



- (51) International Patent Classification:
G06F 17/30 (2006.01) *G06Q 10/06* (2012.01)
- (21) International Application Number:
PCT/MY2015/050011
- (22) International Filing Date:
5 March 2015 (05.03.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
PI2014700545 6 March 2014 (06.03.2014) MY
- (71) Applicant: **MIMOS BERHAD** [MY/MY]; Technology
Park Malaysia, Kuala Lumpur, 57000 (MY).
- (72) Inventors: **KHOR, Swee Eng**; c/o MIMOS BERHAD,
TECHNOLOGY PARK MALAYSIA, Kuala Lumpur,
57000 (MY). **TAN, Chin Teong**; c/o MIMOS BERHAD,
TECHNOLOGY PARK MALAYSIA, Kuala Lumpur,
57000 (MY). **CHIN, Chee Kheen**; c/o MIMOS BERHAD,
TECHNOLOGY PARK MALAYSIA, Kuala Lumpur,

57000 (MY). **GOH, Teck Huat**; c/o MIMOS BERHAD,
TECHNOLOGY PARK MALAYSIA, Kuala Lumpur,
57000 (MY).

(74) Agent: **H A RASHID, Ahmad Fadzlee**; Ipvolusi Sdn Bhd,
A-3-3a, Centrio Pantai Hillpark, No. 1, Jalan Pantai Murni,
Kuala Lumpur, 59200 (MY).

(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,

[Continued on next page]

(54) Title: METHOD FOR DETECTING MISSING ASSOCIATION IN COMPUTING RESOURCES

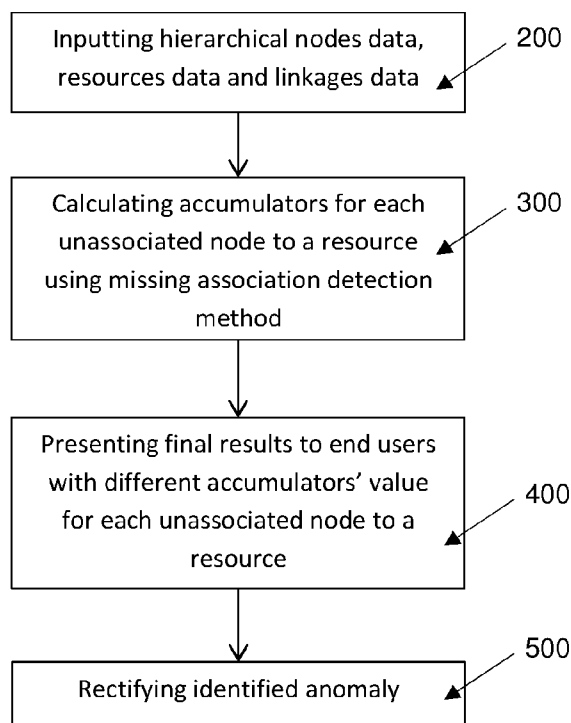


FIG. 3

(57) Abstract: The present invention relates to a method for detecting nodes with resource missing association anomalies in computing resources. The method includes inputting resources data, nodes data and linkages data between the resources and the nodes of an organisation tree structure by a Policy Assignment Module (20). It also includes computing accumulators for each node which is not associated to a resource in the organisation tree structure by a Missing Association Detection component (33) before displaying a real time result report to end users with different accumulators values for each node which is not associated to a resource by a Decision Display Module (40). Additionally, the method also includes rectifying the identified anomaly by an Anomalies Rectification Module (50).



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 applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *of inventorship (Rule 4.17(iv))*

Published:

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

**METHOD FOR DETECTING MISSING ASSOCIATION IN COMPUTING
RESOURCES**

FIELD OF INVENTION

5 The present invention relates to a method for detecting anomalies in computing resources. More particularly, the present invention relates to a method for detecting nodes with missing resource association anomalies in computing resources connected in an organisation tree structure.

10 **BACKGROUND OF THE INVENTION**

 Access right assignments of a policy in a large organisation are becoming more complicated these days. An individual may have multiple roles with multiple kind of access to a lot of software applications. The association or connection of an access right to an individual is usually carried out manually as it is a complicated
15 task and prone to errors.

 Furthermore, for an organisation treelike structure, wherein the organisation is represented by the tree structure; a division or a department of the organisation is represented by a sub-tree of the organisation treelike structure; an assignment is
20 represented by a resource; and an individual in the organisation is represented by a node, there is a high possibility of a missed-association when there are too many nodes linked to a resource. This could lead to a problem whereby an individual who is supposed to have an access to a resource is not assigned to the access right or resource. In other words, when there are too many people in an
25 organisation tree structure, it is highly likely that an individual is not correctly associated to a certain resource.

 Previously, United States Patent Application Publication number 2013/0051248 A1 has disclosed a method for hierarchical anomaly localisation and
30 prioritisation. The method disclosed comprises of obtaining reported status for a plurality of nodes of a hierarchical topology, the reported status for a particular node being at least one of normal, abnormal or indeterminate, and determining a subset of root cause abnormal nodes. The prior art also discloses an apparatus that comprises of a candidate selector to select a set of candidate root cause
35 abnormal nodes from a set of abnormal nodes identified in the hierarchical

topology and a root cause determiner to determine each respective candidate root cause abnormal node and respective subset of the set of abnormal nodes identified in the hierarchical topology.

5 Although many researches have been done for detecting anomalies, little have been done on detecting anomalies in an organisation tree hierarchy structure with missing associations with a resource. There is a need to provide a method that can detect suspected nodes with missing resource association anomalies that addresses the above mentioned drawbacks.

10

SUMMARY OF INVENTION

The present invention relates to a method for detecting nodes with missing resource association anomalies in computing resources connected in an organisation tree structure. The method is characterised by the steps of inputting resources data, nodes data and linkages data between the resources and the nodes of an organisation tree structure by a Policy Assignment Module (20); computing accumulators for each node which is not associated to a resource in the organisation tree structure by a Missing Association Detection component (33); and rectifying the identified anomaly by an Anomalies Rectification Module (50).

20

Preferably, the step of computing accumulators for each node which is not associated to a resource in the organisation tree structure includes retrieving a resource from a list of resources; initialising the accumulator of each node without any association to the retrieved resource to zero if the resource has not been processed; traversing each node and checking whether the node is a leaf or a root node; obtaining a sub-tree if the traversed node is a root node; setting the level of the sub-tree for each sub-tree obtained; computing the percentage of nodes that are not associated to the resource; computing the percentage of nodes associated to the resource; adding the accumulator value of the node with a possible missing link in the sub-tree with the percentage of nodes associated to the resource computed if the percentage of nodes that are not associated to the resource is less than a dynamic threshold; and setting the level of the sub-tree to another level.

35

Preferably, if the traversed node is not a root node, the computation stops.

Preferably, if the percentage of nodes that are not associated to the resource is more than the dynamic threshold there is no increment on the accumulator value of the node.

5 Preferably, the percentage of nodes that are not associated to the resource is computed as number of nodes not associated to a resource in a sub-tree divided by number of nodes in the sub-tree.

10 Preferably, the percentage of nodes associated to the resource is computed as number of nodes associated to a resource in a sub-tree divided by number of nodes in the sub-tree.

15 Preferably, the dynamic threshold is computed as 1 divided by level of the corresponding sub-tree.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention.

20 **FIG. 1** illustrates an architecture diagram of system (**1000**) for detecting resources anomalies according to an embodiment of the present invention.

25 **FIG. 2** illustrates a block diagram of an Authentication Resource Management Controller server (**100**) of **FIG. 1**.

FIG. 3 illustrates a flow chart of a method for detecting nodes with missing resources association anomalies according to an embodiment of the present invention.

30 **FIG. 4** illustrates a flow chart of missing association detection method according to the method shown in **FIG. 3**.

FIGS. 5 (a - i) illustrate an example of detecting nodes with missing resources association anomalies according to the method shown in **FIG. 4**.

35

DESCRIPTION OF THE PREFERRED EMBODIMENT

A preferred embodiment of the present invention will be described herein below with reference to the accompanying drawings. In the following description, well known functions or constructions are not described in detail since they would obscure the description with unnecessary detail.

Reference is made initially to **FIG. 1** which illustrates an architecture diagram of a system (**1000**) for detecting anomalies in computing resources according to an embodiment of the present invention. The anomalies detection system resides in an Authentication Resource Management Controller (ARMC) server (**100**), wherein the ARMC server (**100**) is connected to an ARMC front end webserver (**110**) and a database server (**120**). The ARMC front end webserver (**110**) is used to host the ARMC server (**100**) website, render webpages and deliver the webpages to end users. The database server (**120**) is used to store nodes data, resources data and organisation tree structures. The system for the anomalies detection is shown in detail in **FIG. 2**.

Reference is now made to **FIG. 2** which illustrates a block diagram of the ARMC server (**100**) shown in **FIG. 1**. The ARMC server (**100**) comprises of a Policy Assignment Module (**20**); a Tree Cascading Anomalies Detection Module (**30**); a Decision Display Module (**40**); and an Anomalies Rectification Module (**50**). The Policy Assignment Module (**20**), which is connected to the Tree Cascading Anomalies Detection Module (**30**), is used to input an association of a user to resources in an organisation tree structure. The Tree Cascading Anomalies Detection Module (**30**) further comprises of a Hierarchical Balanced component (**31**) which is used to detect resources over-association anomaly in the organisation tree structure, a Missing Association Detection component (**33**) which is used to detect potential nodes with missing resource association in the organisation tree and to quantify the suspicious level; and a Minority Group Differentiation component (**35**) which is used to detect an intergroup anomaly within the organisation tree. The Tree Cascading Anomalies Detection Module (**30**) is connected to the Decision Display Module (**40**) which is a graphical user-interface (GUI) display to display any detected anomalies to users graphically. It is connected to the Anomalies Rectification Module (**50**) that is used to rectify the anomaly detected such as the missing association of resources for a node.

Referring to **FIG. 3**, it illustrates a flow chart of a method for detecting nodes with missing resource association anomalies in computing resources connected in an organisation tree structure according to an embodiment of the present invention.

Initially, the Policy Assignment Module (20) inputs resources data, nodes data and linkages data between the resources and the nodes as in step 200. The resources data refers to access rights, wherein the access rights are used to determine whether a computing resource is able to access a particular data. The nodes data refers to the computing resources, while the linkages data between these resources and nodes refers to the assignments from a tree hierarchy structure. Next, the Missing Association Detection component (30) of the Tree Cascading Anomalies Detection module (30) computes accumulator values for each node which is not associated to a resource in the organisation tree structure as in step 300 using missing association detection method. Accumulator values are suspicious values for the nodes which are not associated to a resource. The higher the value, the more likely the node has a missing resource association anomaly. Thereon, the Decision Display Module (40) displays the real time result report to end users with different accumulators' values for each node which is not associated to a resource as in step 400. Finally, the Anomalies Rectification Module (50) rectifies the identified anomaly as in step 500 by reassigning the resources to the nodes with high risk of missing resource association anomaly which are determined by their high accumulator values.

Referring now to **FIG. 4**, it illustrates a detailed flow chart of the missing association detection method as in step 300 of the method of **FIG. 3**. The basic concept of the method is if most of the nodes within a sub-tree are associated to a resource, there is a high tendency of the remaining left out nodes are miss associated. Subsequently, the nodes with no association to resources are reflected with accumulator values, wherein the accumulator value is an indicator of a missed-association. The higher the accumulator value is, the higher chance of the node being classified as a missed-association.

After the Policy Assignment Module (20) inputs resources data, nodes data and linkages data between the resources and the nodes as in step 200, the Missing Association Detection component (33) retrieves a resource from a list of resources of an organisation tree structure as in step 301. It processes each node in the

organisation tree structure by iterating from the resource list. If the resource has yet to be processed as in decision **302**, the accumulator of each node without any association to the retrieved resource is initialised to zero as in step **303**. The Missing Association Detection component (**33**) traverses for each node as in step **304** and
5 checks whether the node is a leaf node or a root node as in decision **305**. When the traversed node is not a leaf node but a root node, the Missing Association Detection component (**33**) obtains a sub-tree as in step **306** to compute suspicious score of possible missing link for each traversed node. However, if the traversed node is not a root node but a leaf node, the computation stops.

10 For each sub-tree obtained, the Missing Association Detection component (**33**) sets the level of the sub-tree, n starting from level 2 as in step **307** and step **308**. Thereon, it computes the percentage of nodes that are not associated to a resource, p , wherein p is computed as number of nodes not associated to a resource in a sub-tree divided by number of nodes in the sub-tree as in step **309** and the percentage of
15 nodes associated to the resource, w , wherein w is computed as number of nodes associated to a resource in a sub-tree divided by number of nodes in the sub-tree or $1 - p$ as in step **310**. If p is less than a dynamic threshold, wherein the dynamic threshold is computed as 1 divided by level of the corresponding sub-tree as in
20 decision **311**, the accumulator value of the node with a possible missing link in the sub-tree is added with the value of w as in step **312** before setting the level of the sub-tree to another level as in step **313**.

25 However, if p is more than the dynamic threshold as in decision **311**, there is no increment on the accumulator value of the node and the Missing Association Detection component (**33**) directly sets the level of the sub-tree to another level as in step **313**. The process then repeats from step **309**, which is the step of computing p until the sub-tree has reached the maximum level as in decision **314**. Once the level of the sub-tree has reached the maximum level, the Missing Association Detection
30 component (**33**) checks whether or not the traversed node is the last node in the resource as in decision **315**. If the traversed node is not the last node in the resource, the process repeats from step **304**, wherein the Missing Association Detection component (**33**) traverses to the next node. Once the Missing Association Detection component (**33**) has traversed until the last node in the resource, the process then
35 repeats from step **301**, wherein the Missing Association Detection component (**33**)

retrieves another resource from the list of resources of the organisation tree structure.

Once the sub-tree reaches level n as in step 314 and when the node traversed is the last node as in decision 315, the process repeats from step 301, wherein the Missing Association Detection component (33) retrieves another resource from the resource list. However, if there are no more resources as in decision 302, the process proceeds to step 400 which is displaying a real time result report to end users with different accumulators' values for each node which is not associated to a resource.

Referring to FIGS. 5 (a - i), they illustrate an example of detecting nodes with missing resources association anomalies of an organisation tree structure according to the method shown in FIG. 4. FIG 5 (a) initially shows the organisation tree with missing resources association. Coloured nodes represent the nodes that are already associated to the coloured resource by an administrator, while the white nodes are the nodes with high tendency of missing resources association.

As the first node traversed in the organisation tree is a root node, it gets a sub-tree with different levels. From the organisation tree shown in FIG. 5 (a), the node is able to get 3 sub-trees starting from level 2 to a maximum of level 4 sub-tree. In each level of the sub-tree, the percentage of nodes that are not associated to the resource, p is computed. If the percentage is less than a dynamic threshold, which is 1 divided by level of the corresponding sub-tree, the nodes without the resource get an increment of accumulator value.

FIG. 5 (b) illustrates the first traversed node with the sub-tree at level 2. Initially, the percentage of nodes that are not associated to the resource, p is computed. As there is only 1 out of 3 nodes in the corresponding sub-tree that is not connected to the resource, the percentage of nodes that are not associated to the resource, p is $1/3$. Next, the dynamic threshold of the corresponding sub-tree is computed. As the level of sub-tree for the corresponding sub-tree is 2, the dynamic threshold is $1/2$. Thereon, the percentage of nodes associated to the resource of the sub-tree, w is computed. As there are 2 out of 3 nodes in the corresponding sub-tree associated to the resource, the percentage of nodes associated to the resource

equals to $2/3$ or 0.6667 . The Missing Association Detection component (33) then checks whether the percentage of nodes that are not associated to the resource, p is less than the dynamic threshold. If the percentage of nodes that are not associated to the resource, p is less than the dynamic threshold of the corresponding sub-tree, the nodes that are not associated to the resource of the sub-tree get an increment of accumulator value. Since the percentage of nodes that are not associated to the resource, p which is $1/3$ is less than the dynamic threshold of the corresponding sub-tree which is $1/2$, the node that is not associated to the resource of the sub-tree gets an increment of accumulator value which is equal to the percentage of nodes associated to the resource, w which is 0.6667 .

Thereon, the first traversed node loops to the next level of sub-tree which is level 3 as shown in FIG. 5 (c). Similarly, the Missing Association Detection component (33) computes the percentage of nodes that are not associated to the resource, p of the corresponding sub-tree and it is less than the dynamic threshold, wherein the percentage of nodes that are not associated to the resource, p is $2/7$, and wherein the dynamic threshold of the sub-tree is $1/3$. Since the percentage of nodes that are not associated to the resource, p is less than the dynamic threshold, the nodes that are not associated to the resource of the sub-tree get an increment of accumulator, w which is $5/7$. Thus, the accumulator values of the nodes are 1.3809 and 0.7142 respectively.

Referring now to FIG. 5 (d), the traversed node loops to the next level of sub-tree which is to level 4. Similarly, the Missing Association Detection component (33) computes the percentage of nodes that are not associated to the resource, p of the corresponding sub-tree. Since the percentage of nodes that are not associated to the resource, p of the corresponding sub-tree is $3/10$ and is more than the dynamic threshold which is $1/4$, there is no increment on all accumulators. Once the traversed node has looped till the maximum number of level for the sub-tree, the Missing Association Detection component (33) traverses to the next node.

FIG. 5 (e) illustrates the second traversed node with its first sub-tree, wherein the second traversed node is able to get 2 levels of sub-tress. For the first level of the sub-tree which is at level 2, the Missing Association Detection component (33) computes the percentage of nodes that are not associated to the resource, p of the

corresponding sub-tree which is $1/4$. Since the percentage of nodes that are not associated to the resource, p is less than the dynamic threshold which is $1/2$, the node that is not associated to the resource of the sub-tree gets an increment of $3/4$ on its accumulator value making it a total of 1.4642.

5

Thereon, the Missing Association Detection component (33) loops to the second level of sub-tree of the second traversed node loop which is level 3 as shown in FIG. 5 (f). Since the percentage of nodes that are not associated to the resource, p is $2/7$ which is less than the dynamic threshold which is $1/3$, the node that is not associated to the resource of the sub-tree gets an increment of $5/7$ on its accumulator value making it a total of 2.1785.

10

Next, the Missing Association Detection component (33) traverses to the third node as shown in FIG. 5 (g). The third traversed node only gets one sub-tree with a level of 2. There is no increment of accumulator value as there is no node that is not associated to the resource of the corresponding sub-tree.

15

Referring now to FIG.5 (h), it illustrates the fourth traversed node, wherein it only gets one sub-tree with a level of 2. Since the percentage of nodes that are not associated to the resource, p is $1/4$ which is less than the dynamic threshold which is $1/2$, the node that is not associated to the resource of the corresponding sub-tree gets an increment of $3/4$ on its accumulator value making it a total of 1.4643. The iteration continues for all nodes except leaf nodes. The final accumulation values are shown in FIG. 5 (i), wherein a higher accumulator value shows a higher tendency of a miss-association of the nodes,

20

25

While embodiments of the invention have been illustrated and described, it is not intended that these embodiments illustrate and describe all possible forms of the invention. Rather, the words used in the specifications are words of description rather than limitation and various changes may be made without departing from the scope of the invention.

30

CLAIMS

1. A method for detecting nodes with resource missing association anomalies in computing resources is characterised by the steps of:
 - 5 a) inputting resources data, nodes data and linkages data between the resources and the nodes of an organisation tree structure by a Policy Assignment Module (20);
 - b) computing accumulators for each node which is not associated to a resource in the organisation tree structure by a Missing Association Detection component (33); and
 - 10 c) rectifying the identified anomaly by an Anomalies Rectification Module (50).
2. The method as claimed in claim 1, wherein the step of computing accumulators for each node which is not associated to a resource in the
15 organisation tree structure includes:
 - a) retrieving a resource from a list of resources;
 - b) initialising the accumulator of each node without any association to the retrieved resource to zero if the resource has not been processed;
 - c) traversing each node and checking whether the node is a leaf or a
20 root node;
 - d) obtaining a sub-tree if the traversed node is a root node;
 - e) setting the level of the sub-tree for each sub-tree obtained;
 - f) computing the percentage of nodes that are not associated to the resource;
 - 25 g) computing the percentage of nodes associated to the resource;
 - h) adding the accumulator value of the node with a possible missing link in the sub-tree with the percentage of nodes associated to the resource computed if the percentage of nodes that are not associated to the resource is less than a dynamic threshold; and
 - 30 i) setting the level of the sub-tree to another level.
3. The method as claimed in claim 2, wherein if the traversed node is not a root node, the computation stops.

4. The method as claimed in claim 2, wherein if the percentage of nodes that are not associated to the resource is more than the dynamic threshold there is no increment on the accumulator value of the node.
- 5 5. The method as claimed in claim 2, wherein the percentage of nodes that are not associated to the resource is computed as number of nodes not associated to a resource in a sub-tree divided by number of nodes in the sub-tree.
- 10 6. The method as claimed in claim 2, wherein the percentage of nodes associated to the resource is computed as number of nodes associated to a resource in a sub-tree divided by number of nodes in the sub-tree.
- 15 7. The method as claimed in claim 2, wherein the dynamic threshold is computed as 1 divided by level of the corresponding sub-tree.

20

25

30

35

1/8

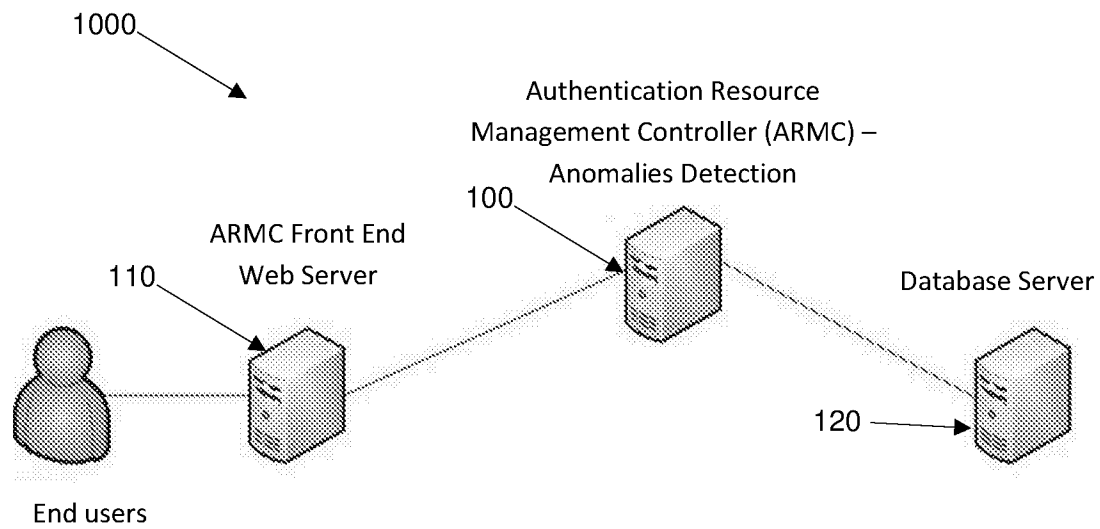


FIG. 1

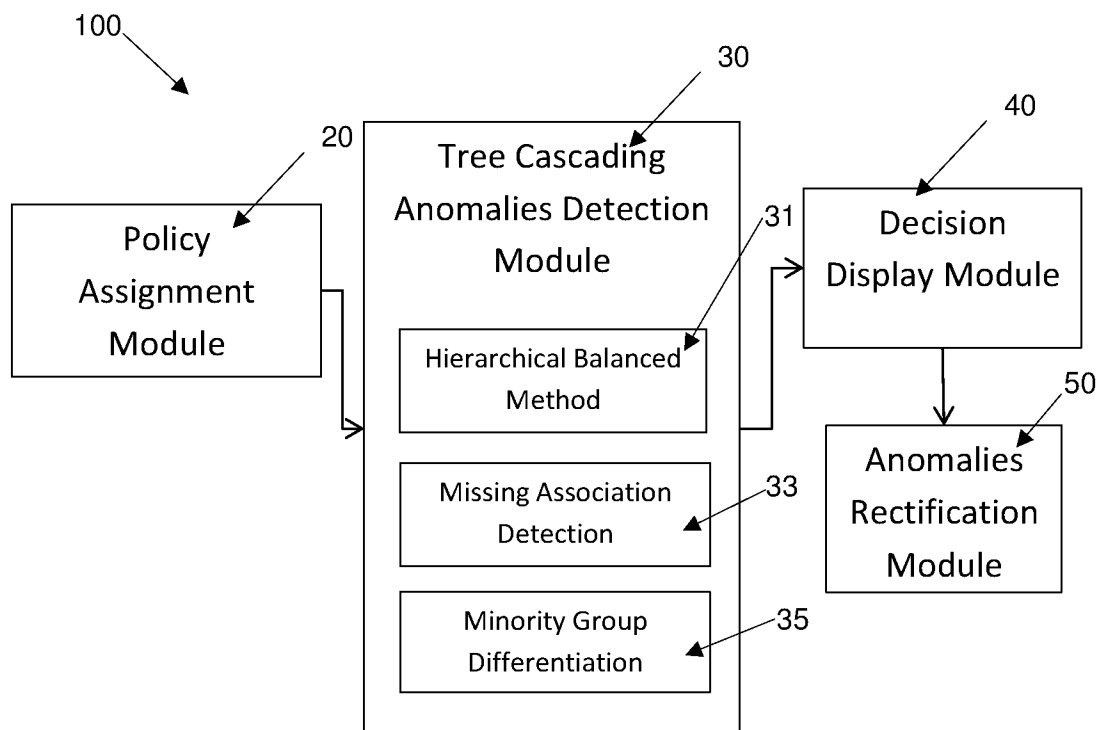
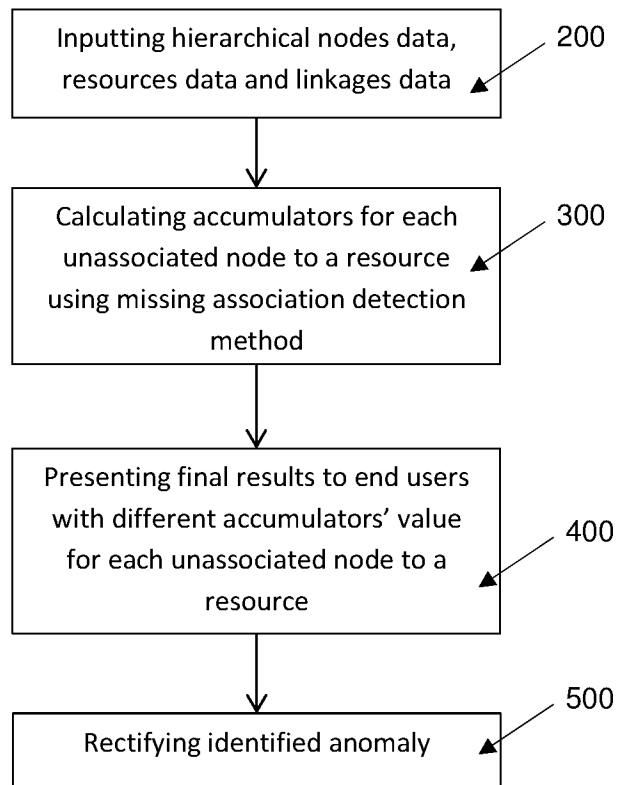


FIG. 2

2/8

**FIG. 3**

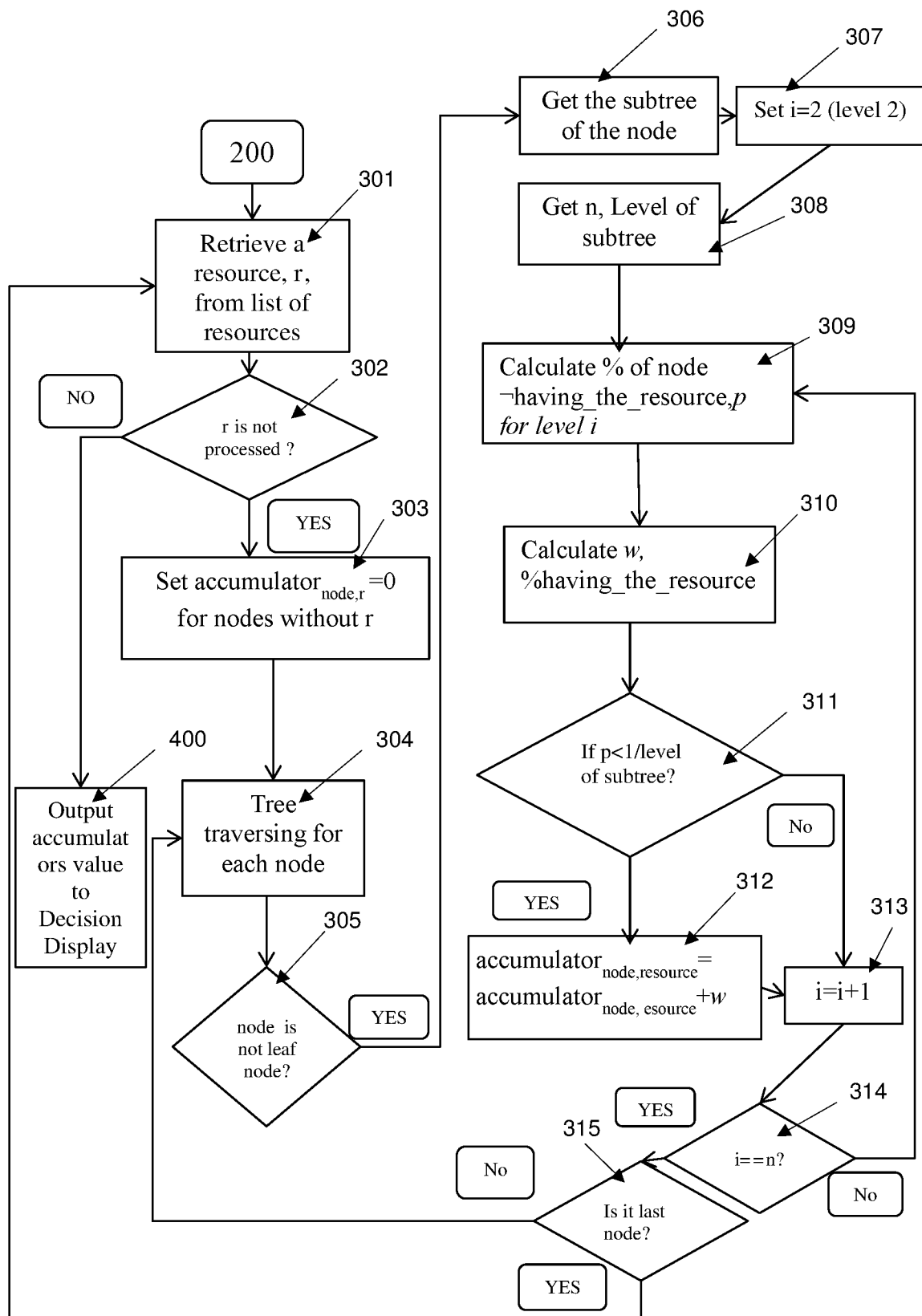


FIG. 4

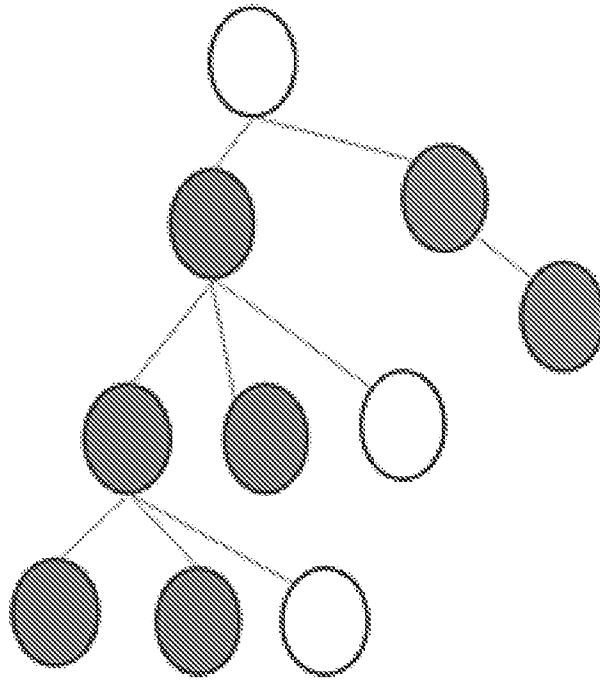
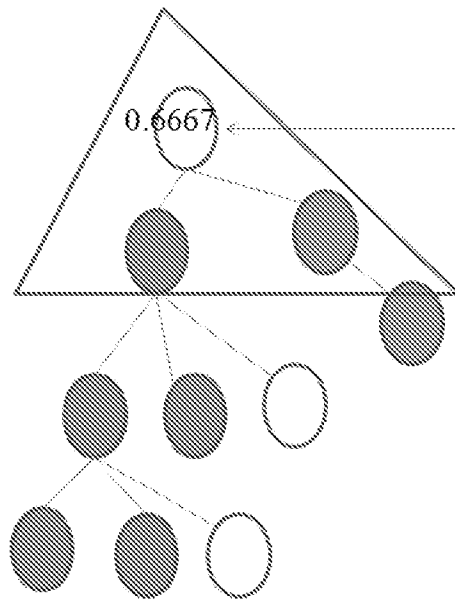


FIG. 5 (a)

- First traversed node:



Level 2

Node and its subtree

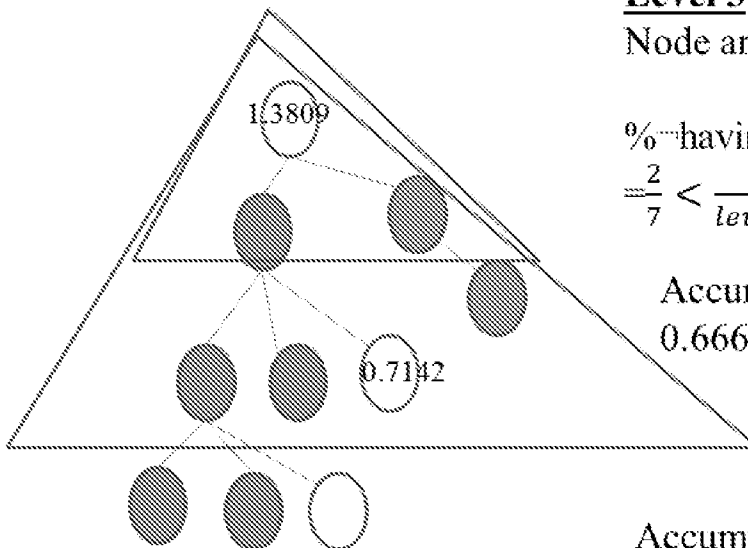
$$\frac{1}{3} < \frac{\% \text{having the resource}}{\text{level of subtree}} = \frac{1}{2}$$

Thus, increase

$$\text{Accumulator}_{\text{root}} = 0.6667$$

FIG. 5 (b)

- First traversed node:



Level 3

Node and its subtree

$$\frac{\% \text{ having the resource}}{7} < \frac{1}{\text{level of subtree}} \approx \frac{1}{3}$$

Accumulator_{root} =
0.6667+0.7142=1.3809

Accumulator_{node3,3}=0.7142

FIG. 5 (c)

6/8

• First traversed node:

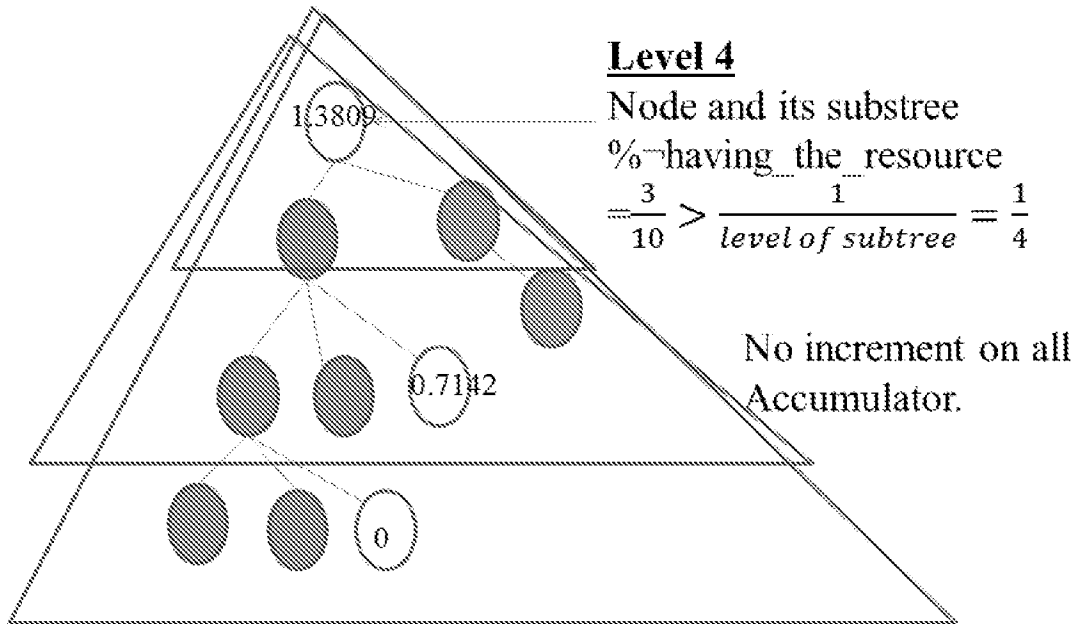


FIG. 5 (d)

• Second traversed node :

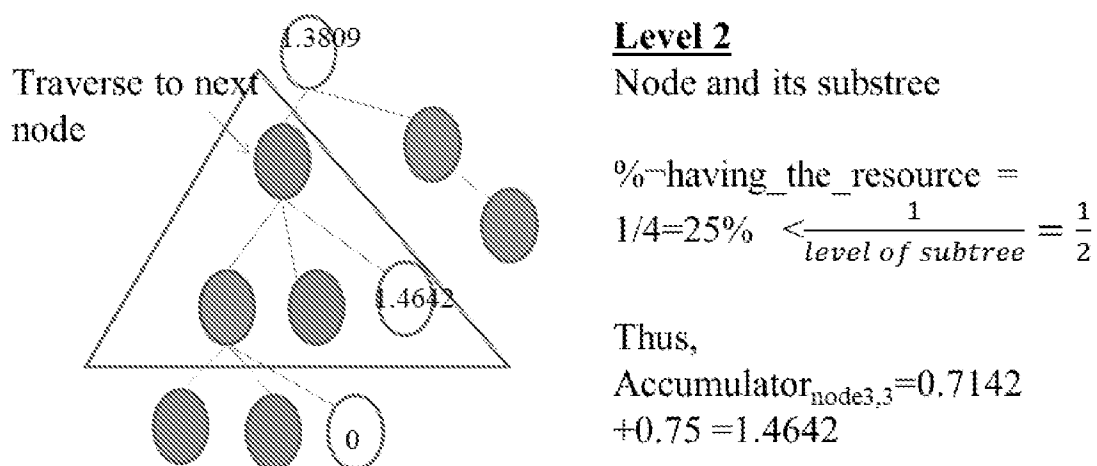


FIG. 5 (e)

7/8

• Second traversed node :

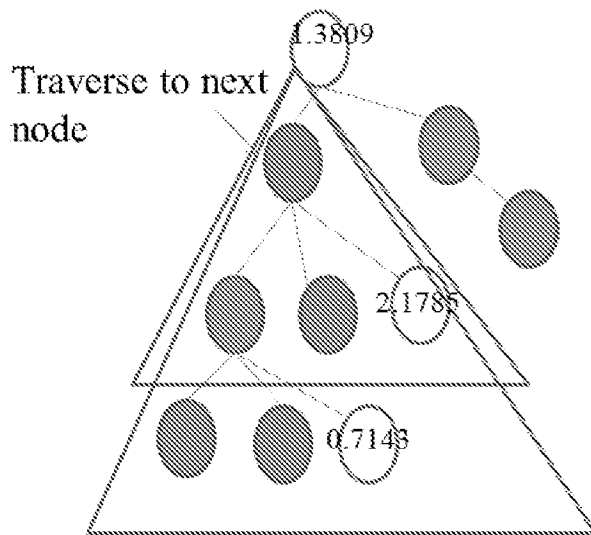


FIG. 5 (f)

Level 3

Node and its subtree

%-having_the_resource =
2/7=28.57%

$$\frac{1}{\text{level of subtree}} = \frac{1}{3}$$

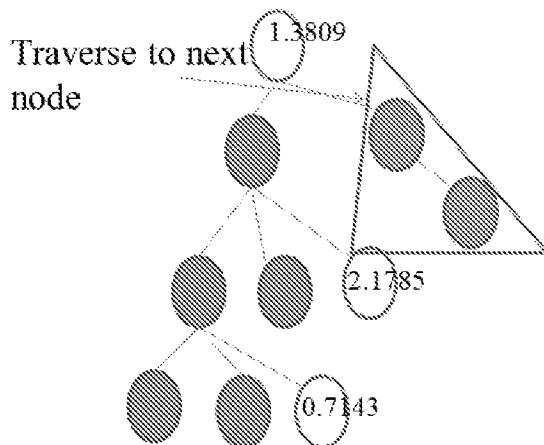
Thus, accumulator

Accumulator_{node3,3} =

1.4642+0.7143=2.1785

Accumulator_{node4,3} = 0.7143

• Third traversed node :



Level 2:

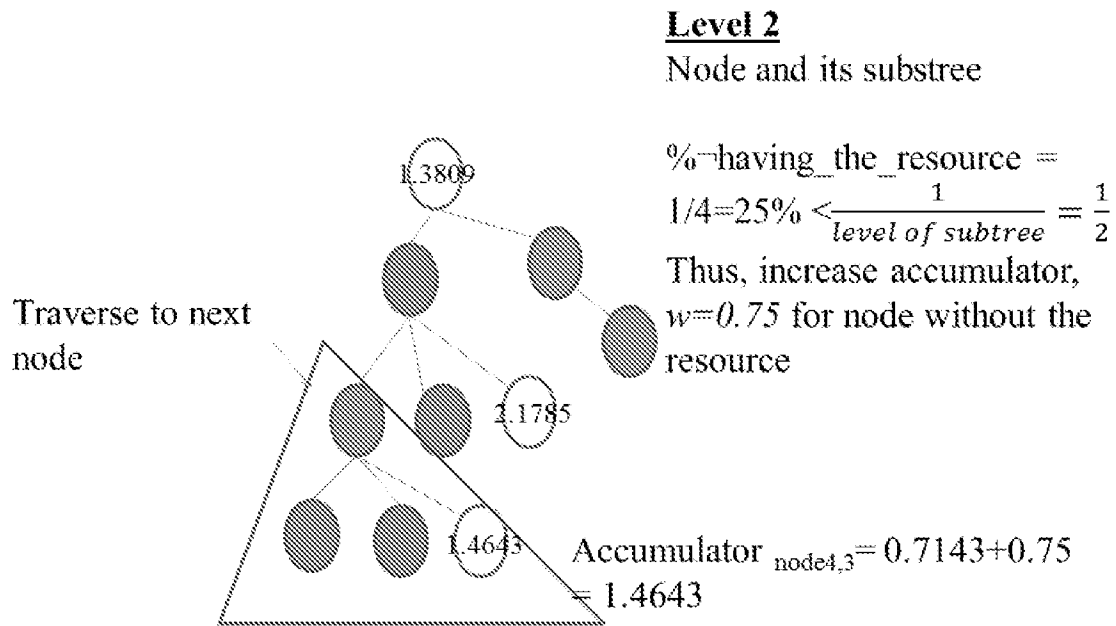
Node and its subtree

%-having_the_resource =
0/2=0% >=

$$\frac{1}{\text{level of subtree}} = \frac{1}{2}$$

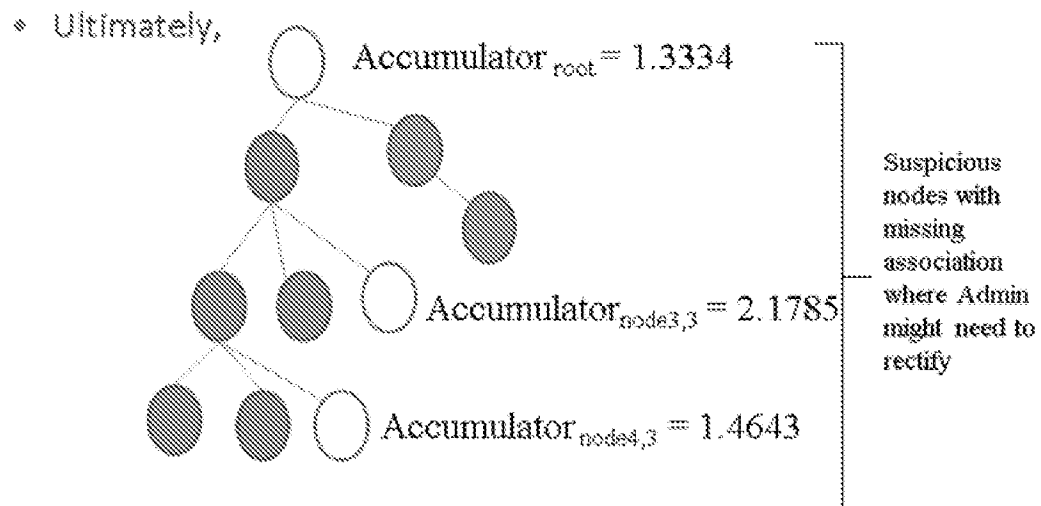
Increment to node not having
the resource in the subtree. As no
Nodes without resource, so no
Accumulator increased

FIG. 5 (g)



.....Continue the iteration for all node others than leaf node

FIG. 5 (h)



Higher accumulator indicate higher tendency a miss-association.

FIG. 5 (i)

INTERNATIONAL SEARCH REPORT

 International application No.
PCT/MY2015/050011

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30 (2006.01) G06Q 10/06 (2012.01)

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)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPIAP & EPODOC, Google patents, Patentscope, freepatentsonline

Applicant/Inventor name search

Keywords & Phrases: access management, user permissions, anomaly detection, organisation tree hierarchy structure, access rights in a organisation tree structure and like terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	Documents are listed in the continuation of Box C	



Further documents are listed in the continuation of Box C



See patent family annex

* "A"	Special categories of cited documents: document defining the general state of the art which is not considered to be of particular relevance	"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
"E"	earlier application or patent but published on or after the international filing date	"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
"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)	"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
"O"	document referring to an oral disclosure, use, exhibition or other means	"&"	document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 27 April 2015	Date of mailing of the international search report 27 April 2015
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA Email address: pct@ipaustalia.gov.au	Authorised officer Rajni Verma AUSTRALIAN PATENT OFFICE (ISO 9001 Quality Certified Service) Telephone No. 0399359635

INTERNATIONAL SEARCH REPORT		International application No.
C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		PCT/MY2015/050011
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	US 8359652 B2 (BHAGWAN et al.) 22 January 2013 col. 5, lines 24-51, col. 5, line 65-col. 6, line 19, Fig. 2, step 140 and associated text Whole Document	1 2-7
Y A	US 2012/0030756 A1 (TODD) 02 February 2012 Fig 3. step 310 and associated text Whole Document	1 2-7
A	US 20120059687 A1 (KEYTE et al.) 08 March 2012 Whole Document	1-7
A	US 2012/0117019 A1 (WOLF et al.) 10 May 2012 Whole Document	1-7

INTERNATIONAL SEARCH REPORT Information on patent family members		International application No. PCT/MY2015/050011	
This Annex lists known patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.			
Patent Document/s Cited in Search Report		Patent Family Member/s	
Publication Number	Publication Date	Publication Number	Publication Date
US 8359652 B2	22 January 2013	US 8359652 B2	22 Jan 2013
US 2012/0030756 A1	02 February 2012	US 8484724 B2	09 Jul 2013
US 20120059687 A1	08 March 2012	AU 2010225512 A1	13 Oct 2011
		AU 2010225512 A2	17 Nov 2011
		WO 2010107326 A1	23 Sep 2010
US 2012/0117019 A1	10 May 2012	WO 2012061254 A2	10 May 2012
End of Annex			
Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001. Form PCT/ISA/210 (Family Annex)(July 2009)			