

USER-DEFINED GROUPS OF GRAPHICAL OBJECTS

FIELD

The embodiments discussed herein are related to user-defined groups of graphical objects.

5 BACKGROUND

Unless otherwise indicated herein, the materials described herein are not prior art to the claims in the present application and are not admitted to be prior art by inclusion in this section.

Some structured source data, e.g., of networks, organizations, or other structured
10 source data, lends itself to representation in the form of a diagram or other visualization, such as a network diagram or an organizational chart. Some diagram applications allow users to generate graphical diagrams on a computer based on the structured source data. Such graphical diagrams may include graphical objects that represent data objects, resources, or nodes in the structured source data.

15 The subject matter claimed herein is not limited to implementations that solve any disadvantages or that operate only in environments such as those described above. Rather, this background is only provided to illustrate one example technology area where some implementations described herein may be practiced.

SUMMARY

20 This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential characteristics of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

In an example embodiment, a computer-implemented method to group graphical
25 objects includes displaying, on a display device, a graphical diagram with multiple graphical objects that represent data of a data source. The method includes receiving input to define one or more groups. The method includes, in response to the input, generating one or more containers, each of the one or more containers representing a different one of the one or more groups; and graphically depicting membership of the graphical objects in
30 the one or more groups by relative arrangement of the graphical objects and the one or more containers according to group membership of each of the graphical objects.

In another example embodiment, a non-transitory computer-readable medium has computer-readable instructions stored thereon that are executable by a processor to perform or control performance of operations. The operations include displaying, on a display

device, a graphical diagram with multiple graphical objects that represent data of a data source. The operations include receiving input to define one or more groups. The operations include, in response to the input, generating one or more containers, each of the one or more containers representing a different one of the one or more groups; and graphically depicting
5 membership of the graphical objects in the one or more groups by relative arrangement of the graphical objects and the one or more containers according to group membership of each of the graphical objects.

Additional features and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be
10 learned by the practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

15 BRIEF DESCRIPTION OF THE DRAWINGS

To further clarify the above and other advantages and features of the present invention, a more particular description of the invention will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. It is appreciated that these drawings depict only typical embodiments of the invention and are
20 therefore not to be considered limiting of its scope. The invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

FIG. 1 is a block diagram of an example operating environment that includes a server and one or more client devices;

25 FIGS. 2A-2D include graphical representations of a user interface (UI) to generate graphical diagrams;

FIGS. 3A-3J include graphical representations of other UIs to generate graphical diagrams;

FIG. 4 illustrates a flowchart of an example method to group graphical objects that
30 may be implemented in the operating environment of FIG. 1 and/or the UIs of FIGS. 2A-3J; and

FIG. 5 is a block diagram illustrating an example computing device 500 that is arranged to generate and/or display graphical diagrams,

all arranged in accordance with at least one embodiment described herein.

DETAILED DESCRIPTION OF SOME EXAMPLE EMBODIMENTS

Some users may desire to group various data objects, nodes, or resources of structured source data together even if they are unrelated in the structured source data. For example, some EC2 instances in AWS structured source data may be part of the same subnet but may serve completely different applications. While the structured source data may reflect the inclusion of the EC2 instances in the same subnet and a suitable diagram application may illustrate the EC2 instances as belonging to the same subnet (or more particularly may illustrate graphical objects representing the EC2 instances within a container graphical object representing the subnet), the structured source data may fail to reflect the relationships between the EC2 instances and the applications they serve and the diagram application may be unable to show that the EC2 instances serve different applications without the relationships in the structure source data.

Some embodiments herein may permit users to define groups, hereinafter user-defined groups, of resources or other entities, or more particularly of graphical objects that represent resources or other entities. For example, a user may define groups based on one or more existing properties, or attributes, of the graphical objects. The attributes may include tags that may be system-defined or user-defined. In this example, a user may define one or more groups by selecting an attribute to group on. The system may identify one or more unique values of the selected attribute, may generate a different container for each unique value, and may group the graphical objects into one or more resulting containers according to the value each graphical object has for the selected attribute. As another example, the user may define a group by selecting a single graphical object or simultaneously selecting multiple graphical objects and selecting an option in a UI to group the selected graphical object(s) together. In this example, the system may generate a container and arrange the selected graphical objects in the container.

Reference will now be made to the drawings to describe various aspects of example embodiments of the invention. It is to be understood that the drawings are diagrammatic and schematic representations of such example embodiments, and are not limiting of the present invention, nor are they necessarily drawn to scale.

FIG. 1 is a block diagram of an example operating environment 100 that includes a server 102 and one or more client devices 104, 106, 108, arranged in accordance with at least one embodiment described herein. The server 102 and/or the client devices 104, 106, 108 may be configured to generate and display graphical diagrams or visualizations that

include graphical objects based on structured source data. The terms graphical diagram and visualization are used interchangeably herein. In some embodiments, users may be able to create user-defined groups of graphical objects in graphical diagrams.

The structured source data may be received from one or more data sources 110. Each of the data sources 110 may include, for example, a cloud infrastructure provider such as AWS, GOOGLE CLOUD, or MICROSOFT AZURE; a human resources (HR) system or database such as BAMBOOHR or WORKDAY; a customer relationship management (CRM) system such as SALESFORCE.COM or MICROSOFT DYNAMICS CRM; a spreadsheet for HR, e.g., with names, pictures, and/or a reporting structure for employees of an entity; a custom internal logging system; a floorplan database (e.g., with employee office/workstation assignments); a database schema (e.g., to generate Entity Relationship Diagrams); a service-oriented architecture; physical network discovery tools; custom-built technical systems that can be modeled similar to a network; or other suitable data source and/or data collection. The data in each of the data sources 110 may have a specific format, which may be different and/or the same from one data source 110 to another. Some example data formats include proprietary formats, Structured Query Language (SQL), and JavaScript Object Notation (JSON).

The structured source data may identify two or more nodes, resources, and/or other entities and relationships among them. For instance, the structured source data may identify various network resources such as compute nodes, storage nodes, database nodes, networking nodes (e.g., virtual private clouds (VPCs)), regions, availability zones (AZs), and/or other resources deployed by a customer in AWS as well as relationships among the entities, such as specific ports over which specific nodes can communicate or specific VPCs and/or AZs to which compute nodes belong. As another example, the structured source data may identify databases and tables within databases and/or other entities in a database schema as well as relationships among the entities, such as foreign and/or primary keys. As another example, the structured source data may identify employees, executives, owners and/or other personnel of a company as well as their relationships to each other, such as relative ranks and organization into one or more divisions, departments, or other logical units.

Each graphical diagram may graphically represent one or more of the nodes using one or more graphical objects. In particular, each node may be represented by or otherwise associated with a set of one or multiple graphical objects. For example, compute nodes and database nodes of AWS structured source data may each be represented by a corresponding

graphical node in a graphical diagram. As another example, a node of an employee or other individual in an organization may be graphically represented by a box or other shape outline (e.g., a first graphical object), a photo of the employee (e.g., a second graphical object), text (e.g., a third graphical object) that may include the employee's name, office location, and/or other information, and/or a connector (e.g., a fourth graphical object) that indicates the employee's supervisor or manager or other individual the employee reports to.

Graphical objects that represent nodes in structured source data may be referred to as data-backed graphical objects. Data-backed graphical objects may be explicitly stored as and/or represented by data collections in structured source data. Each data-backed graphical object may have a corresponding data collection or attribute schema in a set of structured source data that explicitly identifies attributes of the graphical object, such as graphical attributes (e.g., fill color, outline color, shape) as well as other attributes such as resource type or other attribute(s).

The server 102 and/or the client devices 104, 106, 108 may be configured to automatically generate graphical diagrams based on structured source data. The graphical diagrams may be generated based on contract maps. Alternatively or additionally, automatically-generated graphical diagrams or graphical objects therein may be customized by application of conditional formatting. In some embodiments, updates made to the graphical drawings, e.g., to graphical objects within the graphical drawings that represent the nodes, may be propagated back to the structured source data in some circumstances. Alternatively or additionally, the server 102 and/or the client devices 104, 106, 108 may be configured to apply one or more reversible data transforms to the structured source data when generating graphical diagrams. Additional details regarding contract maps, customization of graphical diagrams, conditional formatting, updating structured source data, and reversible data transforms are disclosed in the following U.S. patent documents which are incorporated herein by reference in their entireties: U.S. Patent Pub. No. 20200004865 (U.S. App. No. 16/024,533, filed June 29, 2018), U.S. Patent Pub. No. 20200004866 (U.S. App. No. 16/024,544, filed June 29, 2018), U.S. Patent Pub. No. 20200004872 (U.S. App. No. 16/024,551, filed June 29, 2018), and U.S. App. No. 16/796,757, filed February 20, 2020.

Although one server 102 and three client devices 104, 106, 108 are illustrated in FIG. 1, the operating environment 100 may more generally include one or more servers 102 and one or more client devices 104, 106, 108. In these and other embodiments, the

operating environment 100 may include other servers and/or devices not illustrated in FIG. 1.

The operating environment 100 may additionally include a network 112. In general, the network 112 may include one or more wide area networks (WANs) and/or local area networks (LANs) that enable the server 102, the client devices 104, 106, 108, and the data sources 110 to communicate with each other. In some embodiments, the network 112 may include the Internet, including a global internetwork formed by logical and physical connections between multiple WANs and/or LANs. Alternately or additionally, the network 112 may include one or more cellular radio frequency (RF) networks and/or one or more wired and/or wireless networks such as 802.xx networks, Bluetooth access points, wireless access points, Internet Protocol (IP)-based networks, or other wired and/or wireless networks. The network 112 may also include servers that enable one type of network to interface with another type of network.

In general, the server 102 may host a web-based diagram application (hereinafter application) 114 that allows the client devices 104, 106, 108 to generate and display graphical diagrams. In other embodiments, the application 114 may include a non-web-based application but may generally be described herein as a web-based application for simplicity. Alternatively or additionally, some or all of the functionality described as being performed by the diagram application 114 may be performed locally on the client devices 104, 106, 108, such as by a browser or other application executed by the client devices 104, 106, 108.

The server 102 may additionally include a processor 116 and a storage medium 118. The processor 116 may be of any type such as a central processing unit (CPU), a microprocessor (μ P), a microcontroller (μ C), a digital signal processor (DSP), or any combination thereof. The processor 116 may be configured to execute computer instructions that, when executed, cause the processor 116 to perform or control performance of one or more of the operations described herein with respect to the server 102.

The storage medium 118 may include any non-transitory computer-readable medium, including volatile memory such as random access memory (RAM), persistent or non-volatile storage such as read only memory (ROM), electrically erasable and programmable ROM (EEPROM), compact disc-ROM (CD-ROM) or other optical disk storage, magnetic disk storage or other magnetic storage device, NAND flash memory or other solid state storage device, or other persistent or non-volatile computer storage

medium. The storage medium 118 may store computer instructions that may be executed by the processor 116 to perform or control performance of one or more of the operations described herein with respect to the server 102.

5 The storage medium 118 may additionally store one or more sets of structured source data 120, one or more models 122, model rules and/or heuristics 124 (“Model Rules/Heuristics” in FIG. 1), and/or one or more graphical diagrams 126. Each set of structured source data 120 may be imported from a corresponding one of the data sources 110. The structured source data 120 may be imported over the network 112 or directly from the corresponding data source 110 if a direct connection exists.

10 The model rules and/or heuristics 124 include rules and/or heuristics that may be applied to the structured source data 120 to derive information about the nodes and/or their relationships represented in the structured data. The structured source data 120 and/or the information derived by the model rules and/or heuristics 124 may be used by the server 102 to generate the models 122, each model 122 corresponding to a different set of structured
15 source data 120. Each of the graphical diagrams 126 may be generated directly from a corresponding set of the structured source data 120 and/or from a corresponding one of the models 122 derived from the corresponding set of structured source data 120. The graphical diagrams 126 may be generated based on input received from users at the client devices 104, 106, 108 and may include some or all of the nodes or other entities represented in the
20 corresponding set of structured source data 120 and/or in the corresponding model 122 derived from the corresponding set of structured source data 120.

Each of the client devices 104, 106, 108 may execute an application, such as the browser 128, configured to communicate through the network 112 with the server 102. The browser 128 may include an Internet browser or other suitable application for
25 communicating through the network 112 with the server 102. The browser 128 may generate, download and/or interact with structured source data 120 and/or graphical diagrams 126. Each of the client devices 104, 106, 108 may include a desktop computer, a laptop computer, a tablet computer, a mobile phone, a smartphone, a personal digital assistant (PDA), a wearable device (e.g., a smart watch), or another suitable client device.

30 Each of the client devices 104, 106, 108 may additionally include a processor and a storage medium, such as a processor 130 and a storage medium 132 as illustrated for the client device 104 in FIG. 1. Each of the other client devices 106, 108 may be similarly configured. Similar to the processor 116 of the server 102, the processor 130 may be of any type such as a CPU, a μ P, a μ C, a DSP, or any combination thereof. The processor 130 may

be configured to execute computer instructions that, when executed, cause the processor 130 to perform or control performance of one or more of the operations described herein with respect to the client device 104 and/or the browser 128.

Similar to the storage medium 118 of the server 102, the storage medium 132 of the client device 104 may include any non-transitory computer-readable medium, including volatile memory such as RAM, persistent or non-volatile storage such as ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage device, NAND flash memory or other solid state storage device, or other persistent or non-volatile computer storage medium. The storage medium 132 may store computer instructions that may be executed by the processor 130 to perform one or more of the operations described herein with respect to the client device 104 and/or the browser 128. The storage medium 132 may additionally store, at least temporarily, a graphical diagram 126, some of a corresponding set of structured source data 120 or model 122, and/or other content obtained from the server 102 and/or generated locally on the client device 104.

In some embodiments, the browser 128 on the client devices 104, 106, 108, the diagram application 114 on the server 102, and/or other application, system, or device may execute a layout algorithm to generate and display graphical diagrams 126 on displays of or coupled to the client devices 104, 106, 108. The resources, nodes, or other entities of the structured source data 120 may be arranged hierarchically. In these and other embodiments, the layout algorithm may recursively call down to the bottom of the hierarchy how much space is needed for graphical objects representing resources or nodes at each level of the hierarchy. At each level of the hierarchy, the algorithm may determine how much space a graphical object such as a container graphical object needs based on the space needed by each graphical object at the next lowest level of the hierarchy that is included in the container graphical object and an amount of padding (e.g., around the constituent graphical objects or the container graphical object itself) which may be a fixed parameter, a configurable parameter, and/or may be determined in some other manner. In effect, layout information at lower levels of the hierarchy bubbles up to the higher levels of the hierarchy allowing the layout algorithm to determine which containers to generate, the size of each container, where to position them, a number and size and location of rows and/or columns within each container, a size and location of each graphical object, and so on.

Embodiments described herein are not limited to using a browser to communicate with the server 102 to generate and display graphical diagrams 126 and group and collapse graphical objects therein. For example, rather than or in addition to a browser, the client

devices 104, 106, 108 may include a native app as are often used on client devices such as mobile devices including smartphones and tablet computers. Accordingly, embodiments described herein generally include generating and displaying graphical diagrams and grouping and collapsing graphical objects therein using a browser, a native app, or another
5 suitable application on the client devices 104, 106, 108.

FIGS. 2A-2D include graphical representations 200A, 200B, 200C, 200D (collectively “graphical representations 200”) of a UI to generate graphical diagrams, arranged in accordance with at least one embodiment described herein. The UI may be provided by the browser 128 and/or the diagram application 114 of FIG. 1. For instance,
10 the browser 128 in cooperation with the diagram application 114 may present the UI to a user through a display of the client device 104.

In FIGS. 2A-2D, the graphical representations 200 include a graphical drawing canvas 202, a panel 204 to one side of the graphical drawing canvas 202, one or more menus, and/or other UI elements. In other embodiments, the panel 204 may be positioned
15 in some other location relative to the graphical drawing canvas 202 and/or may be expanded into two or more panels. For example, the panel 204 may be positioned on a left side, above, or below the graphical drawing canvas 202 or other location relative to the graphical drawing canvas 202 and/or may be divided into two or more panels positioned in two or more locations relative to the graphical drawing canvas 202.

In general, graphical drawings or diagrams of structured source data may be displayed in the graphical drawing canvas 202. In FIGS. 2A-2D, the graphical drawing canvas 202 displays an example graphical diagram 206 that includes various graphical objects that represent resources or nodes in underlying structured source data. The graphical objects in this example include various AWS icons to represent AWS nodes, resources,
20 entities, or the like, including a region graphical object 208 that represents a region, virtual private cloud (VPC) graphical objects 210A, 210B (hereinafter collectively “VPC graphical objects 210”) that represent VPCs, compute node graphical objects 212A, 212B, 212C, 212D, 212E (hereinafter collectively “compute node graphical objects 212”) that represent compute resources, auto-scaling graphical objects 214A, 214B (hereinafter collectively
25 “auto-scaling graphical objects 214”) that represent auto-scaling resources, SQS queue graphical objects 216A, 216B, 216C (hereinafter collectively “SQSQ graphical objects 216”) that represent SQS queue resources, and an application programming interface (API) gateway graphical object 218 that represents an API gateway resource.

The panel 204 includes an input field 220, such as a dropdown list from which a particular view of structured source data may be selected. In the example of FIGS. 2A-2D, the selected view is “Infrastructure”. The panel 204 additionally includes an Explore tab, a Layout tab, and a Data tab.

5 Although not shown in FIGS. 2A-2D, the Explore tab may include a hierarchy selection area in which a user can turn on or off various hierarchy levels, or container levels, in the graphical diagram 206 shown in the graphical drawing canvas 202. The hierarchy selection area may also permit a user to specify which types of nodes the user desires to see at each container level, e.g., by selecting a dropdown arrow to display a list of all nodes at
10 that container level which the user can select or deselect.

The Layout tab has been selected in FIGS. 2A-2D and includes an example UI to group graphical objects. In particular, the Layout tab includes buttons 222, 224 that may be toggled between off and on to turn various types of grouping off or on.

In FIG. 2A, both buttons 222, 224 are toggled to off.

15 In FIG. 2B, the button 222 is toggled to on while the button 224 is toggled to off. Toggling the button 222 to on may cause graphical objects to be grouped by resource type, which may be one of the attributes of each graphical object in their underlying schema. In this example, each graphical object that has the same value for the resource type attribute may be grouped together. For example, as illustrated in FIG. 2B, each of the VPC graphical
20 objects 210 may have a value of VPC for their resource type attribute and may be grouped together, each of auto-scaling graphical objects 214 may have a value of auto-scaling for their resource type attribute and may be grouped together as shown in FIG. 2B, each of the SQSQ graphical objects 216 may have a value of SQS queue for their resource type attribute and may be grouped together, and the API gateway graphical object 218 may have
25 a value of API gateway for its resource type attribute and may be grouped by itself.

Graphical objects may be grouped in this and other embodiments by arranging them next to or near each other, within a container that may or may not be visible, and/or within duplicated versions of a container. In the example of FIGS. 2B-2D, graphical objects that are grouped in response to input, e.g., from a user, are arranged next to each other and
30 within containers that are invisible. For example, in FIG. 2B, the auto-scaling graphical objects 214 are positioned within an invisible container that may have a boundary indicated by a dashed line 226 in FIG. 2B for illustrative purposes. The VPC graphical objects 210 are similarly positioned near each other within another invisible container. The SQSQ graphical objects 216 are similarly positioned near each other within another invisible

container. The API gateway graphical object 218 is similarly positioned within another invisible container. The use of visible or invisible containers for grouping graphical objects may facilitate use of a layout algorithm such as described elsewhere herein to arrange grouped graphical objects within the graphical drawing canvas 202.

5 In some embodiments, grouping may be applied at a single hierarchical level and/or within a single selected graphical object. For example, in FIGS. 2A-2D, the region graphical object 208 may be selected to apply grouping to graphical objects within the region graphical object 208. Alternatively or additionally, grouping may be applied only at a hierarchical level immediately below a hierarchical level of the selected graphical object.

10 For example, in FIGS. 2A-2D, and assuming the region graphical object 208 has been selected, grouping may be applied to VPC graphical objects 210, auto-scaling graphical objects 214, SQSQ graphical objects 216, and API gateway graphical object 218 that represent resources or nodes at a child hierarchical level immediately beneath a parent hierarchical level of a resource or node represented by the region graphical object 208. In

15 some embodiments, the grouping is not applied to the compute node graphical objects 212 because they represent resources or nodes in a next level down from the child hierarchical level, e.g., in a grandchild hierarchical level.

 In FIG. 2C, the button 222 is toggled to off while the button 224 is toggled to on. Toggling the button 224 to on may cause graphical objects that are not container graphical

20 objects to be grouped together. A container graphical object may be any graphical object that may enclose at least one other graphical object therein. Alternatively or additionally, a container graphical object may be any graphical object that represents a resource or node with at least one child resource or node in structured source data. Graphical objects that are not container graphical objects may include any graphical object that does not enclose at

25 least one other graphical object therein and/or that represents a resource or node with no child resources or nodes in the structured source data. As illustrated in FIG. 2C, the auto-scaling graphical objects 214, the SQSQ graphical objects 216, and the API gateway graphical object 218 are not container graphical objects and are grouped together. In FIG.

30 2C, the auto-scaling graphical objects 214, the SQSQ graphical objects 216, and the API gateway graphical object 218 are grouped next to and/or near each other and/or may be grouped within a visible or invisible container. In addition, the grouping of graphical objects that are not container graphical objects may be applied only to graphical objects within a selected container graphical object (e.g., the region graphical object 208 in this example) and/or at a single hierarchical level within the selected container graphical object

(e.g., at the child hierarchical level relative to the parent hierarchical level of the region graphical object 208). Alternatively, the grouping of graphical objects may be applied in other manners, e.g., to all or a subset of graphical objects within the graphical drawing canvas 202 and/or to all or a subset of hierarchical levels.

5 In FIG. 2D, both buttons 222, 224 are toggled to on. As a result, graphical objects are grouped based on both the value of their resource type attribute and their status as a container or non-container graphical object. For example, the VPC graphical objects 210 are grouped together as container graphical objects each having a resource type of VPC, the auto-scaling graphical objects 214 are grouped together as non-container graphical
10 objects each having a resource type of auto-scaling, the SQSQ graphical objects 216 are grouped together as non-container graphical objects each having a resource type of SQS queue, and the API gateway graphical object 218 is grouped by itself as a non-container graphical object having a resource type of API gateway. As in other embodiments, the grouping of graphical objects may be applied only to graphical objects within a selected
15 container graphical object (e.g., the region graphical object 208 in this example) and/or at a single hierarchical level within the selected container graphical object. Alternatively, the grouping of graphical objects may be applied in other manners, e.g., to all or a subset of graphical objects within the graphical drawing canvas 202 and/or to all or a subset of hierarchical levels.

20 FIGS. 3A-3J include graphical representations 300A, 300B, 300C, 300D, 300E, 300F, 300G, 300H, 300I, 300J (collectively “graphical representations 300”) of other UIs to generate graphical diagrams, arranged in accordance with at least one embodiment described herein. The UIs may be provided by the browser 128 and/or the diagram application 114 of FIG. 1. For instance, the browser 128 in cooperation with the diagram
25 application 114 may present the UIs to a user through a display of the client device 104.

In FIGS. 3A-3J, the graphical representations 300 include a graphical drawing canvas 302. Although not shown in all of FIGS. 3A-3J, the graphical representations 300 may further include one or more panels, such as a panel 301 illustrated in FIG. 3A, one or more menus, and/or other UI elements, some examples of which are described with respect
30 to FIGS. 2A-2D. As in FIGS. 2A-2D, in general, graphical drawings or diagrams of structured source data may be displayed in the graphical drawing canvas 302. In FIGS. 3A-3J, the graphical drawing canvas 302 displays an example graphical diagram 304 that includes various graphical objects that represent resources, nodes, entities, or the like in underlying structured source data. The graphical objects in this example include various

AWS icons to represent AWS nodes, resources, entities, or the like, including a download distribution graphical object 306 that represents a download distribution resource, API gateway graphical objects 308A, 308B, 308C (hereinafter collectively API gateway graphical objects 308”) that represent API gateway resources, an SQSQ graphical object 5 310 that represents an SQS queue resource, a bucket graphical object 312 that represents an S3 Bucket resource, application load balancer (ALB) graphical objects 314A, 314B (hereinafter collectively “ALB graphical objects 314”) that represents ALB resources, Amazon relational database service (RDS) graphical objects 316A, 316B (hereinafter collectively “Amazon RDS graphical objects 316”), compute node graphical objects 318A, 10 318B, 318C, 318D (hereinafter collectively “compute node graphical objects 318”) that represent compute resources, Elastic Network Interface (ENI) graphical objects 320A, 320B (hereinafter collectively “ENI graphical objects 320”) that represent ENI resources, lambda function graphical objects 322A, 322B (hereinafter collectively “lambda function graphical objects 322”) that represent lambda function resources, an account graphical 15 object 324 that represents an account, a region graphical object 326 that represents a region, a VPC graphical object 328 that represents a VPC, and a private subnet graphical object 330 that represents a private subnet. Each of the account graphical object 324, the region graphical object 326, the VPC graphical object 328, and the private subnet graphical object 330 is a container graphical object and may be referred to hereinafter as a corresponding 20 container, e.g., the account container 324.

In an example implementation, a user may provide input to define one or more groups based on an attribute and/or based on an arbitrary selection. The resulting groups may be referred to as user-defined groups. When based on an attribute, graphical objects that have the same value for a given attribute may be grouped together. When based on 25 arbitrary selection, graphical objects that are selected, e.g., by the user, may be grouped together. A UI to define one or more groups may be implemented in any suitable manner. FIGS. 2A-2D illustrate one such UI in the form of the panel 204 in which groups may be defined by turning on buttons 222, 224.

The same, similar, or different UI as the panel 204 may be implemented to define 30 groups in FIGS. 3B-3J. For example, FIG. 3A includes the panel 301 with UI elements to define one or more groups. The panel 301 may include a list 303, such as a scrolling list, of various attributes that may be selected to define one or more groups. The list 303 in this example includes an “Environment tag” attribute, an “Application tag” attribute, and a “Resource type” attribute. More generally, such a list may include one or more selectable

attributes. In some embodiments, selection of the desired attribute from the list 303 may define the groups and result in generation of one or more containers for one or more resulting user-defined groups and relative arrangement of the generated containers and the graphical objects to depict membership of the graphical objects in the groups. Alternatively or additionally, after selecting the desired attribute from the list 303, the user may select a “group” button 305A, 305B (hereinafter generically “group button 305”) to define the groups and cause generation of the containers and relative arrangement of the generated containers and graphical objects to depict membership. Selection of the group button 305A may group graphical objects in the graphical diagram 304 that represent all resources in underlying source data according to their value for the selected attribute. The group button 305A may be displayed in the panel 301 when no graphical object is selected in the graphical diagram 304. Selection of the group button 305B may group only those graphical objects in the graphical diagram 304 that represent resources of a given resource type in underlying source data according to their respective value for the selected attribute. The group button 303 may be displayed in the panel 301 when a graphical object representing a resource with the given resource type has been selected to apply grouping as described herein only to graphical objects that represent resources of the given resource type, e.g., graphical objects that have the same value for their resource type attribute as the selected graphical object.

Alternatively or additionally, the panel 301 may include layout elements 307 to select a layout of the graphical diagram 304. The layout elements 307 may be operable with or without application of grouping based on user-defined groups as described herein. In the illustrated example, the layout elements 307 include selectable layout options, including a selectable “Cross-layout” option and a selectable “Hierarchical” layout option. An example of the graphical diagram 304 with the cross-layout option applied is depicted in FIG. 3B. Examples of the graphical diagram 304 with various hierarchical layout options applied are depicted in FIGS. 3C-3E. In some embodiments, the hierarchical layout option may include a drop-down menu or other UI element to select or identify a location at which to position containers generated for user-defined groups.

In the examples of FIGS. 3B-3E, user input is provided through the panel 301 or other suitable UI to define groups based on a selected attribute, which in this example is the environment tag attribute in which various graphical objects have a value of “production” (“prod” in the FIGS.) or “preproduction” (“preprod” in the FIGS.) for the environment tag attribute. The groups thus defined include a first group of graphical objects

306, 308A, 308C, 310, 312, 314B, 316B, 318C, 318D, 320B, 322B with a value of “production” for the environment tag attribute and a second group of graphical objects 308B, 314A, 316A, 318A, 318B, 320A, 322A with a value of “preproduction” for the environment tag attribute. More generally, the defined groups may include one group for each unique value of the selected attribute. In response to the input, one or more containers are generated, including a first container 332 for the first group of “production” graphical objects and a second container 334 for the second group of “preproduction” graphical objects in this example. In some embodiments, each of the first and second containers 332, 334 may include a label to identify the group represented by the container. For example, the first container 332 may have a “prod” label to indicate that it represents the “production” group while the second container 334 may have a “preprod” label to indicate that it represents the “preprod” group. Membership of the graphical objects 306, 308, 310, 312, 314, 316, 318, 320, 322 in the groups may be graphically depicted by relative arrangement of the graphical objects 306, 308, 310, 312, 314, 316, 318, 320, 322 and the one or more containers, e.g., first and second containers 332, 334 in this example, on the graphical drawing canvas 302 according to group membership of each of the graphical objects 306, 308, 310, 312, 314, 316, 318, 320, 322. Numerous arrangements to show membership are possible including some specific examples depicted in FIGS. 3B-3E.

In these and other examples, the graphical objects may represent data objects of a data source and the data objects may have a hierarchy. The hierarchy may be graphically represented in the illustrated embodiment by arrangement of the graphical objects within nested containers 324, 326, 328, 330. For example, the private subnet container 330 is positioned within the VPC container 328, which is positioned within the region container 326, which is positioned within the account container 324, thus indicating the level in the hierarchy of each of the underlying account, region, VPC, and private subnet corresponding to these containers. Further, the various graphical objects are positioned within a corresponding one of the containers 324, 326, 328, 330 according to the level of the hierarchy at which the underlying data objects are located.

One layout and/or arrangement to show membership is illustrated in FIG 3B as a cross-layout arrangement of the generated containers 332, 334 with the nested container 324, 326, 328, 330. In this example, the first container 332 for the “production” group of graphical objects and the second container 334 for the “preproduction” group of graphical objects are arranged side-by-side and extending across the nested containers 324, 326, 328, 330, with any graphical objects belonging to each first or second container 332, 334 and a

corresponding one of the nested containers 324, 326, 328, 330 being positioned at a corresponding intersection of the first or second container 332, 334 with the corresponding one of the nested containers 324, 326, 328, 330. For example, the download distribution graphical object 306 is a member of the “production” group (e.g., it has “production” as the value of its environment tag attribute) and belongs to the account container 324 but not any of the other nested containers 326, 328, 330 so the download distribution graphical object 306 is displayed at the intersection of the first container 332 with the account container 324 in an area where the other nested containers 326, 328, 330 are absent. As another example, the compute node graphical object 318A is a member of the “preproduction” group (e.g., it has “preproduction” as the value of its environment tag attribute) and belongs to the private subnet container 330 nested within the other containers 328, 326, 324 so the compute node graphical object 318A is displayed at the intersection of the second container 334 with the private subnet container 330. Example embodiments of how to generate graphical diagrams with such cross-layout arrangements are disclosed in U.S. Patent App. No. 17/176,975 filed February 16, 2021 and titled DIAGRAMMING CHILD NODES WITH MULTIPLE PARENT NODES, which is incorporated herein by reference in its entirety.

The cross-layout arrangement of FIG. 3B may preserve hierarchical information represented by the graphical diagram 304 since the cross-layout arrangement of FIG. 3B depicts members of each of the graphical objects 306, 308, 310, 312, 314, 316, 318, 320, 322 in both a correct location of the nested containers 324, 326, 328, 330 (representing the correct hierarchical level of the underlying data objects) and the corresponding group container 332, 334. FIGS. 3C-3E depict other arrangements that may preserve hierarchical information while showing membership in user-defined groups such as the “production” group and the “preproduction” group in these examples.

Referring to FIG. 3C, the first and second containers 332, 334 are placed at a highest level of the hierarchy represented in FIG. 3C, e.g., above the account container 324. To preserve the hierarchical information, the nested containers 324, 326, 328, 330 may be duplicated. A different duplicate of the nested containers 324, 326, 328, 330 is positioned within each of the first and second containers 332, 334. Each of the graphical objects 306, 308, 310, 312, 314, 316, 318, 320, 322 is then positioned at the appropriate hierarchical level within one of the duplicates in a corresponding one of the first and second containers 332, 334. For example, each of the graphical objects 306, 308A, 308C, 310, 312, 314B, 316B, 318C, 318D, 320B, 322B in the “production” group is positioned at the appropriate hierarchical level within the duplicate of the nested containers 324, 326, 328, 330 on the

left in FIG. 3C, e.g., the duplicate within the first container 332. Similarly, each of the graphical objects 308B, 314A, 316A, 318A, 318B, 320A, 322A in the “preproduction” group is positioned at the appropriate hierarchical level within the duplicate of the nested containers 324, 326, 328, 330 on the right in FIG. 3C, e.g., the duplicate within the second container 334.

Referring to FIG. 3D, the first and second containers 332, 334 are placed at a lower level of the hierarchy than in FIG. 3C, e.g., above the region container 326 but below the account container 324. Since the account container 324 is above the hierarchical level at which the first and second containers 332, 334 are placed, the account container 324 is not duplicated. However, to preserve the hierarchical information, the nested containers 326, 328, 330 below the hierarchical level of the first and second containers 332, 334 may be duplicated. A different duplicate of the nested containers 326, 328, 330 is positioned within each of the first and second containers 332, 334. Similar to FIG. 3C, each of the graphical objects 306, 308, 310, 312, 314, 316, 318, 320, 322 is then positioned at the appropriate hierarchical level within one of the duplicates in a corresponding one of the first and second containers 332, 334. For example, each of the graphical objects 306, 308A, 308C, 310, 312, 314B, 316B, 318C, 318D, 320B, 322B in the “production” group is positioned at the appropriate hierarchical level within the duplicate of the nested containers 324, 326, 328, 330 on the left in FIG. 3D, e.g., the duplicate within the first container 332. Similarly, each of the graphical objects 308B, 314A, 316A, 318A, 318B, 320A, 322A in the “preproduction” group is positioned at the appropriate hierarchical level within the duplicate of the nested containers 324, 326, 328, 330 on the right in FIG. 3D, e.g., the duplicate within the second container 334.

Referring to FIG. 3E, the first and second containers 332, 334 are placed at each level of the hierarchy around any graphical objects at each level of the hierarchy that belong to the corresponding group. In particular, the first and second containers 332, 334 may be duplicated and distributed across the nested containers 324, 326, 328, 330 such that each set of one or more graphical objects that belong to a corresponding one of the groups at each hierarchical level immediately below a corresponding one of the nested containers 324, 326, 328, 330 is positioned within a corresponding one of the duplicates. For example, the first container 332 is duplicated and then distributed throughout the nested containers 324, 326, 328, 330 including one duplicate of the first container 332 beneath the hierarchical level represented by the account container 324 being positioned around the download distribution graphical object 306, another duplicate of the first container 332

beneath the hierarchical level represented by the region container 326 being positioned around the graphical objects 308A, 308C, 310, 312, another duplicate of the first container 332 beneath the hierarchical level represented by the VPC container 328 being positioned around the graphical objects 314B, 316B, and another duplicate of the first container 332
5 beneath the hierarchical level represented by the private subnet container 330 being positioned around the graphical objects 318C, 318D, 320B, 322B. As another example, the second container 334 is duplicated and then distributed throughout the nested containers 324, 326, 328, 330 including one duplicate of the second container 334 beneath the hierarchical level represented by the region container 326 being positioned around the API
10 graphical object 308B, another duplicate of the second container 334 beneath the hierarchical level represented by the VPC container 328 being positioned around the graphical objects 314A, 316A, and another duplicate of the second container 334 beneath the hierarchical level represented by the private subnet container 330 being positioned around the graphical objects 318A, 318B, 320A, 322A. Optionally, duplicates of the
15 containers generated for the user-defined groups may be provided beneath hierarchical levels at which there are no members of the group. For example, although the “preproduction” group does not include any graphical objects in the hierarchical level immediately beneath the account container 324, the graphical diagram 304 may nevertheless include an empty duplicate of the second container 334 beneath the
20 hierarchical level represented by the account container 324, as illustrated in FIG. 3E.

FIG. 3C shows containers generated for user-defined groups (e.g., the first and second containers 332, 334) placed at a highest level of the hierarchy, FIG. 3D shows containers generated for user-defined groups placed at a next level down compared to FIG. 3C, and FIG. 3E shows containers generated for user-defined groups placed at all levels of
25 the hierarchy. More generally, the containers generated for user-defined groups may be placed at any level in the hierarchy or at multiple levels of the hierarchy. The particular placement level may determine whether some or all of the nested containers are duplicated as in FIGS. 3C and 3D or whether one or more of the containers generated for the user-defined groups are duplicated. Some implementations may involve both duplication of
30 some nested containers and duplication of the containers generated for the user-defined groups. For example, suppose the first and second containers 332, 334 are placed at the level in the hierarchy immediately beneath the region container 326. The VPC and private subnet containers 328, 330 would be duplicated with one of the duplicates positioned within the first container 332 inside the region container 326 and the other duplicate positioned

within the second container 334 also inside the region container 326. In addition, at least the first container 332 would be duplicated so that one of the duplicates could be placed around the download distribution graphical object 306 outside the region container 326 but inside the account container 324.

5 When the containers generated for user-defined groups are duplicated and one of the duplicates of a given container is selected, some embodiments may alter the graphical diagram 304 to indicate that all other duplicates of the given container are duplicates of the same container as the selected duplicate. Altering the graphical diagram 304 may include highlighting all other duplicates of the first or second container or displaying connectors
10 between the selected duplicate of the first or second container and all other duplicates of the first or second container.

 For example, FIG. 3F illustrates the same layout and/or arrangement as in FIG. 3E. In FIG. 3F, one duplicate of the first container 332 has been selected. In particular, the duplicate of the first container 332 immediately beneath the level of the hierarchy
15 represented by the region container 326 has been selected. The selection of the given duplicate of the first container 332 (and/or of other graphical objects) may be indicated by adding one or more graphical object controls to, around, and/or near the selected duplicate, by highlighting the selected duplicate, or more generally by altering the display of the selected duplicate in some manner. Other duplicates of the first container 332 may be
20 highlighted or more generally altered in some manner to indicate that the other duplicates are duplicates of the same first container 332 as the selected duplicate. By way of example, in FIG. 3F, the selected duplicate of the first container 332 is indicated by added graphical object controls to the middle of each side, top, and bottom of the selected duplicate of the first container 332 and by increased or otherwise altered brightness, line thickness, color,
25 and/or other characteristic of the selected duplicate. In addition, all other duplicates of the first container 332 are highlighted to indicate that they are duplicates of the same container as the selected duplicate.

 Another example UI to define one or more groups of one or more graphical objects in the graphical diagram 304 is illustrated in FIG. 3G. In this example, the user may position
30 a mouse cursor 336 (or other input system or device) to simultaneously select multiple graphical objects, including the graphical objects 306, 310, 314A, 316B, 318A, and 320A in this example. To say that multiple graphical objects are simultaneously selected does not necessarily mean that an act of selection by a user of all of the graphical objects occurs at the same time. Rather, it means that multiple graphical objects have been selected

(sequentially or simultaneously) and all remain selected at a given point in time that may be referred to as the time at which they are simultaneously selected. Multiple graphical objects may be simultaneously selected by a single act of selection (e.g., by dragging a single mouse window around all the desired graphical objects at once) or by multiple non-
5 simultaneous acts of selection (e.g., by sequentially selecting the graphical objects with the mouse cursor 336 while the control key of a keyboard is depressed at least during each selection by the mouse cursor 336) or a combination of the two. At the time at which the desired graphical objects are simultaneously selected and the mouse cursor 336 is positioned over one of the selected graphical objects, the user may provide a suitable input,
10 such as a right mouse input, to open a menu that may include an option to group the selected graphical objects. The option to group the selected graphical objects may be selected from the menu and the graphical objects may then be grouped together in an arbitrary user-defined group. In FIG. 3G, for instance, the mouse cursor 320 may be positioned on the ENI graphical object 320A after all of the graphical objects 306, 310, 314A, 316B, 318A,
15 320A are simultaneously selected and a menu 338 may be opened responsive to right mouse input. The menu 338 includes various options, including a “Group Selected” option that may be selected to define an arbitrary user-defined group that includes all of the selected graphical objects 306, 310, 314A, 316B, 318A, 320A.

In response to selection of the “Group Selected” option or other suitable input, a
20 container 340 may be generated for the arbitrary user-defined group. In addition, membership of the selected graphical objects 306, 310, 314A, 316B, 318A, 320A in the arbitrary user-defined group may be graphically depicted by relative arrangement of the selected graphical objects 306, 310, 314A, 316B, 318A, 320A and the generated container 340. The generated container may include a label that may be a default label and/or a user-
25 defined label. The label is “Application 1” in this example.

Various example arrangements of the graphical diagram 304 after selection of the “Group Selected” option in the menu 338 are illustrated in FIGS. 3H-3J. As illustrated in FIGS. 3H-3J, the graphical diagram 304 may be arranged in one or more of the same or similar layouts as in FIGS. 3B-3E. For example, FIG. 3H illustrates the graphical diagram
30 304 with a cross-layout arrangement, FIG. 3I illustrates the graphical diagram 304 with a hierarchical arrangement in which the generated container 340 is at a highest level of the hierarchy with duplicated nested containers 324, 326, 328, 330, and FIG. 3J illustrates the graphical diagram 304 with another hierarchical arrangement in which the generated container 340 is duplicated and distributed throughout the nested containers 324, 326, 328,

330. In the example of FIGS. 3G-3J, a panel (such as the panel 301 of FIG. 3A) or other UI with one or more layout elements, such as the layout elements 307, may be provided for the user to select and/or otherwise configure a desired layout, such as one or more of the layouts depicted in FIGS. 3H-3J.

5 In some embodiments, existing user-defined groups such as those generated as described with respect to FIGS. 3A-3J may be ungrouped. For instance, the user may select an “ungroup button” displayed in a UI panel (such as the panel 301) that may be displayed in the UI panel after the group button 305 has been selected in the generation of the user-defined group(s) to turn off the grouping and ungroup the group graphical objects.

10 Alternatively or additionally, for an arbitrary user-defined group, the user may position the mouse cursor 336 over any graphical object of the arbitrary user-defined group and provide a right mouse input or other suitable input with the mouse cursor 336 over the graphical object to open a menu, similar to the menu 338, that may have an option to ungroup the graphical objects of the arbitrary user-defined group. Selection of the ungroup option may

15 ungroup the graphical objects. Alternatively or additionally, additional graphical objects may be added to existing arbitrary user-defined groups. For instance, the user may position the mouse cursor 336 over a graphical object the user desires to add to an existing arbitrary user-defined group and provide a right mouse input or other suitable input with the mouse cursor 336 over the graphical object to open a menu, similar to the menu 338, that may

20 have an option to add the graphical object to an existing arbitrary user-defined group. As another example, the user may position the mouse cursor 336 over any graphical object of an arbitrary user-defined group and provide a right mouse input or other suitable input with the mouse cursor 336 over the graphical object to open a menu, similar to the menu 338, that may have an option to edit the group, e.g., by adding additional graphical objects to

25 the group. Selection of the editing option may allow the user to select the desired additional graphical object to add to the group.

FIG. 4 illustrates a flowchart of an example method 400 to group graphical objects, arranged in accordance with at least one embodiment described herein. The method 400 may be performed by any suitable system, apparatus, or device. For example, any one or

30 more of the client devices 104, 106, 108, the collaboration application 114, and/or the server 102 of FIG. 1 may perform or direct performance of one or more of the operations associated with the method 400. In these and other embodiments, the method 400 may be performed or controlled by one or more processors based on one or more computer-readable instructions stored on one or more non-transitory computer-readable media.

Alternatively or additionally, embodiments herein may include a non-transitory computer-readable medium having computer-readable instructions stored thereon that are executable by a processor to perform or control performance of the method 400 or one or more operations thereof. The method 400 may include one or more of blocks 402, 404, 406,
5 and/or 408.

At block 402, the method 400 may include displaying, on a display device, a graphical diagram with multiple graphical objects that represent data of a data source. For example, block 402 may include displaying the graphical diagram 206, 304 with the corresponding graphical objects representing structured source data 120 on a display device
10 included in or coupled to the client device 104 of FIG. 1. Block 402 may be followed by block 404.

At block 404, the method 400 may include receiving input to define one or more groups. For example, block 404 may include the client device 104 of FIG. 1 receiving input to toggle any of the buttons 222, 224 to on. Alternatively or additionally, block 404 may
15 include receiving input to select a “Group Selected” option after manual selection of multiple graphical objects to include in the group, such as the “Group Selected” option from the menu 338 of FIG. 3G. Block 404 may be followed by block 406.

At block 406, the method 400 may include, in response to the input, generating one or more containers. For example, block 406 may include generating an invisible container
20 such as the invisible container 226 of FIG. 2B, or a visible container such as the containers 332, 334, 340 of FIGS. 3B-3F and 3H-3J. Block 406 may be followed by block 408.

At block 408, the method 400 may include, in response to the input, graphically depicting membership of the graphical objects in the one or more groups by relative arrangement of the graphical objects and the one or more containers according to group
25 membership of each of the graphical objects. As an example, block 408 may include grouping two or more graphical objects within an invisible container, such as grouping all the auto-scaling graphical objects 214 of FIG. 2B together in one invisible container, grouping all the SQSQ graphical objects 216 of FIG. 2B together in another invisible container, and grouping the API gateway graphical object 218 of FIG. 2B by itself in
30 another invisible container. As another example, block 408 may include grouping two or more graphical objects within a visible container, such as grouping the graphical objects 306, 308A, 310, 308C, 312, 314B, 316B, 318C, 318D, 320B, 322B in the first container 332 and grouping the graphical objects 308B, 314A, 316A, 318A, 318B, 320A, 322A in the second container 334 of FIGS. 3B-3F.

In some embodiments, the input received at block 404 includes first input to define an arbitrary group and the method 400 may further include, prior to receiving the first input, receiving second input to simultaneously select two or more of the graphical objects to include in the arbitrary group. For example, the second input may include input to simultaneously select the graphical objects 306, 310, 314A, 316B, 318A, 320A and the first input may include input to select the “Group Selected” button in the menu 338 as described with respect to FIGS. 3G-3J.

In some embodiments, the input received at block 404 includes first input to select a pre-existing container and second input to group graphical objects located within the pre-existing container into the one or more containers according to a respective value of an attribute of each of the graphical objects, e.g., as described with respect to FIGS. 2A-2D. In these and other embodiments, the grouping may apply only at one hierarchical level, e.g., below a hierarchical level of the pre-existing container, e.g., as described with respect to FIGS. 2A-2D. In more detail, the graphical objects may represent data objects of the data source, the data objects having a hierarchy. The one or more containers may include two or more containers generated within the pre-existing container for two or more different subsets of the graphical objects each grouped together in a corresponding one of the two or more containers. Two or more graphical objects grouped together in a first container of the two or more containers may each have a first value for the attribute. Two or more other graphical objects grouped together in a second container of the two or more containers may each have a second value for the attribute that is different than the first value. The grouping together of the graphical objects located within the pre-existing container into the two or more containers may apply exclusively to graphical objects that represent data objects at a first level of the hierarchy below a second level of the hierarchy represented by the pre-existing container.

In some embodiments, the input received at block 404 includes first input including a selection of an attribute to group graphical objects and the method 400 may further include identifying unique values of the attribute across the graphical objects. The one or more containers may include at least one container for each of the unique values of the attribute.

In some embodiments, the graphical objects represent data objects of the data source, the data objects having a hierarchy. The hierarchy may be graphically represented by arrangement of the graphical objects within multiple nested containers, e.g., as described with respect to FIGS. 3A-3J. The method 400 may further include preserving the hierarchy

while graphically depicting membership of the graphical objects in the one or more groups. In these and other embodiments, the method 400 may further include receiving second input to select: a level of the hierarchy at which to arrange the one or more containers; or a cross-layout arrangement in which to arrange the one or more containers with the nested
5 containers. Such second input may include, e.g., selection of one of the layout elements 307 of the panel 301 of FIG. 3A.

In some embodiments, the one or more containers include at least a first container and a second container. In these and other embodiments, preserving the hierarchy while graphically depicting membership of the graphical objects in the one or more groups may
10 include one of: arranging the first and second containers in a cross-layout arrangement with the nested containers (e.g., as described with respect to FIG. 3B); duplicating the nested containers and arranging the first and second containers at a highest level of the hierarchy (e.g., as described with respect to FIG. 3C); or arranging the first and second containers at a level of the hierarchy beneath the highest level of the hierarchy (e.g., as described with
15 respect to FIGS. 3D and 3E). Duplicating the nested containers and arranging the first and second containers at the highest level of the hierarchy may include positioning a first duplicate of the nested containers inside the first container; positioning graphical objects that belong to a first group represented by the first container within the first duplicate of the nested containers; positioning a second duplicate of the nested containers inside the
20 second container; and positioning graphical objects that belong to a second group represented by the second container within the second duplicate of the nested containers.

In some embodiments, the one or more containers include at least a first container and a second container. In these and other embodiments, preserving the hierarchy while graphically depicting membership of the graphical objects in the one or more groups may
25 include arranging the first container at each level of the hierarchy that includes at least one graphical object that belongs to the first group represented by the first container and the second container at each level of the hierarchy that includes at least one graphical object that belongs to the second group represented by the second container. The foregoing may include distributing duplicates of the first container across the nested containers such that
30 each set of one or more graphical objects that belong to the first group at each hierarchical level immediately beneath a corresponding one of the nested containers is positioned within one of the duplicates of the first container; and distributing duplicates of the second container across the nested containers such that each set of one or more graphical objects that belong to the second group at each hierarchical level immediately beneath a

corresponding one of the nested containers is positioned within one of the duplicates of the second container. The method 400 may further include receiving second input to select one of the duplicates of the first or second container; and in response to the second input, altering the graphical diagram to indicate that all other duplicates of the first or second container are duplicates of the same container as the selected duplicate (e.g., as described with respect to FIG. 3F). In some embodiments, the altering of the graphical diagram includes at least one of: highlighting all other duplicates of the first or second container; or displaying connectors between the selected duplicate of the first or second container and all other duplicates of the first or second container.

FIG. 5 is a block diagram illustrating an example computing device 500 that is arranged to generate and/or display graphical diagrams, arranged in accordance with at least one embodiment described herein. The computing device 500 may include, be included in, or otherwise correspond to either or both of the server 102 or the client devices 104, 106, 108 of FIG. 1. In a basic configuration 502, the computing device 500 typically includes one or more processors 504 and a system memory 506. A memory bus 508 may be used to communicate between the processor 504 and the system memory 506.

Depending on the desired configuration, the processor 504 may be of any type including, but not limited to, a microprocessor (μ P), a microcontroller (μ C), a digital signal processor (DSP), or any combination thereof. The processor 504 may include one or more levels of caching, such as a level one cache 510 and a level two cache 512, a processor core 514, and registers 516. The processor core 514 may include an arithmetic logic unit (ALU), a floating point unit (FPU), a digital signal processing core (DSP Core), or any combination thereof. An example memory controller 518 may also be used with the processor 504, or in some implementations the memory controller 518 may include an internal part of the processor 504.

Depending on the desired configuration, the system memory 506 may be of any type including volatile memory (such as RAM), nonvolatile memory (such as ROM, flash memory, etc.), or any combination thereof. The system memory 506 may include an operating system 520, one or more applications 522, and program data 524. The application 522 may include a diagram application 526 that is arranged to generate and/or display graphical diagrams. The diagram application 526 may include, be included in, or otherwise correspond to the diagram application 114 of FIG. 1. The program data 524 may include graphical diagrams 528 (which may include, be included in, or otherwise correspond to the graphical diagrams 126 of FIG. 1) as is described herein, structured source data collections

(such as structured source data 120 of FIG. 1), models of structured source data (such as models 122 of FIG. 1) and/or other graphical data. In some embodiments, the application 522 may be arranged to operate with the program data 524 on the operating system 520 such that one or more methods may be provided as described herein.

5 The computing device 500 may have additional features or functionality, and additional interfaces to facilitate communications between the basic configuration 502 and any involved devices and interfaces. For example, a bus/interface controller 530 may be used to facilitate communications between the basic configuration 502 and one or more data storage devices 532 via a storage interface bus 534. The data storage devices 532 may
10 be removable storage devices 536, non-removable storage devices 538, or a combination thereof. Examples of removable storage and non-removable storage devices include magnetic disk devices such as flexible disk drives and hard-disk drives (HDDs), optical disk drives such as compact disk (CD) drives or digital versatile disk (DVD) drives, solid state drives (SSDs), and tape drives to name a few. Example computer storage media may
15 include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer-readable instructions, data structures, program modules, or other data.

 The system memory 506, the removable storage devices 536, and the non-removable storage devices 538 are examples of computer storage media or non-transitory
20 computer-readable media. Computer storage media or non-transitory computer-readable media includes RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVDs) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other non-transitory medium which may be used to store the desired information and which may be accessed
25 by the computing device 500. Any such computer storage media or non-transitory computer-readable media may be part of the computing device 500.

 The computing device 500 may also include an interface bus 540 to facilitate communication from various interface devices (e.g., output devices 542, peripheral
30 interfaces 544, and communication devices 546) to the basic configuration 502 via the bus/interface controller 530. The output devices 542 include a graphics processing unit 548 and an audio processing unit 550, which may be configured to communicate to various external devices such as a display or speakers via one or more A/V ports 552. Diagrams, flowcharts, organizational charts, connectors, and/or other graphical objects generated by the diagram application 526 may be output through the graphics processing unit 548 to such

a display. The peripheral interfaces 544 include a serial interface controller 554 or a parallel interface controller 556, which may be configured to communicate with external devices such as input devices (e.g., keyboard, mouse, pen, voice input device, touch input device, etc.), sensors, or other peripheral devices (e.g., printer, scanner, etc.) via one or more I/O ports 558. Such input devices may be operated by a user to provide input to the diagram application 526, which input may be effective to, e.g., generate curved connectors, designate points as designated points of one or more curved connectors, relocate one or more designated points, and/or to accomplish other operations within the diagram application 526. The communication devices 546 include a network controller 560, which may be arranged to facilitate communications with one or more other computing devices 562 over a network communication link via one or more communication ports 564.

The network communication link may be one example of a communication media. Communication media may typically be embodied by computer-readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and may include any information delivery media. A “modulated data signal” may be a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media may include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), microwave, infrared (IR), and other wireless media. The term “computer-readable media” as used herein may include both storage media and communication media.

The computing device 500 may be implemented as a portion of a small-form factor portable (or mobile) electronic device such as a smartphone, a personal data assistant (PDA) or an application-specific device. The computing device 500 may also be implemented as a personal computer including tablet computer, laptop computer, and/or non-laptop computer configurations, or a server computer including both rack-mounted server computer and blade server computer configurations.

Embodiments described herein may be implemented using computer-readable media for carrying or having computer-executable instructions or data structures stored thereon. Such computer-readable media may be any available media that may be accessed by a general-purpose or special-purpose computer. By way of example, such computer-readable media may include non-transitory computer-readable storage media including RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, flash memory devices (e.g., solid state memory devices),

or any other storage medium which may be used to carry or store desired program code in the form of computer-executable instructions or data structures and which may be accessed by a general-purpose or special-purpose computer. Combinations of the above may also be included within the scope of computer-readable media.

5 Computer-executable instructions may include, for example, instructions and data which cause a general-purpose computer, special-purpose computer, or special-purpose processing device (e.g., one or more processors) to perform a certain function or group of functions. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined
10 in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

Unless specific arrangements described herein are mutually exclusive with one another, the various implementations described herein can be combined to enhance system
15 functionality or to produce complementary functions. Likewise, aspects of the implementations may be implemented in standalone arrangements. Thus, the above description has been given by way of example only and modification in detail may be made within the scope of the present invention.

With respect to the use of substantially any plural or singular terms herein, those
20 having skill in the art can translate from the plural to the singular or from the singular to the plural as is appropriate to the context or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity. A reference to an element in the singular is not intended to mean “one and only one” unless specifically stated, but rather “one or more.” Moreover, nothing disclosed herein is intended to be dedicated to the
25 public regardless of whether such disclosure is explicitly recited in the above description.

In general, terms used herein, and especially in the appended claims (e.g., bodies of the appended claims) are generally intended as “open” terms (e.g., the term “including” should be interpreted as “including but not limited to,” the term “having” should be interpreted as “having at least,” the term “includes” should be interpreted as “includes but
30 is not limited to,” etc.). Furthermore, in those instances where a convention analogous to “at least one of A, B, and C, etc.” is used, in general, such a construction is intended in the sense one having skill in the art would understand the convention (e.g., “a system having at least one of A, B, and C” would include but not be limited to systems that include A alone, B alone, C alone, A and B together, A and C together, B and C together, or A, B,

and C together, etc.). Also, a phrase presenting two or more alternative terms, whether in the description, claims, or drawings, should be understood to include one of the terms, either of the terms, or both terms. For example, the phrase “A or B” will be understood to include the possibilities of “A” or “B” or “A and B.”

5 The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes
10 which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

What is claimed is:

1. A computer-implemented method to group graphical objects, the method
5 comprising:
displaying, on a display device, a graphical diagram with a plurality of graphical
objects that represent data of a data source;
receiving input to define one or more groups; and
in response to the input:
10 generating one or more containers, each of the one or more containers
representing a different one of the one or more groups; and
graphically depicting membership of the plurality of graphical objects in the
one or more groups by relative arrangement of the plurality of graphical objects and
the one or more containers according to group membership of each of the plurality
15 of graphical objects.
2. The computer-implemented method of claim 1, wherein the input comprises first
input to define an arbitrary group, the method further comprising, prior to receiving the first
input, receiving second input to simultaneously select two or more graphical objects of the
20 plurality of graphical objects to include in the arbitrary group.
3. The computer-implemented method of claim 1, wherein the input comprises first
input to select a pre-existing container and second input to group graphical objects located
within the pre-existing container into the one or more containers according to a respective
25 value of an attribute of each of the graphical objects.
4. The computer-implemented method of claim 3, wherein:
the plurality of graphical objects represents data objects of the data source, the data
objects having a hierarchy;
30 the one or more containers include two or more containers generated within the pre-
existing container for two or more different subsets of the plurality of graphical objects
each grouped together in a corresponding one of the two or more containers;
two or more graphical objects grouped together in a first container of the two or
more containers each has a first value for the attribute;

two or more other graphical objects grouped together in a second container of the two or more containers each has a second value for the attribute that is different than the first value; and

the grouping together of the graphical objects located within the pre-existing container into the two or more containers applies exclusively to graphical objects that represent data objects at a first level of the hierarchy below a second level of the hierarchy represented by the pre-existing container.

5. The computer-implemented method of claim 1, wherein:
the input comprises first input including a selection of an attribute to group graphical objects;
the method further comprises identifying unique values of the attribute across the graphical objects; and
the one or more containers include at least one container for each of the unique values of the attribute.

6. The computer-implemented method of claim 1, wherein:
the plurality of graphical objects represents data objects of the data source, the data objects having a hierarchy;
the hierarchy is graphically represented by arrangement of the plurality of graphical objects within multiple nested containers;
the method further comprises preserving the hierarchy while graphically depicting membership of the plurality of graphical objects in the one or more groups.

7. The computer-implemented method of claim 6, further comprising receiving second input to select:
a level of the hierarchy at which to arrange the one or more containers; or
a cross-layout arrangement in which to arrange the one or more containers with the nested containers.

8. The computer-implemented method of claim 6, wherein the one or more containers include at least a first container and a second container and wherein preserving the hierarchy while graphically depicting membership of the plurality of graphical objects in the one or more groups includes one of:

arranging the first and second containers in a cross-layout arrangement with the nested containers;

duplicating the nested containers and arranging the first and second containers at a highest level of the hierarchy, including:

5 positioning a first duplicate of the nested containers inside the first container;

 positioning graphical objects that belong to a first group represented by the first container within the first duplicate of the nested containers;

 positioning a second duplicate of the nested containers inside the second
10 container; and

 positioning graphical objects that belong to a second group represented by the second container within the second duplicate of the nested containers; or

 arranging the first and second containers at a level of the hierarchy beneath the highest level of the hierarchy.

15

9. The computer-implemented method of claim 6, wherein:

 the one or more containers include at least a first container and a second container;

 preserving the hierarchy while graphically depicting membership of the plurality of graphical objects in the one or more groups includes arranging the first container at each
20 level of the hierarchy that includes at least one graphical object that belongs to the first group represented by the first container and the second container at each level of the hierarchy that includes at least one graphical object that belongs to the second group represented by the second container, including:

 distributing duplicates of the first container across the nested containers
25 such that each set of one or more graphical objects that belong to the first group at each hierarchical level immediately beneath a corresponding one of the nested containers is positioned within one of the duplicates of the first container; and

 distributing duplicates of the second container across the nested containers
 such that each set of one or more graphical objects that belong to the second group
30 at each hierarchical level immediately beneath a corresponding one of the nested containers is positioned within one of the duplicates of the second container; and
 the method further includes:

 receiving second input to select one of the duplicates of the first or second container; and

in response to the second input, altering the graphical diagram to indicate that all other duplicates of the first or second container are duplicates of the same container as the selected duplicate.

5 10. The computer-implemented method of claim 9, wherein the altering of the graphical diagram comprises at least one of:

highlighting all other duplicates of the first or second container; or

displaying connectors between the selected duplicate of the first or second container and all other duplicates of the first or second container.

10

11. A non-transitory computer-readable medium having computer-readable instructions stored thereon that are executable by a processor to perform or control performance of operations comprising:

displaying, on a display device, a graphical diagram with a plurality of graphical

15 objects that represent data of a data source;

receiving input to define one or more groups; and

in response to the input:

generating one or more containers, each of the one or more containers representing a different one of the one or more groups; and

20 graphically depicting membership of the plurality of graphical objects in the one or more groups by relative arrangement of the plurality of graphical objects and the one or more containers according to group membership of each of the plurality of graphical objects.

25 12. The non-transitory computer-readable medium of claim 11, wherein the input comprises first input to define an arbitrary group, the operations further comprising, prior to receiving the first input, receiving second input to simultaneously select two or more graphical objects of the plurality of graphical objects to include in the arbitrary group.

30 13. The non-transitory computer-readable medium of claim 11, wherein the input comprises first input to select a pre-existing container and second input to group graphical objects located within the pre-existing container into the one or more containers according to a respective value of an attribute of each of the graphical objects.

14. The non-transitory computer-readable medium of claim 13, wherein:
the plurality of graphical objects represents data objects of the data source, the data objects having a hierarchy;
the one or more containers include two or more containers generated within the pre-existing container for two or more different subsets of the plurality of graphical objects each grouped together in a corresponding one of the two or more containers;
two or more graphical objects grouped together in a first container of the two or more containers each has a first value for the attribute;
two or more other graphical objects grouped together in a second container of the two or more containers each has a second value for the attribute that is different than the first value; and
the grouping together of the graphical objects located within the pre-existing container into the two or more containers applies exclusively to graphical objects that represent data objects at a first level of the hierarchy below a second level of the hierarchy represented by the pre-existing container.
15. The non-transitory computer-readable medium of claim 11, wherein:
the input comprises first input including a selection of an attribute to group graphical objects;
the operations further comprise identifying unique values of the attribute across the graphical objects; and
the one or more containers include at least one container for each of the unique values of the attribute.
16. The non-transitory computer-readable medium of claim 11, wherein:
the plurality of graphical objects represents data objects of the data source, the data objects having a hierarchy;
the hierarchy is graphically represented by arrangement of the plurality of graphical objects within multiple nested containers;
the operations further comprise preserving the hierarchy while graphically depicting membership of the plurality of graphical objects in the one or more groups.
17. The non-transitory computer-readable medium of claim 16, the operations further comprising receiving second input to select:

a level of the hierarchy at which to arrange the one or more containers; or
a cross-layout arrangement in which to arrange the one or more containers with the
nested containers.

- 5 18. The non-transitory computer-readable medium of claim 16, wherein the one or
more containers include at least a first container and a second container and wherein
preserving the hierarchy while graphically depicting membership of the plurality of
graphical objects in the one or more groups includes one of:

arranging the first and second containers in a cross-layout arrangement with the
10 nested containers;

duplicating the nested containers and arranging the first and second containers at a
highest level of the hierarchy, including:

positioning a first duplicate of the nested containers inside the first
container;

15 positioning graphical objects that belong to a first group represented by the
first container within the first duplicate of the nested containers;

positioning a second duplicate of the nested containers inside the second
container; and

positioning graphical objects that belong to a second group represented by
20 the second container within the second duplicate of the nested containers; or

arranging the first and second containers at a level of the hierarchy beneath the
highest level of the hierarchy.

19. The non-transitory computer-readable medium of claim 16, wherein:
25 the one or more containers include at least a first container and a second container;
preserving the hierarchy while graphically depicting membership of the plurality of
graphical objects in the one or more groups includes arranging the first container at each
level of the hierarchy that includes at least one graphical object that belongs to the first
group represented by the first container and the second container at each level of the
30 hierarchy that includes at least one graphical object that belongs to the second group
represented by the second container, including:

distributing duplicates of the first container across the nested containers
such that each set of one or more graphical objects that belong to the first group at

each hierarchical level immediately beneath a corresponding one of the nested containers is positioned within one of the duplicates of the first container; and

5 distributing duplicates of the second container across the nested containers such that each set of one or more graphical objects that belong to the second group at each hierarchical level immediately beneath a corresponding one of the nested containers is positioned within one of the duplicates of the second container; and the operations further include:

receiving second input to select one of the duplicates of the first or second container; and

10 in response to the second input, altering the graphical diagram to indicate that all other duplicates of the first or second container are duplicates of the same container as the selected duplicate.

20. The non-transitory computer-readable medium of claim 19, wherein the altering of
15 the graphical diagram comprises at least one of:

highlighting all other duplicates of the first or second container; or

displaying connectors between the selected duplicate of the first or second container and all other duplicates of the first or second container.