Abstract:

Title: IDENTIFICATION OF A SET OF OBJECTS BASED ON A FOCAL OBJECT

FIG. 1

(57) Abstract: In one implementation, a system for managing data includes a tag engine to maintain associations among objects of a data set, an associates engine to identify a first set of objects having a tag that matches a tag coupled to a focal object, a filter engine to identify a second set of objects based on the filter evaluated on the first set of objects, and a load engine to cause the second set of objects to load for display in a window of a UI.
identification of a Set of Objects based on a Focal Object

BACKGROUND

[0001] Data is ubiquitous. Data creation, storage, and analysis are common operations in information technology ("IT") industry and solutions for facilitating those operations are commonly developed for industry-specific or problem-specific purposes. For example, a database of IT service tickets can be maintained to discover common bugs in a system or a log of security alerts can be searched and/or analyzed to discover a system breach. Often, IT monitoring tools rely on strictly typed models that are populated with many details. One form of data consumption is through data exploration and/or navigation. For example, data sets may be presented in tabular form or aggregated as charts.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Figure 1 depicts an example user interface ("UP) in which various systems for managing data can be implemented.
[0003] Figures 2 and 3 are block diagrams depicting example systems for managing data.
[0004] Figure 4 depicts an example environment in which various systems for managing data can be implemented.
[0005] Figure 5 depicts example modules used to implement example systems for managing data.
[0006] Figures 6-8 are flow diagrams depicting example methods for managing data.

DETAILED DESCRIPTION
INTRODUCTION: in the following description and figures, some example implementations of systems and/or methods for managing data are described. A data management paradigm of the industry has shifted from "management by exception" to "collect everything and analyze on demand." Searching and browsing through large data sets can be ineffective due to the size of the data set and navigational functionality. The goal of assisting users to detect, identify, and resolve availably and performance issues of applications and IT infrastructures is constrained by user interface ("UI") workflows and unwanted data constraints.

Various examples described below relate to managing data in a semi-structured fashion to allow a user to walk through data based on tagged associations of data objects. By utilizing tags, data objects can be loaded for view and explored using gestures to find data objects associated with a designated focal object (i.e., an object designated as the focus of exploration). Gestures can allow for a user to observe a window of associated data objects and change the presented data by using a filter of tags and/or changing the focal object. In this manner, a user can effectively traverse large data sets in an intuitive and flexible form.

The following description is broken into sections. The first, labeled "Components," describes examples of physical and logical components for implementing various embodiments. The second, labeled "Environment," describes an example of a network environment in which various embodiments may be implemented. The third section, labeled "Operation," describes steps taken to implement various embodiments.

The terms "include," "have," and variations thereof, as used herein, mean the same as the term "comprise" or appropriate variation thereof. Furthermore, the term "based on," as used herein, means "based at least in part on." Thus, a feature that is described as based on some stimulus can be based only on the stimulus or a combination of stimuli including the stimulus. Furthermore, the term "maintain" (and variations thereof) as used herein means "to create, delete, add, remove, access, update, manage, and/or modify."

COMPONENTS: Figure 1 depicts an example UI 102 in which various systems 100 for managing data can be implemented. The example UI 102 is a graphical
Ui that provides visual representations of functionalities to allow a user to explore a
large data set with the ease of changing what part of the data set is being observed at
any given time using gestures.

The example UI 102 of Figure 1 is organized into five main areas. The first
area 104 (the area of the us 102 for traversing the data set) displays a portion of a data
set as objects 120 where one object is the focal object associated with the displayed
objects as designated at the focal object location 122. The first area 104 can include an
indication of the total number 124 of elements associated with the focal object and an
indication of the position 126 of the objects currently visible in the first area 104. The
objects 120 can be moved within the window 132 to show more elements of the results
that are hidden from view as indicated by the total number 124 and the position 126,
such as by using a clockwise or counterclockwise gesture 146. In some examples, the
outline of the window 132 may not be visible. In some examples, the objects 120 that
are associated with the focal object at location 122 can follow a reference arc 114
(which may or may not be visible). The reference arc 114 represents an expected path
and/or positions of the objects 120 being made visible at any given time. Limiting the
visibility of the data set aids the user in consuming the information of the data set as
well as the ability to move objects 120 in and out of the window 132.

Each object 120 is associated with at least one tag 130. For example, the
object detail area 106 displays the tags 130 associated with one of the objects 120. The
tags 130 can be used to determine the results shown in the window 132. In particular,
tags 130 can be added to a filter on the objects 120 associated with the focal object. For
example, a user can select the calendar icon 152 to create a tag associated with a time
period to filter the objects in the window to objects that have been updated during the
specified time period. The applied filters area 108 displays the tags 130 of the filter. The
tags 130 of the filter can be joined to narrow or broaden the results based on the toggle
134. Tags 130 that are used often, or otherwise may want to be remembered, can be
saved into the bookmark area 110 that displays a list of saved tags. For example, a user
can use the gesture 140 to move a tag 130 of an object from the object detail area 106
to the bookmark area 110. When the tag 130 of the object 120 is desired to be used in
the filter, the tag 130 can be dragged directly from the object details area 106 or from
the bookmark area 110 to the applied filters area 108 (e.g., using gesture 142). The indicator 128 associated with each visible object 120 can indicate how well the object 120 matches the filter.

[0014] The focal object can be searched for in the data set based on initialization parameters from the system 100, using the search box 136, or by pulling an object 120 from the window 132 to the focal object location 122 (e.g., using the gesture 144). A history of which objects 120 have been in the focal object location 122 are displayed in the history area 112. The history of the focal objects can be traversed (such as by using gesture 148) and a former focal object can be returned to the focal object location 122 by using a gesture 150 to select a former focal object 120 and dragging the former focal object 120 to the focal object location 122. The above collection of functionalities offered by the UI can provide flexible ability to traverse large data sets in a manner intuitive to the user.

[0015] Figures 2 and 3 are block diagrams depicting example systems 200 and 300 for managing data. Referring to Figure 2, the example system 200 of Figure 2 generally includes a data store 202, a tag engine 204, an associates engine 206, a filter engine 208, and a load engine 210. In general, the load engine 210 can provide objects associated with a focal object determined by the associates engine 206 and the filter engine 210 utilizing tags maintained by the tag engine 204 to associate objects of a data set. The example system 200 can include a filter match engine 212, a group engine 214, a detail engine 216, a gesture recognition engine 218, a bookmark engine 220, a history engine 222, and an anchor engine 224. In general, the engines 204-224 can facilitate exploration of a data set comprising objects capable of being tagged to represent associations with other objects of the data set.

[0016] The system 200 is useful to explore large data set and can be applied to large data sets that are semi-structured. For example, the partial structure of a data set can be a label associated with an attribute of the object or a pointer reference from one object to another. The partial structure, as discussed herein, is represented as tags that are coupled to objects of the data set. The tags represent associations of the data set. For example, the associations can be relationships, commonalities, attributes, designations, etc. that are able to be represented by tags. The tags are to be coupled.
with objects. For example, each object of a particular type can be tagged with a tag associated with designating the object type and each object created at a particular time may have a creation time-stamp tag that represents the date of creation of the object. Tags may be represented in any appropriate form such as a number, a value, a character, a string, a category, an icon, an image, or other identifier.

Referring to Figure 2, the tag engine 204 represents any circuitry or combination of circuitry and executable instructions to maintain associations among objects of a data set. The associations between objects can be explicit or implicit correlations among objects of the data sets. The associations can be determined and/or designated manually or automatically. The tag engine 204 can assist in maintaining the semi-structure of the data set to allow for the data set to be explored.

The semi-structured data set can be represented as a graph, where each object is represented as a node and each node can be connected by an edge to another node when both nodes are coupled with the same tag. The tag engine 204 can include circuitry or a combination of circuitry and executable instructions to generate a graph of the objects of the data set based on the tags. The graph can be a loosely-coupled graph where edges are created between objects based on associations (represented by tags) as the associations arise or as the objects are requested from the system 200. For example, data objects used by a particular user set may be connected to one another. For another example, data objects that are sub-objects or descendants of an object class may be connected to one another. Tags can be manually created to assist business analysis and the loose-coupling allows for changes to data objects to be found through chains of objects connected to one another. In this manner, a user may be able b "walk" (e.g., traverse) from one data object to another (e.g., a path comprising nodes and edges of the graph) based on tags at each object.

Exploration of the data set is assisted by identifying related objects. For example, if there is an error associated with a function (represented as an object), the error could be in the underlying data or sub-functions of the function, where the underlying data and sub-functions are represented as objects in the data set. When a data object is selected for exploration, the object is designated as the focal object. The objects associated with the focal objects are determined by the associates engine 206.
The associates engine 206 represents any circuitry or combination of circuitry and executable instructions to identify a set of objects having a first tag that matches a second tag coupled to the focal object. For example with reference to Figure 1, a first object of the objects of the data set can be designated as the focal object by pulling an object to the focal object location 122, and the window 132 can be filled with a first set of objects that are directly related to the focal object (i.e., the first set of objects are a subset of the data set that are coupled with a tag that is coupled to the focal object). The associates engine 206 can obtain an object identifier associated with the focal object and utilize the object identifier to discover objects that are directly related based on object identifiers that are associated with the same tag(s) as the object identifier of the focal object.

The filter engine 208 represents any circuitry or combination of circuitry and executable instructions to identify a set of objects based on a filter. A filter comprises a number of tags and the tags of the filter can be compared to the tags coupled to the objects of the data set. For example, a filter can at least include a tag of the focal object. For another example, a filter can include multiple associations that are taggable in the system 200, such as a data type, a pointer reference, or a time period. For example, a system log data set can assist in determining a problem discovered on a certain day, by filtering the objects associated with the focal object to only show data objects modified within 24 hours prior to the discovery of the problem.

The filter can be maintained by the filter engine 208 to change the scope of the relations to the focal object. For example, the resulting data set of the filter can be narrowed by adding another tag to the filter. The number of tags of the filter are maintainable based on gesture input, such as gesture input representing a change in a location of one of the number of tags on a UI. For example, a gesture to drag-and-drop a tag from one area of the UI to another area can move a tag in or out of the filter. The system 200 can provide for automatic refreshing of the resulting objects in the results window based on changes to the filter or focal object. For example, the filter engine 208 can initiate a narrowing of objects identified as associated with the focal object when a tag is added to the filter based on a drag-and-drop gesture of a tag on to the applied
filter area. In general, the filter engine 208 can identify a second set of objects based on
the filter evaluated on the first set of objects identified by the associated engine 206.

[0023] The filter engine 208 can also maintain whether a plurality of tags of the
filter are to broaden or narrow the data based on the relationship of the tags. The
relationship between the tags can be automatically determined or manually selected by
a user. For example, the filter may narrow the results to an intersection of results from a
first group of objects coupled to a first tag and a second group of objects coupled to a
second tag. For another example, the filter may provide results to a union of a first
group of objects coupled to a first tag and a second group of objects coupled to a
second tag. In this manner, the relationship between the tags (i.e., whether an
intersection or a union) can produce various results for view by the user.

[0024] The load engine 210 represents any circuitry or combination of circuitry
and executable instructions to cause the set of the objects to load for display in a
window of the UI. For example, the load engine 210 can designate the identified set of
objects to be loaded for display and prepare, at a data server (e.g., service device 434
of Figure 4), to transmit a subset of an identified set of objects located in a data store
(such as data store 202) that are associated with the focal object to a client device, such
as a user device 436 of Figure 4.

[0025] The load engine 210 can cause a different set of objects to load based on
how the data is explored. For example, the load engine 210 can cause a different set of
objects to load in a results window based on at least one of a change of the focal object
(e.g., pulling a different object to the focal object location) and a change of the filter
(e.g., removing or adding a tag to the filter).

[0026] The detail engine 212 represents any circuitry or combination of circuitry
and executable instructions to maintain a list of tags coupled with each object. For
example, the data structure representing an object can contain an array allocated to
store the tags associated with the object. An object can be coupled to multiple tags
directly and/or indirectly. For example, an object may not contain an explicit time-stamp
tag, but a tag applied to a certain time may identify the object based on metadata, such
as the creation date or modification date of the object. For another example, the types
of data of the object may be associated with multiple tables (or other data organization
structures) and, therefore, the object can be respectively coupled with a tag for each table. The tags associated with each object can be presented to a user to assist the user in finding other related objects. The detail engine 212 can also represent any circuitry or combination of circuitry and executable instructions to cause the details of the object (e.g. the tags coupled to the object) to be displayed based on the position of the object in the results window. For example, the detail engine 212 can track a position of an object in the results window and a particular position in the results window may be designated as the detail position where the details of an object in that position may be displayed in a details area of the UI.

[0027] The group engine 214 represents any circuitry or combination of circuitry and executable instructions to maintain host objects to represent groups of objects. For example, the group engine 214 can represent a combination of circuitry and executable instructions to determine whether a number of objects associated with a particular tag achieves a group threshold (i.e., a minimum number of objects to form a group) and, in response to the group threshold being achieved, create a dummy object coupled with the particular tag. In that example, a node can be created in a graph representing the data set and connections from the new node can be made to each of the objects of the group. In that manner, the number of objects are identified by the associates engine 404 when the dummy object is designated as the focal object.

[0028] The filter match engine 216 represents any circuitry or combination of circuitry and executable instructions to maintain a degree of conformity to a filter for each object. For example, a filter may contain multiple tags (e.g., likely a subset of all the tags associated with a data set) and a few objects of the data set may contain all the tags of the filter, but objects that include less than all the tags may still be relevant to the data exploration session. As such, a degree of conformity (e.g., the closeness or "best match" to the filter) can be determined (e.g., calculated based on a comparison of tags of an object to tags of the filter) and made presentable to the user on the subset of objects visible on the UI. The degree of conformity can be represented in any appropriate form, such as a number, a value, a character, a string, a category, or other identifier.
The gesture recognition engine 218 represents any circuitry or combination of circuitry and executable instructions to identify gestures performed on the UI. A gesture can be identified based on the location of input and action represented by the input. For example, the location of the input can be an area of the UI and an action can be a double-click on the area. Such input can be provided from a touch screen or a peripheral device, such as a computer mouse or motion detection system.

Gestures can include any appropriate form of interaction with a UI. For example, a gesture can be a drag-and-drop gesture to move an element of the UI to a different location, a double-tap gesture to select an element of the UI, a click-and-hold gesture to select an element of the UI, a rotational gesture to move an element of the UI in a circular direction or make a selection from a list, etc. For another example, a gesture along the path of the reference arc (e.g., reference arc 114 of Figure 1) can change the subset of objects displayed in the results window (e.g., move objects in from one side or the other until an end of the results is reached). Gestures can represent selection of a UI element, movement within a list, movement from one location to another, modification of an attribute, or other capabilities associated with data exploration.

The bookmark engine 220 represents any circuitry or combination of circuitry and executable instructions to maintain a list of tags designated as saved for potential use with the filter. For example, a data object may contain two tags, where the first tag can be sent to the filter to see more objects related to that tag and the second tag can be bookmarked (i.e., saved) to allow the user to come back to that tag to explore at a later time.

The history engine 222 represents any circuitry or combination of circuitry and executable instructions to maintain a log of objects that have been designated as the focal object in a data exploration session. For example, a user may begin to explore a path through the graph of the data set and then return back to a node in the graph by selecting a list of objects that have previously been selected to be the focal object.

The anchor engine 224 represents any circuitry or combination of circuitry and executable instructions to designate an object of the data set as the focal object. The focal object can be designated in a number of ways. For example, a link of an IT
ticket may be clicked to search through the data set related to the ticket and the information of the ticket can be used by a mechanism designated by the ticket system to determine the focal object and that designation can be sent to the system 200 via an application programming interface ("API"). The anchor engine 224 can maintain the changes in the focal object during a data exploration session. For example, the anchor engine 224 can be a combination of circuitry and executable instructions to maintain the focal object based on at least one of a first gesture that represents selection of an object from the results window or history area as the focal object and a search instruction that represents a retrieval of window or history area as the focal object and a search instruction that
represents a retrieval of the focal object form the data set based on a text entry.

[0034] The data store 202 can contain information utilized by the engines 204-224. For example, the data store 202 can store the data set, a tag, a filter, a graph, a log of the history of the focal objects, gesture information, etc.

[0035] Figure 3 depicts the example system 300 can comprise a memory resource 330 operatively coupled to a processor resource 328. The processor resource 328 can be operatively coupled to a data store 302. The data store 302 can be the same as the data store 202 of Figure 2.

[0036] Referring to Figure 3, the memory resource 330 can contain a set of instructions that are executable by the processor resource 328. The set of instructions are operable to cause the processor resource 328 to perform operations of the system 300 when the set of instructions are executed by the processor resource 328. The set of instructions stored on the memory resource 330 can be represented as a tag module 304, an associates module 306, a fitter module 308, a load module 310, a detail module 312, a group module 314, a filter match module 316, a gesture recognition module 318, a bookmark module 320, a history module 322, and an anchor module 324. The modules 304-324 represent program instructions that, when executed, function as the tag engine 204, the associates engine 206, the load engine 208, the filter engine 210, the detail engine 212, the group engine 214, the filter match engine 216, the gesture recognition engine 218, the bookmark engine 220, the history engine 222, and the anchor engine 224 of Figure 2, respectively. The processor resource 328 can carry out a set of instructions to execute the modules 304-324, and/or any other appropriate operations among and/or associated with the modules of the system 300. For example.
the processor resource 328 can carry out a set of instructions to designate a focal object (e.g., based on at least one of a gesture, an API input, and a text input), identify a set of objects related to the focal object based on a filter applied to the focal object, load the set of objects from a data store 302, and designate a subset of the set of objects to cause to be presented in a window of a UI. For another example, the processor resource 328 can carry out a set of instructions to track a position of a first object of the subset of the set of objects displayed in the results window and change the subset of visible objects based on at least one of a gesture performed on an area of the UI associated with the results window, a gesture to change the focal object, or a gesture to update the filter. For yet another example, the processor resource 328 can carry out a set of instructions to maintain a plurality of tags among the objects of the data set, generate a graph of the objects of the data set based on the tags, generate a dummy object based on a number of objects to achieve a group threshold, and cause the number of objects to load for display when the dummy object is designated as the focal object because the dummy object represents a parent node having the number of objects as descendants (e.g., related nodes from the selected parent node).

[0037] Although these particular modules and various other modules are illustrated and discussed in relation to Figure 3 and other example implementations, other combinations or sub-combinations of modules can be included within other implementations. Said differently, although the modules illustrated in Figure 3 and discussed in other example implementations perform specific functionalities in the examples discussed herein, these and other functionalities can be accomplished, implemented, or realized at different modules or at combinations of modules. For example, two or more modules illustrated and/or discussed as separate can be combined into a module that performs the functionalities discussed in relation to the two modules. As another example, functionalities performed at one module as discussed in relation to these examples can be performed at a different module or different modules.

Figure 5 depicts yet another example of how functionality can be organized into modules.

[0038] The processor resource 328 can be any appropriate circuitry capable of processing (e.g., computing) instructions, such as one or multiple processing elements.
capable of retrieving instructions from the memory resource 330 and executing those instructions. For example, the processor resource 328 can be a central processing unit ("CPU") that enables data management by fetching, decoding, and executing modules 304-324. Example processor resources 328 include at least one CPU, a semiconductor-based microprocessor, an application specific integrated circuit ("ASiC"), a field-programmable gate array ("FPGA"), and the like. The processor resource 328 can include multiple processing elements that are integrated in a single device or distributed across devices. The processor resource 328 can process the instructions serially, concurrently, or in partial concurrence.

[0039] The memory resource 330 and the data store 302 represent a medium to store data utilized and/or produced by the system 300. The medium can be any non-transitory medium or combination of non-transitory mediums able to electronically store data, such as modules of the system 300 and/or data used by the system 300. For example, the medium can be a storage medium, which is distinct from a transitory transmission medium, such as a signal. The medium can be machine-readable, such as computer-readable. The medium can be an electronic, magnetic, optical, or other physical storage device that is capable of containing (i.e., storing) executable instructions. The memory resource 330 can be said to store program instructions that when executed by the processor resource 328 cause the processor resource 328 to implement functionality of the system 300 of Figure 3. The memory resource 330 can be integrated in the same device as the processor resource 328 or it can be separate but accessible to that device and the processor resource 328. The memory resource 330 can be distributed across devices. The memory resource 330 and the data store 302 can represent the same physical medium or separate physical mediums. The data of the data store 302 can include representations of data and/or information mentioned herein.

[0040] In the discussion herein, the engines 204-224 of Figure 2 and the modules 304-324 of Figure 3 have been described as circuitry or a combination of circuitry and executable instructions. Such components can be implemented in a number of fashions. Looking at Figure 3, the executable instructions can be processor-executable instructions, such as program instructions, stored on the memory resource 330, which is
a tangible, non-transitory computer-readable storage medium, and the circuitry can be electronic circuitry, such as processor resource 328, for executing those instructions. The instructions residing on the memory resource 330 can comprise any set of instructions to be executed directly (such as machine code) or indirectly (such as a script) by the processor resource 328.

[0041] In some examples, the system 300 can include the executable instructions can be part of an installation package that when installed can be executed by the processor resource 328 to perform operations of the system 300, such as methods described with regards to Figures 5-8. In that example, the memory resource 330 can be a portable medium such as a compact disc, a digital video disc, a flash drive, or memory maintained by a computer device, such as a service device 434 of Figure 4, from which the installation package can be downloaded and installed. In another example, the executable instructions can be part of an application or applications already installed. The memory resource 330 can be a non-volatile memory resource such as read only memory ("ROM"), a volatile memory resource such as random access memory ("RAM"), a storage device, or a combination thereof. Example forms of a memory resource 330 include static RAM ("SRAM"), dynamic RAM ("DRAM"), electrically erasable programmable ROM ("EEPROM"), flash memory, or the like. The memory resource 330 can include integrated memory such as a hard drive ("HD"), a solid state drive fSSD"), or an optical drive.

[0042] ENVIRONMENT: Figure 4 depicts example environments in which various example systems 400 can be implemented. The example environment 490 is shown to include an example system 400 for managing data. The system 400 (described herein with respect to Figures 2 and 3) can represent generally any circuitry or combination of circuitry and executable instructions to manage a data set. The system 400 can include a tag engine 404, an associates engine 406, a filter engine 408, a load engine 410, a detail engine 412, a group engine 414, a filter match engine 416, a gesture recognition engine 418, a bookmark engine 420, a history engine 422, and an anchor engine 424 that are the same as the tag engine 204, the associates engine 206, the filter engine 208, the load engine 210, the detail engine 212, the group engine 214, the filter match engine 216, the gesture recognition engine 218, the bookmark engine 220, the history
engine 222, and the anchor engine 224 of Figure 2, respectively, and the associated
descriptions are not repeated for brevity. The system 400 can also include a UI engine
426 representing any circuitry or combination of circuitry and executable instructions to
communicate information from the system 400 to the UI, such as the UI presented via a
browser on a user device 438. As shown in Figure 4, the engines 404-426 can be
integrated into a compute, such as a service device. The engines 404-426 can be
integrated via circuitry or as installed instructions into a memory resource of the
compute.

[0043] The example environment 490 can include compute devices, such as
developer devices 432, service devices 434, and user devices 436. A first set of
instructions can be developed and/or modified on a developer device 432 and
associated with a data set 440 of objects. For example, an application can be
developed and modified on a developer device 432 and stored onto a web server, such
as a service device 434, and the data set 440 associated with the application can be
stored in a data store 402 on a data server, such as a service device 434. The service
devices 434 represent generally any compute devices to respond to a network request
received from a user device 436, whether virtual or real. For example, the service
device 434 can operate a combination of circuitry and executable instructions to provide
a network packet in response to a request for a page or functionality of an application.
The user devices 436 represent generally any compute devices to communicate a
network request and receive and/or process the corresponding responses. For
example, a browser application may be installed on the user device 436 to receive the
network packet from the service device 434 and utilize the payload of the packet to
display an element of a page via the browser application.

[0044] The compute devices can be located on separate networks 430 or part of
the same network 430. The example environment 490 can include any appropriate
number of networks 430 and any number of the networks 430 can include a cloud
compute environment. A cloud compute environment may include a virtual shared pool
of compute resources. For example, networks 430 can be distributed networks
comprising virtual computing resources. Any appropriate combination of the system 400
and compute devices can be a virtual instance of a resource of a virtual shared pool of
resources. The engines and/or modules of the system 400 herein can reside and/or execute "on the cloud" (e.g. reside and/or execute on a virtual shared pool of resources).

[0045] A link 438 generally represents one or a combination of a cable, wireless connection, fiber optic connection, or remote connections via a telecommunications link, an infrared link, a radio frequency Sink, or any other connectors of systems that provide electronic communication. The link 438 can include, at least in part, intranet, the Internet, or a combination of both. The link 438 can also include intermediate proxies, routers, switches, load balancers, and the like.

[0046] Referring to Figures 2-4, the engines 204-224 of Figure 2 and/or the modules 304-324 of Figure 3 can be distributed across devices 432, 434, 436, or a combination thereof. The engines and/or modules can complete or assist completion of operations performed in describing another engine and/or module. For example, the associates engine 406 of Figure 4 can request, complete, or perform the methods or operations described with the associates engine 206 of Figure 2 as well as the tag engine 204, the filter engine 208, and the load engine 210 of Figure 2. Thus, although the various engines and modules are shown as separate engines in Figures 2 and 3, in other implementations, the functionality of multiple engines and/or modules may be implemented as a single engine and/or module or divided in a variety of engines and/or modules. In some example, the engines of the system 400 can perform example methods described in connection with Figures 5-8.

[0047] OPERATION: Figure 5 depicts example modules used to implement example systems for data management. Referring to Figure 5, the example modules of Figure 5 generally include a tag module 504, an anchor module 524, an associates module 506, a filter module 508, and a load module 510. The example modules of Figure 5 can be implemented on a computer device, such as a service device 434.

[0048] A data request 562 can be received and queued by the system and cause the data set 564 to be retrieved. The tag module 504 represents program instructions that when executed by a processor resource cause the data set 564 to be maintained with tags coupled to objects of the data set 564. The tag module 504 can include program instructions, such as a correlation module 540 and an assign module 542, to
assist in maintaining the data set 564. For example, the correlation module 540 can represent program instructions that, when executed, cause a processor resource to identify a relationship between two data objects of the data set 584 and the assign module 542 can represent program instructions that, when executed, cause the processor resource to assign a tag to each object associated with the relationship.

[0049] The data request 562 can cause an anchor module 524 to execute and, in turn, cause a processor resource to appoint (e.g., designate) a focal object based on a selection gesture 566. The anchor module 524 can include program instructions, such as a designation module 544 and an attribute module 546, to assist in appointing the focal object. For example, the designation module 544, when executed, can cause a processor resource to identify the object to be designated as the focal object, such as by setting a flag of an object based on the selection gesture 566. For another example, the attribute module 546, when executed, can cause a processor resource to identify an attribute 570 of the object designated by the selection gesture 566. The identified attribute 570 can be used by a processor resource executing the associates module 506 to produce a number of results (e.g., a set of objects of the data set related to the focal object based on the attribute 570). The associates module 506 can include program instructions, such as a focus module and a connection module 550, to assist in identifying associates of the focal object 568. For example, the focus module 548, when executed, can cause a processor resource to search for a node of a graph associated with the focal object 568 and the connection module 550, when executed, can cause a processor resource to follow an edge of the graph associated with the attribute 570 to another node. The related node can be added to a list of results to potentially load and cause to be presented via a UI.

[0050] The filter module 508 can include program instructions, such as a pool module 552 and a relationship module 554, to assist in filtering the objects identified (i.e., the associate objects 572) by the processor resource executing the associates module. For example, the pool module 552, when executed, can cause a processor resource to identify the tags of the associates 572, and the relationship module 554, when executed, can identify which tags are also located within the tag list 574 of the filter and filter relationship (e.g., whether the tag list 574 are joined as a union or
intersection) and, based on the comparison of each associated tag set to the tag list
574, determine a subset of the set of associates that fit the filter.

[0051] The load module 510 can include program instructions (such as a visible
module 556, a hidden module 558, and a provision module 560) to prepare the filtered
associates. For example, the visible module 556 represents program instructions that
when executed cause a processor resource to designate which subset of the associates
are to be visible on the UI based on a data walk gesture 576. A data walk gesture 576
represents the input that causes an instruction of traversing the data to identify a
different subset to view on the UI. For another example, the hidden module 558
represents program instructions that when executed cause a processor resource to
track which objects are to be hidden from view on the UI (i.e., hidden from presentation
but still load the objects for potential view by the user based on the data walk gesture
576). For yet another example, the provision module 560 represents program
instructions that when executed cause a processor resource to prepare a data
transmission 580 (i.e., prepare data packets) to a user device (such as user device 436
of Figure 4) based on an APS 578 for interacting with the UI (i.e., interacting with the
browser on the user device).

[0052] Figures 6-8 are flow diagrams depicting example methods for managing
data. Referring to Figure 6, example methods for managing data can generally
comprise appointing an object of a data set as a focal object, identifying a set of objects
of the data set that are related to the focal object based on a filter applied to the data
set, and determining a subset of the set of objects as visible for presentation within a
results window of a UI.

[0053] At block 602, a first object of a data set is appointed as a focal object. The
first object can be appointed by a text search, a gesture on the UI, or an external
mechanism in communication with the data management system via an API. At block
604, a set of objects that are related to the focal object are identified based on a filter.
The objects of the data set can be associated with a plurality of tags and the filter can
be a subset of the plurality of tags. The tags can be represented as edge connections of
a graph of the data objects, and the determination of the set of objects can be based on
following the edges of the focal object that represent the tags.
At block 608, a first subset of the set of objects can be determined as visible for presentation within a results window of a UI. In relation to that, a second subset of objects can be determined as hidden from presentation. The second subset can be a mathematical complement to the first subset where the second subset of the set of objects are hidden from presentation in the UI, but have been loaded for potential view. The visible objects can be designated by a flag or order recognized by the user device upon receipt of the list of objects. The objects designated as hidden can be at least partially loaded to allow for ready visibility by the user based on gestures on the UI.

Figure 7 includes blocks similar to blocks of Figure 6 and provides additional blocks and details. In particular, Figure 6 depicts additional blocks and details generally regarding maintaining tags among the data set, generating a graph of the data set, and designating a membership change of the visible subset. Blocks 704, 710, and 714 are the same as blocks 602, 604, and 606 of Figure 6 and, for brevity, their respective descriptions are not repeated.

At block 702, tags are maintained among the data set. As mentioned herein, the tags may be maintained automatically (such as by determining host groups), via a system of the data set, or manually by a user or system of the data set. A graph of the data set can be generated from the relationships among the objects of the data set represented by tags at block 706. The semi-structure of the data set can be utilized to generate a graph of nodes representing objects of the data set. Thus, when a focal object is selected at block 705, a first node representing the focal object can be investigated (e.g., discover a node connected to another node via an edge) for a second node that is related to the first node based on an edge connection representing a tag.

At block 712, the set of objects can be supplied from the data store to a user device and designated as a set of results identified as related to the focal object based on the filter. At block 716, the user can traverse the set of objects by designating an object to change membership from the hidden complementary set of supplied objects to the visible subset of objects and/or designating an object to change membership from the visible subset of objects to the hidden complementary set. The UI
can update as the membership changes between the group of visible objects and the
group of hidden objects.

[0058] Figure 8 includes blocks similar to blocks of Figure 7 and provides
additional blocks and details. In particular, Figure 7 depicts additional blocks and details
generally regarding maintaining tags among the data set, identifying a set of objects
related to the focal object, and determining a subset of the set of objects as visible.
Blocks 802, 804, 806 are similar b blocks 702, 710, and 714 of Figure 7 and, for
brevity, their respective descriptions have not been repeated in their entirety.

[0059] At block 802, tags are maintained among the data set by identifying a
threshold number of objects that are related at block 810 and, in response to the
threshold being achieved, generating a host object (i.e., dummy object) of a number of
objects of the data set based on the tag at block 812. For example, the number of
objects are assigned to a first tag of a plurality of tags and the number is identifying as
achieving a predetermined threshold, where the threshold may be predetermined based
on the number at which creating an indirection in the data coupling provides for
improved user intuition to walk through the semi-structured data. In that example, the
number of objects can be designated as descendants of the focal objects when the host
object is appointed as the focal object. In this manner, the host objects are to assist a
user as the user explores the data by providing object groups in a hierarchy or other
organizational form that may be instinctive and/or instructive to the user.

[0060] At block 804, a graph may be generated and utilized to identify a set of
objects that are related to the focal object. At block 820, a graph of the data set is
generated from relationships among the objects of the data set. For example, each
object may be related to at least one other objects, as represented by an edge in the
graph coupled to each related object. In this manner, the graph can represent a loose
coupling of the objects based on the plurality of tags (e.g., string representations of the
relationships). A first node of the graph that represents the focal object can be selected
at block 822 in response to a gesture that causes the focal object to be appointed. At
block 824, a second node is determined as related to the first node and the object
associated with the second node can be designated to be loaded for display in the
results window when the first node is selected and the second node passes the filter
because the first node is connected to the second node on the graph and the object of
the second node has at least one attribute prescribed by the filter.

[0061] At block 806, a subset of the objects are determined as visible. Visibility in
the user window can be determined based on an order of the set of objects, which may,
in turn, be based on the degree of conformity of the set of objects to the filter. For
example, at block 830, a plurality of degrees of conformity of the set of objects to the
filter are identified. The plurality of degrees of conformity may represent any degree of
conformity between a range of no match of any tags of the filter to an exact match of all
the tags of the filter. The conformity of an object of the filter can be based on a
comparison of the tags of the filter to the tags (and/or attributes) of the objects. At block
832, the set of objects can be ordered based on the plurality of degrees of conformity.
For example, the objects identified as having the highest degrees of conformity (e.g.,
highest number of tags matched when compared to the tags of the filter) can be placed
at the beginning of an order of the set of objects (e.g., the visible subset) and the
objects identified has having the least degrees of conformity (e.g., the lowest number of
tags match when compared to the tags of the filter) can be placed at the end of the
order. in this manner, the best matched results may be initially presented to the user to
allow the user to quickly explore the most-related objects of the data set before
exploring less-related objects of the data set.

[0062] CONCLUSION: Although the flow diagrams of Figures 5-8 illustrate specific
orders of execution, the order of execution may differ from that which is illustrated. For
example, the order of execution of the blocks may be scrambled relative to the order
shown. Also, the blocks shown in succession may be executed concurrently or with
partial concurrence. All such variations are within the scope of the present description.

[0063] The present description has been shown and described with reference to
the foregoing examples. It is understood, however, that other forms, details, and
examples may be made without departing from the spirit and scope of the following
claims. The use of the words "first," "second," or related terms in the claims are not used
to limit the claim elements to an order or location, but are merely used to distinguish
separate claim elements.
CLAIMS

What is claimed is:

1. A system comprising:
   a tag engine to maintain associations among objects of a data set, the
   associations represented by tags coupled with the objects;
   an associates engine to identify a first set of the objects having a first tag that
   matches a second tag coupled to the focal object, a first object of the objects
   designated as the focal object;
   a filter engine to:
      maintain a filter comprising a number of tags, the number of tags
      maintainable based on gesture input representing a change in location of one of
      the number of tags on a user interface; and
      identify a second set of objects based on the filter evaluated on the first set of
      objects; and
   a load engine to cause the second set of the objects to load for display in a
   window of the user interface.

2. The system of claim 1, wherein the data set is semi-structured, the system
   comprising:
   a details engine to maintain a list of tags coupled with each object; and
   a group engine to;
   determine whether a number of objects associated with a third tag
   achieves a group threshold; and
   in response to the group threshold being achieved, create a dummy object
   coupled with the third tag, the number of objects to be identified by the
   associates engine when the dummy object is designated as the focal object.

3. The system of claim 1, comprising:
   a filter match engine to maintain a degree of conformity to the filter for each
   object of the second set of objects, wherein the second set of objects results from an
application of the filter on the first set of objects based on a relationship of the
number of tags, the relationship being one of a union and an intersection.

4. The system of claim 1, comprising:
   a bookmark engine to maintain a list of tags designated as saved for potential
   use with the filter.

5. The system of claim 1, comprising:
   a gesture recognition engine to identify gestures performed on the user interface;
   and
   an anchor engine to maintain the focal object based on at least one of:
   a first gesture that represents selection of one of the second set of objects
   from the window as the focal object; and
   a search instruction that represents a retrieval of the focal object from the
   data set based on a text entry:
   wherein:
   the filter engine is to add a tag to the filter based on a second gesture; and
   the load engine is to cause a different set of objects to load based on at
   least one of:
   a change of the focal object: and
   a change of the filter.

6. A computer-readable storage medium comprising a set of instructions executable by
   a processor resource to:
   designate a focal object based on at least one of a first gesture, an application
   programming interface input, and a text input;
   identify a set of objects related to the focal object based on a filter applied to the
   focal object, the filter to contain a set of tags associated with the set of objects;
   toad the set of objects from a data store; and
designate a subset of the set of objects to cause to be presented in a window of
a UI, the subset of the set of objects selected based on a second gesture performed
on an area of the user interface associated with the window.

7. The medium of claim 6, wherein the set of instructions is executable by the
processor resource to:
change the subset of objects based on gesture input, the gesture input to include
at least one of the first gesture and the second gesture; and
track a position of the first object of the subset in the set of objects.

8. The medium of claim 7, wherein:
the first gesture includes at least one of:
a first drag-and-drop gesture to move a first object of the set of objects
from the window to a focal point location of the user interface; and
a second drag-and-drop gesture to move a second object of a history of
focal point objects from a history area of the user interface to a focal point
location of the user interface; and
the second gesture includes at least one of:
a rotational gesture to change a position of the first object in the window;
or
a third drag-and-drop gesture to add a tag to the filter, the set of objects to
reduce the data set to the subset of the set of objects that include the tag.

9. The medium of claim 6, wherein the set of instructions is executable by the
processor resource to:
maintain a plurality of tags among the objects of a data set, the tags to represent
relationships among the objects and the set of tags being a subset of the plurality of
tags; and
generate a graph of the objects of the data set based on the tags.
10. The medium of claim 9, wherein the set of instructions is executable by the processor resource to:
   generate a dummy object based on a number of objects to achieve a group threshold, the dummy object to represent a parent node having the number of objects as descendants; and
   cause the number of objects to load for display when the dummy object is designated as the focal object.

11. A method for managing data comprising:
   appointing, in response to a first gesture, a first object of a data set as a focal object;
   identifying a set of objects of the data set that are related to the focal object based on a filter applied to the data set, the objects of the data set associated with a plurality of tags and the filter to be a subset of the plurality of tags; and
   determining a first subset of the set of objects as visible for presentation within a results window of a user interface, the first subset being complementary to a second subset of the set of the objects that are hidden from presentation.

12. The method of claim 11, comprising, in response to a second gesture:
   designating a second object to change membership from the complementary set to the subset; and
   designating a third object to change membership from the subset to the complementary set.

13. The method of claim 11, comprising:
   maintaining the plurality of tags among the data set in a data store;
   supplying the set of objects from the data store as a set of results identified as related to the focal object based on the filter;
   generating a graph of the data set from relationships among the objects of the data set, the graph to represent a loose coupling of the objects based on the plurality of tags;
selecting, in response to the first gesture, a first node of the graph that
represents the first object appointed as the focal object; and
designating a second object to be loaded for display in the results window when
the first node is connected to the second node and the second node passes the
filter, the second object represented by a second node of the graph.

14. The method of claim 11, comprising:
identifying a plurality of degrees of conformity of the set of the objects to the
filter; and
ordering the set of objects based on the plurality of degrees of conformity.

15. The method of claim 14, comprising:
identifying a threshold number of objects assigned to a first tag of the plurality of
tags is achieved;
generating, in response to the threshold being achieved, a host object of a
number of objects of the data set based on the first tag; and
designating the number of objects as the descendants when the host object is
appointed as the focal object.
FIG. 4
FIG. 5
5/6

602 ~ APPOINT A FIRST OBJECT OF A DATA SET AS A FOCAL OBJECT

604 ~ IDENTIFY A SET OF OBJECTS ARE RELATED TO THE FOCAL OBJECT BASED ON A FILTER

606 ~ DETERMINE A SUBSET OF THE SET OF OBJECTS AS VISIBLE FOR PRESENTATION WITHIN A RESULTS WINDOW OF A USER INTERFACE AND THE COMPLEMENTARY SET OF OBJECTS AS HIDDEN FROM PRESENTATION

FIG. 6

702 ~ MAINTAIN TAGS AMONG A DATA SET

704 ~ APPOINT A FIRST OBJECT OF A DATA SET AS A FOCAL OBJECT

706 ~ GENERATE A GRAPH OF THE DATA SET FROM RELATIONSHIPS AMONG THE OBJECTS OF THE DATA SET

708 ~ SELECT A FIRST NODE OF THE GRAPH THAT REPRESENTS THE FOCAL OBJECT

710 ~ IDENTIFY A SET OF OBJECTS ARE RELATED TO THE FOCAL OBJECT BASED ON A FILTER

712 ~ SUPPLY THE SET OF OBJECTS FROM THE DATA STORE AS A SET OF RESULTS IDENTIFIED AS RELATED TO THE FOCAL OBJECT BASED ON THE FILTER

714 ~ DETERMINE A SUBSET OF THE SET OF OBJECTS AS VISIBLE FOR PRESENTATION WITHIN A RESULTS WINDOW OF A USER INTERFACE AND THE COMPLEMENTARY SET OF OBJECTS AS HIDDEN FROM PRESENTATION

716 ~ DESIGNATE A MEMBERSHIP CHANGE

FIG. 7
Maintain tags among the data set in a data store

1. Identify a threshold number of objects
2. Generate a dummy object of a number of objects of the data set based on the tag

Identify a set of objects of the data set that are related to focal object

3. Generate a graph of the data set from relationships among the objects of the data set
4. Select a first node of the graph that represents the focal object
5. Designate a second object to be loaded for display

Determine a subset of the set of objects as visible

6. Identify a plurality of degrees of conformity of the set of the objects to the filter
7. Order the set of objects based on the plurality of degrees of conformity

**Fig. 8**
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30(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 17/30; G06F 15/16

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: tag, filter, gesture, association, data set, focal object, match, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US 2012-0221567 A1 (CHARLES A. FINNEAST et al.) 30 August 2012</td>
<td>1, 3, 6, 11, 14</td>
</tr>
<tr>
<td>A</td>
<td>See paragraphs [0016]–[0053] and figures 1A–6.</td>
<td>2, 4, 5, 7–10, 12, 13, 15</td>
</tr>
<tr>
<td>Y</td>
<td>US 2012-0109986 A1 (MICHAEL F. PALEMBITI, 11) 03 May 2012</td>
<td>1, 3, 6, 11, 14</td>
</tr>
<tr>
<td>A</td>
<td>See paragraphs [0027]–[0065] and figures 1–10.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>See paragraphs [0020]–[0071] and figures 2–8D.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>See paragraphs [0030]–[0062] and figures 2–4.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>See paragraphs [0068]–[0084] and figures 3–5.</td>
<td></td>
</tr>
</tbody>
</table>

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
30 June 2015 (30.06.2015)

Date of mailing of the international search report
30 June 2015 (30.06.2015)

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongna-ro, Seo-gu, Daegu Metropolitan City, 302-701, Republic of Korea
Facsimile No. +82-42-472-7140

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

Form PCT/ISA/210 (second sheet) (January 2015)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>US 2010-325581 A1</td>
<td>23/12/2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 7792868 B2</td>
<td>07/09/2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 8195675 B2</td>
<td>05/06/2012</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 8533205 B2</td>
<td>10/09/2013</td>
</tr>
<tr>
<td>US 2012-0109986 A1</td>
<td>03/05/2012</td>
<td>US 2015-095850 A1</td>
<td>02/04/2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 8892578 B2</td>
<td>18/11/2014</td>
</tr>
<tr>
<td>US 2013-0166550 A1</td>
<td>27/06/2013</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>US 2013-0346405 A1</td>
<td>26/12/2013</td>
<td>GB 2504610 A</td>
<td>05/02/2014</td>
</tr>
</tbody>
</table>