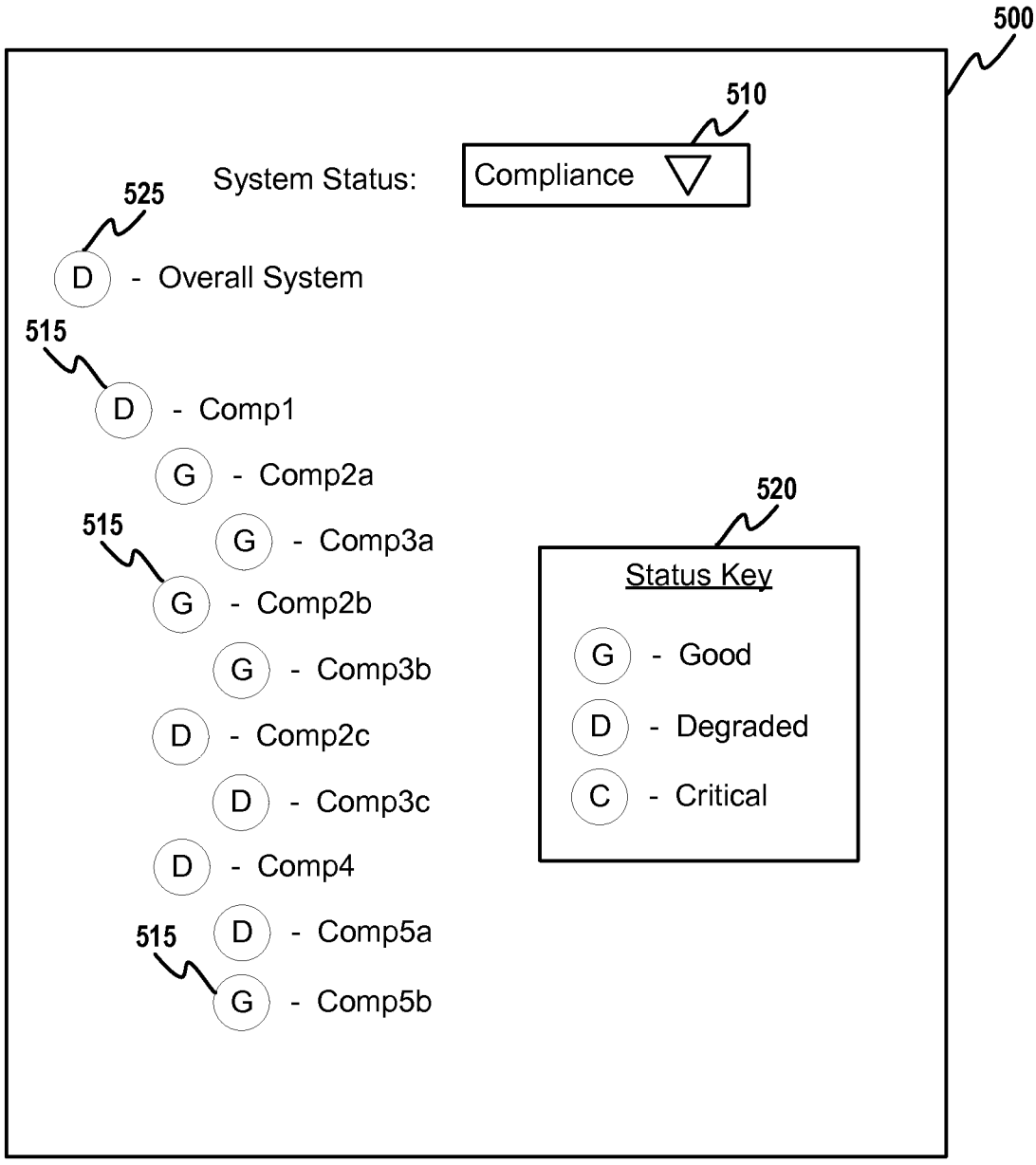

500
510
System Status:
Compliance
525
D - Overall System
515
D - Comp1
G - Comp2a
515
G - Comp3a
G - Comp2b
G - Comp3b
D - Comp2c
D - Comp3c
D - Comp4
515
D - Comp5a
G - Comp5b
520
Status Key
G - Good
D - Degraded
C - Critical

FIG. 5

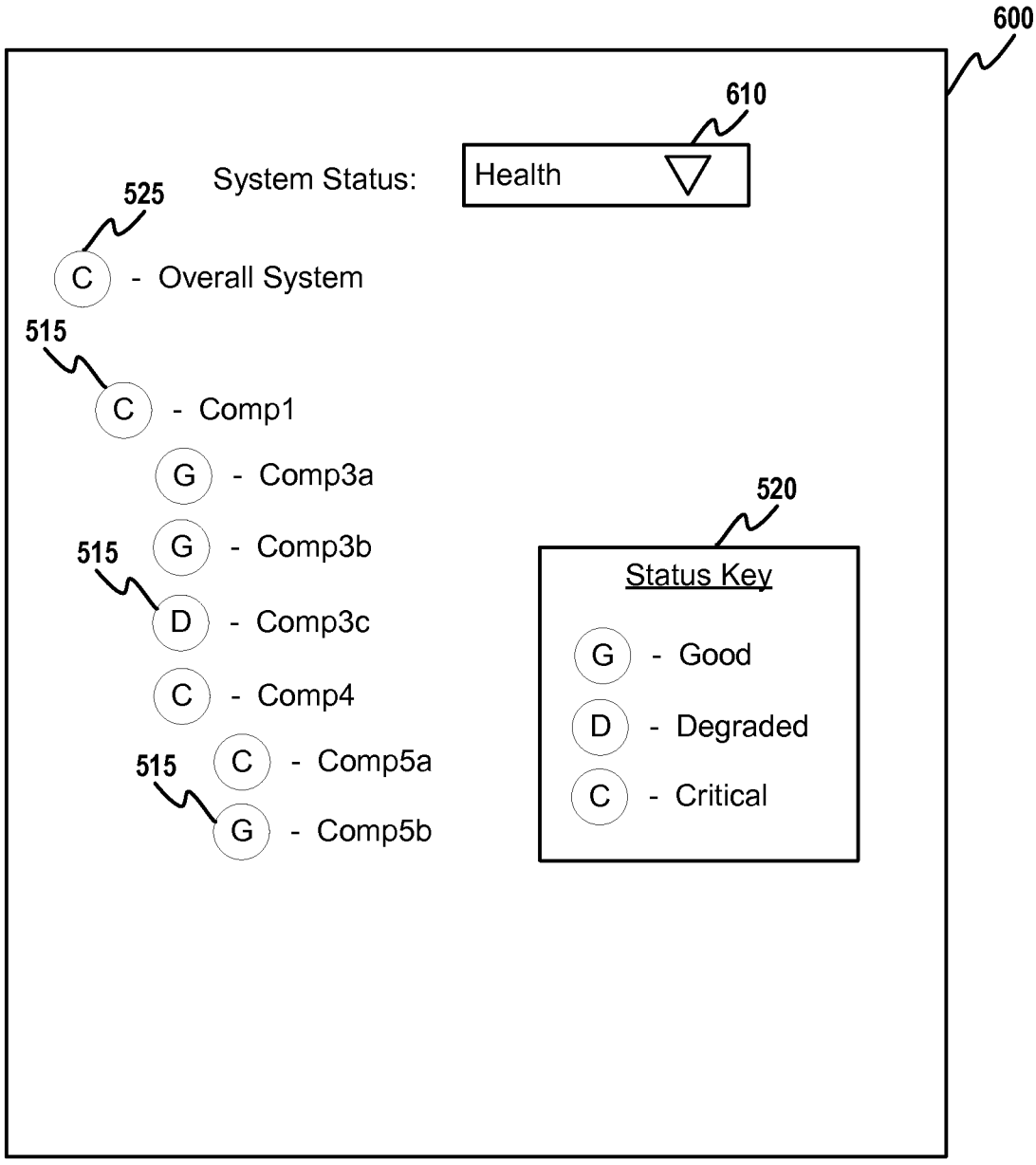

600
610
System Status:
Health
525
C - Overall System
515
C - Comp1
G - Comp3a
G - Comp3b
515
D - Comp3c
C - Comp4
C - Comp5a
515
G - Comp5b
520
Status Key
G - Good
D - Degraded
C - Critical

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/014009

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/32(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 11/32; G06G 7/62; G06F 11/07; G06F 15/173; G06F ; G06N 5/02; H02H 3/05

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: diagnostic, state of system, component, topology map, relation, algorithm, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2011-0153540 A1 (MIRZA MOHSIN BEG et al.) 23 June 2011 See paragraphs [0036], [0039], [0048], [0054], [0070]-[0077], [0084]-[0089], [0144], and [0153]; and figures 1-3 and 9.	1-3,5-6,8-10,13-14
Y		4,7,11-12,15
Y	US 2003-0204781 A1 (HAROLD L.PEEBLES et al.) 30 October 2003 See paragraph [0272] and figure 8B.	4,7,11-12,15
A	WO 02-079928 A2 (COVASOFT, INC.) 10 October 2002 See page 11, line 13 - page 12, line 13; and figure 1.	1-15
A	US 2007-0079243 A1 (ANDREW RALPH LEIGH et al.) 05 April 2007 See paragraphs [0039]-[0041]; and figures 3 and 5.	1-15
A	US 2010-0306588 A1 (KELVIN HAMILTON et al.) 02 December 2010 See paragraphs [0017]-[0018], [0044], and [0067]; and figures 5 and 12-13.	1-15

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 October 2015 (30.10.2015)	**30 October 2015 (30.10.2015)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
NHO, Ji Myong
Telephone No. +82-42-481-8528

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/014009

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0153540 A1	23/06/2011	US 8612377 B2	17/12/2013
US 2003-0204781 A1	30/10/2003	None	
WO 02-079928 A2	10/10/2002	AU 2002-254364 A1	15/10/2002
		US 2002-0173997 A1	21/11/2002
		US 2002-0184065 A1	05/12/2002
		US 2008-0040174 A1	14/02/2008
		US 7065566 B2	20/06/2006
		US 7237023 B2	26/06/2007
		US 7634563 B2	15/12/2009
		WO 02-079928 A3	21/08/2003
US 2007-0079243 A1	05/04/2007	None	
US 2010-0306588 A1	02/12/2010	EP 1714222 A2	25/10/2006
		GB 0325560 D0	03/12/2003
		US 2008-0059839 A1	06/03/2008
		US 7860682 B2	28/12/2010
		US 8086423 B2	27/12/2011
		WO 2005-045693 A2	19/05/2005
		WO 2005-045693 A3	19/03/2009