

US PATENT & TRADEMARK OFFICE

PATENT APPLICATION FULL TEXT AND IMAGE DATABASE



(1 of 1)

United States Patent Application

20190311069

Kind Code

A1

IVANOV; Konstantin ; et al.

October 10, 2019

METHOD FOR DISCOVERING A GROUP OF INTERDEPENDENT COMPUTING OBJECTS WITHIN UNDIRECTED GRAPH STRUCTURE IN A COMPUTING ENVIRONMENT CORRESPONDING TO A SET OF KNOWN CONDITIONS

Abstract

Various embodiments relate to a method and apparatus for discovering a group of interdependent computing objects within an undirected graph structure of objects in a computing environment, the method comprising the steps of performing object identification, performing relations identification, selecting objects by performing the steps comprising defining required properties, matching the required properties with the properties for each of the objects, dropping the objects which have at least one of the properties which do not match with the required properties, making a selection by performing object intersection on the objects by performing the steps of reading location and the relations for each of the objects, selecting a group including each of the objects in the sets of objects which are identical based on the relations for each of the objects, determining whether each of the objects in the group are related to each of the objects in each of the sets of objects, and determining whether at least one of the objects in each of the sets of objects is related to at least one of the objects in another set of the sets of objects.

Inventors: **IVANOV; Konstantin;** (*Brno, CZ*) ; **DURIS; Marek;** (*Brno, CZ*) ; **SALI; Aylin;** (*Brno, CZ*) ; **MARKOV; Stanimir;** (*Sliven, BG*) ; **RADU; Ionut;** (*Sasar, RO*)

Applicant: **Name** **City** **State** **Country** **Type**

Runecast Solutions Limited London GB

Assignee: **Runecast Solutions Limited**

Family ID: **68097215**

Appl. No.: **15/947485**

Filed: **April 6, 2018**

Current U.S. Class:

1/1

Current CPC Class:

G06F 16/289 20190101; G06F 16/284 20190101; G06F 16/2455 20190101; G06F 16/90335 20190101; G06F 16/9024 20190101;

Claims

1. A method for discovering a group of interdependent computing objects within an undirected graph structure of objects in a computing environment, the method comprising the steps of: performing object identification; performing relations identification; selecting objects by performing the steps comprising: defining required properties; matching the required properties with the properties for each of the objects; dropping the objects which have at least one of the properties which do not match with the required properties; making a selection by performing object intersection on the objects by performing the steps of: reading location and the relations for each of the objects; selecting a group including each of the objects in the sets of objects which are identical based on the relations for each of the objects; determining whether each of the objects in the group are related to each of the objects in each of the sets of objects, and determining whether at least one of the objects in each of the sets of objects is related to at least one of the objects in another set of the sets of objects.
2. The method for discovering a group of interdependent computing objects within an undirected graph structure of objects in the computing environment of claim 1, the method comprising the steps of: performing the object identification by performing the steps comprising: receiving a search query including objects and operators; selecting an organization area; isolating object types for the objects in the search query; grouping the objects into sets of objects, and recording properties for each of the objects.
3. The method for discovering a group of interdependent computing objects within an undirected graph structure of objects in the computing environment of claim 1, the method comprising the steps of: performing the relations identification by performing the steps comprising: recording relatability for each of the object types for the objects, and recording relations for each of the objects.
4. The method for discovering a group of interdependent computing objects within an undirected graph structure of objects in the computing environment of claim 1, the method comprising the steps of: making a selection by performing object intersection on the objects by performing the steps of: dropping all of the objects from the group when each of the objects in the group are not related to each of the objects in each of the sets of objects.
5. The method for discovering a group of interdependent computing objects within an undirected graph structure of objects in the computing environment of claim 1, the method comprising the steps of: making a selection by performing object intersection on the objects by performing the steps of: dropping all of the objects from the group when each of the sets of objects is not related to at least one of the objects in another set of the sets of objects.
6. The method for discovering a group of interdependent computing objects within an undirected graph structure of objects in the computing environment of claim 1, the method comprising the steps of: making a selection by performing object union on the objects by performing the steps of: reading location and the relations for each of the objects, and selecting each of the objects that are in each of the sets of objects.
7. The method for discovering a group of interdependent computing objects within an undirected graph structure of objects in the computing environment of claim 6, the method comprising the steps of: making a selection by performing object union on the objects by performing the steps of: dropping each of the objects that are not in each of the sets of objects.
8. The method for discovering a group of interdependent computing objects within an undirected graph structure of objects in the computing environment of claim 1, the method comprising the steps of: making a selection by performing object combination on the objects by performing the steps of: reading location and the relations for

each of the objects; selecting each of the objects that are in each of the sets of objects, and determining whether each of the objects in a first set of objects is related to at least one the objects in a second set of objects.

9. The method for discovering a group of interdependent computing objects within an undirected graph structure of objects in the computing environment of claim 8, the method comprising the steps of: making a selection by performing object combination on the objects by performing the steps of: dropping each of the objects in a first set of objects which is not related to at least one the objects in a second set of objects.

10. A non-transitory computer readable medium configured discovering a group of interdependent computing objects within an undirected graph structure in a computing environment, the device comprising: a memory; and a processor configured to: perform object identification by performing the steps comprising: perform relations identification by performing the steps comprising: make a selection by performing object intersection on the objects by performing the steps of: reading location and the relations for each of the objects; selecting a group including each of the objects in the sets of objects which are identical based on the relations for each of the objects; determining whether each of the objects in the group are related to each of the objects in each of the sets of objects, and determining whether at least one of the objects in each of the sets of objects is related to at least one of the objects in another set of the sets of objects.

11. The non-transitory computer readable medium of claim 10, the processor configured to: perform object identification by performing the steps comprising: receiving a search query including objects and operators; selecting an organization area; isolating object types for the objects in the search query; grouping the objects into sets of objects, and recording properties for each of the objects.

12. The non-transitory computer readable medium of claim 10, the processor configured to: perform relations identification by performing the steps comprising: recording relatability for each of the object types for the objects; recording relations for each of the objects.

13. The non-transitory computer readable medium of claim 10, the processor configured to: make a selection by performing object intersection on the objects by performing the steps of: dropping all of the objects from the group when each of the objects in the group are not related to each of the objects in each of the sets of objects.

14. The non-transitory computer readable medium of claim 10, the processor configured to: make a selection by performing object intersection on the objects by performing the steps of: dropping all of the objects from the group when each of the sets of objects is not related to at least one of the objects in another set of the sets of objects.

15. The non-transitory computer readable medium of claim 10, the processor configured to: make a selection by performing object intersection on the objects by performing the steps of: reading location and the relations for each of the objects, and selecting each of the objects that are in each of the sets of objects.

16. The non-transitory computer readable medium of claim 15, the processor configured to: make a selection by performing object intersection on the objects by performing the steps of: dropping each of the objects that are not in each of the sets of objects.

17. The non-transitory computer readable medium of claim 10, the processor configured to: make a selection by performing object intersection on the objects by performing the steps of: reading location and the relations for each of the objects; selecting each of the objects that are in each of the sets of objects, and determining whether each of the objects in a first set of objects is related to at least one the objects in a second set of objects.

18. The non-transitory computer readable medium of claim 17, the processor configured to: make a selection by performing object intersection on the objects by performing the steps of: dropping each of the objects in a first set of objects which is not related to at least one the objects in a second set of objects.

Description

[0018] In an embodiment of the present disclosure, the non-transitory computer readable medium, the processor configured to make a selection by performing object intersection on the objects by performing the steps of dropping all of the objects from the group when each of the objects in the group are not related to each of the objects in each of the sets of objects.

[0019] In an embodiment of the present disclosure, the non-transitory computer readable medium, the processor configured to make a selection by performing object intersection on the objects by performing the steps of dropping all of the objects from the group when each of the sets of objects is not related to at least one of the objects in another set of the sets of objects.

[0020] In an embodiment of the present disclosure, the non-transitory computer readable medium, the processor configured to make a selection by performing object intersection on the objects by performing the steps of reading location and the relations for each of the objects, and selecting each of the objects that are in each of the sets of objects.

[0021] In an embodiment of the present disclosure, the non-transitory computer readable medium, the processor configured to make a selection by performing object intersection on the objects by performing the steps of dropping each of the objects that are not in each of the sets of objects.

[0022] In an embodiment of the present disclosure, the non-transitory computer readable medium, the processor configured to make a selection by performing object intersection on the objects by performing the steps of reading location and the relations for each of the objects, selecting each of the objects that are in each of the sets of objects, and determining whether each of the objects in a first set of objects is related to at least one the objects in a second set of objects.

[0023] In an embodiment of the present disclosure, the non-transitory computer readable medium, the processor configured to make a selection by performing object intersection on the objects by performing the steps of dropping each of the objects in a first set of objects which is not related to at least one the objects in a second set of objects.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views, together with the detailed description below, are incorporated in and form part of the specification, and serve to further illustrate example embodiments of concepts found in the claims, and explain various principles and advantages of those embodiments.

[0025] These and other more detailed and specific features are more fully disclosed in the following specification, reference being had to the accompanying drawings, in which:

[0026] FIG. 1 illustrates a flowchart of the object identification process, the relations identification process and the making selection process of the current embodiment.

[0027] FIG. 2 illustrates the results of the object identification process, the relations identification process and the making selection process of the current embodiment.

[0028] FIG. 3 illustrates a flowchart for each of the three types of object selection, including object intersection, object union and object combination.

[0029] FIG. 4 illustrates the results of each of the three types of object selection, including object intersection, object union and object combination.

[0030] FIG. 5 illustrates a block diagram of the behavior of the object selection process.

[0042] A set is a selection of objects by their type. Each set may contain only objects of the same type.

[0043] A group is a selection of objects from multiple sets pertaining to one query. Each group can consist of objects from multiple sets, therefore, having objects of multiple types within the same group.

[0044] A sentence is an input for each configuration search (i.e., query). A sentence sets what objects by type and properties must be identified before query processing, how their relations will be recorded and what selectors (i.e., operations) will be made over them.

[0045] The embodiments described herein provide a method for finding all relevant objects, their types and value of properties. These objects are all known and are part of the same OA. Each object has one type and multiple properties different or the same from one another. Each object also determines to which other objects it may be related, to provide expected OA functionality.

[0046] There may be relatable objects and non-relatable objects.

[0047] Relatable objects (e.g., a network server connects with data stores which hold data) and non-relatable objects (e.g., a network client server which does not connect to the network server's data store, it only connects to the network server itself).

[0048] Relatability includes some differences to the objects relations.

[0049] For example, in the case of sets of relatable object types, the query returns an object only if each described set of objects of a given object type contains at least one object after the operation. If any of the sets are empty (i.e., there is not at least one object of each type in the set), the query returns an empty set of objects.

[0050] In the case of sets of non-relatable object types, sets of non-relatable objects cannot have any relations between each other. The query returns objects only if each described set of objects of a given object type contains at least one object after the operation. The sets of relatable objects must still fulfill the condition of being interconnected among each other.

[0051] In either case, the result of the whole query is not an empty set when all the objects within the group, which may be related to each other (logically or technically), are related. Only objects whose types cannot have a relation remain in the unrelated group.

[0052] The resulting set, after the query processing, is based on the mathematical operations of union (|) and intersection (&) of objects collections. The query addresses 2 . . . n objects of the OA and defines a logical structure to optimize the value search.

[0053] The resulting set is a set of objects sharing the same type or different types, which were used for the object identification.

[0054] Unlike standard mathematics operations, the intersection of an object resulting in a new set of resulting objects is not limited to a set of objects of the same type but may be altered to become a set of objects of different types and members of other sets, if they share logical interconnection. Therefore, a blend of objects of different types will be returned in case they form a logical entity within the OA that works in conjunction to provide a working item.

[0055] The operation (r-AND) of intersection to create resulting sets of objects that do not need to share the same type depending on whether logical links amongst those exist creates these resulting sets. The logical link is determined by the type of OA.

[0056] This embodiment may be applied to systems that have a large number of objects, a large number of

properties of the objects with various values that can be assumed, a large number of relations among objects, a high change rate of number of objects, relations and values of objects properties and a low number of types or objects which are steady in time.

[0057] This embodiment uses two sets of operations: $\&\text{.sub.R}$ (operation r-AND) and .andgate . which is a standard set union operator.

[0058] For example, there are five sets of objects, of which three sets are of type A and two sets are of type B: A.sub.1, A.sub.2, A.sub.3, B.sub.1, B.sub.2 where the goal is to intersect the sets: $\&\text{.sub.R A.sub.1} \&\text{.sub.R A.sub.2} \&\text{.sub.R B.sub.1} \&\text{.sub.R A.sub.3} \&\text{.sub.R B.sub.2}$ which would not be possible, because they are not of the same type and the result would not return the intersection. Therefore, a decomposition and relation must be performed before the standard intersection can be applied:

[0059] Below is an example of an evaluation of unrelated types (i.e., decomposition to objects of same types for standard set operations).

TABLE-US-00001 Sets of objects .fwdarw. Each set is a results of different queries Result Sentence A.sub.1 $\&\text{.sub.R A.sub.2} \&\text{.sub.R B.sub.1} \&\text{.sub.R A.sub.3} \&\text{.sub.R B.sub.2}$ Types of objects .dwnarw. Each set from the row above contains objects of the type in the rows below A a.sub.1, a.sub.2 .andgate. a.sub.1, a.sub.3 .andgate. .sub.---- .andgate. a.sub.1, a.sub.2, a.sub.5 .andgate. .sub.---- a.sub.1 B .sub.---- .andgate. .sub.---- .andgate. b.sub.1, b.sub.2, b.sub.5 .andgate. .sub.---- .andgate. b.sub.1, b.sub.2 b.sub.1, b.sub.2

[0060] The sentence evaluation result is a1, b1, b2.

[0061] Below is an example of an evaluation of related types (i.e., substitution of missing objects by related objects of the other types).

TABLE-US-00002 Sets of objects .fwdarw. Each set is a results of different queries Result Sentence A.sub.1 $\&\text{.sub.R A.sub.2} \&\text{.sub.R B.sub.1} \&\text{.sub.R A.sub.3} \&\text{.sub.R B.sub.2}$ Types of objects Since operations cannot contain objects of other types than its own, .dwnarw. relationships must be included to complete operations A a.sub.1, a.sub.2 .andgate. a.sub.1, a.sub.3 .andgate. a.sub.1, a.sub.2, a.sub.5 .andgate. a.sub.1, a.sub.2, a.sub.5 .andgate. a.sub.1, a.sub.2 a.sub.1 Implementing Relation of all objects of type A towards each related object of type relations B will be determined, and the other way round, filling the gaps in the other operation .uparw..dwnarw.. B b.sub.1, b.sub.2, b.sub.4 .andgate. b.sub.1 .andgate. b.sub.1, b.sub.2, b.sub.5 .andgate.b.sub.1, b.sub.2, b.sub.4, b.sub.5 .andgate. b.sub.1, b.sub.2 b.sub.1

[0062] The sentence evaluation result is a1, b1.

[0063] However, values of embedded systems' properties are not the only determinant of their state.

[0064] A manual configuration option may be used to determine objects parameters, where needed. meaning that in addition to autonomous discovery of identifiable objects, the final result may be influenced by the configuration area, system users, administrators, owners who may add manually defined properties and their values to each of the object. These manually added parameters may be treated the same way as objects' own, preset, properties into the formula.

[0065] These queries may be mathematically chained together (i.e., stacking the principle into layers encompassing different sets but computed within a single run), each query may work with different or identical objects within their own sets and combine the result with results of other queries, limiting the number of result objects and minimizing it to the least number sharing the searched parameters.

[0066] FIG. 1 illustrates a flowchart of the object identification process, the relations identification process and the making selection process of the current embodiment.

mounted but do not host any virtual server, are dropped.

[0112] For example, assuming the persons are members of set I, houses are in set II and cars in set III. Firstly, all detached houses and all SUV cars (regardless whether they ever park in the houses) are listed. Secondly, all females that inhabit a detached house and drive a SUV car, without ever using that car for coming to that house, stay. All females who only inhabit a detached house but drive no SUV or drive an SUV but live in other estate but a detached house, are dropped.

[0113] FIG. 4 illustrates the results of each of the three types of object selection, including object intersection, object union and object combination.

[0114] FIG. 5 illustrates a block diagram of the behavior of the object selection process.

[0115] There are three OAs illustrated in FIG. 5 which show for all of them how the object selection changes for groups if the relation of the object is changed.

[0116] For example, in FIG. 5, OA I 132 has objects that are relatable in all the selected sets. Set I.2 includes two selected objects of the group, the other sets each have one. If the relation represented by the edge I.5 is lost, the condition of all the objects in all sets must be interconnected is not fulfilled for the object I.6 and the object is then dropped. The condition of having at least one object of the selected group in each set still exists and it is met. The result will return all the other objects except I.6.

[0117] For example, in FIG. 5, OA II 133 have objects that are relatable in all the selected sets. If the relation II.5 is lost, the condition of all the objects in all sets must be interconnected is not fulfilled for the object II.6. The set II.4 would be empty after the object II.6 is removed from the group. Since the condition of having at least 1 object of the selected group in each set still exist and it is not met, all object of the group are dropped and the result is an empty set for this query.

[0118] For example, in FIG. 5, OA III 134 have three relatable types of object in sets III.1, III.2 and III.3. The type of object in set III.4 is not relatable to the type of set III.1. If the relation III.5 is lost the condition of having at least one object of the selected group in each set still exist and it is not met. All object of the group are dropped and the result is an empty set for this query. That means that even though there is no relation between III.1 and III.4, as it technically cannot be, the other objects must still be interconnected among them whenever a connection is technically possible to keep in the group.

[0119] The operator may be used for all purposes when finding relationships among objects of different types is required within a described system. The system may be heterogeneous (i.e., consisting of member of different technical and logical types).

[0120] A topology of the system must be determinable (i.e., relations and characteristics of the members must be known or retrievable). This may be either through information in the configuration database ("CMDB") or must be available by inquiring the systems' parameters.

[0121] The &.sub.R operator may be used to retrieve objects of interest. This is useful when large number of queries must be made over large number of sets of impacted objects. Results can be then retrieved in an efficient and fast manner. This can be used for a network connecting servers and switches across several data centers, providing services to multiple clients. Server, business applications, network ports, data centers, clients, support personnel may be types of objects.

[0122] For example, to find relations for all impacted ports by intervention of support personnel, the &.sub.R or r-AND operators (which are different notations of the same operator) may be used (e.g., ports, staff members and changing relationships between those two types).

[0123] The power of the method is multiplied with a growing number of queries to fetch systems' objects of

[0144] It should be appreciated by those skilled in the art that any blocks and block diagrams herein represent conceptual views of illustrative circuitry embodying the principles of the invention. Implementation of particular blocks can vary while they can be implemented in the hardware or software domain without limiting the scope of the invention. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudo code, and the like represent various processes which may be substantially represented in machine readable media and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

[0145] Accordingly, it is to be understood that the above description is intended to be illustrative and not restrictive. Many embodiments and applications other than the examples provided would be apparent upon reading the above description. The scope should be determined, not with reference to the above description or Abstract below, but should instead be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled. It is anticipated and intended that future developments will occur in the technologies discussed herein, and that the disclosed systems and methods will be incorporated into such future embodiments. In sum, it should be understood that the application is capable of modification and variation.

[0146] The benefits, advantages, solutions to problems, and any element(s) that may cause any benefit, advantage, or solution to occur or become more pronounced are not to be construed as a critical, required, or essential features or elements of any or all the claims. The invention is defined solely by the appended claims including any amendments made during the pendency of this application and all equivalents of those claims as issued.

[0147] All terms used in the claims are intended to be given their broadest reasonable constructions and their ordinary meanings as understood by those knowledgeable in the technologies described herein unless an explicit indication to the contrary is made herein. In particular, use of the singular articles such as "a," "the," "said," etc. should be read to recite one or more of the indicated elements unless a claim recites an explicit limitation to the contrary.

[0148] The Abstract of the Disclosure is provided to allow the reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, in the foregoing Detailed Description, it can be seen that various features are grouped together in various embodiments for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus the following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separately claimed subject matter.

* * * * *

