

VISUALIZATION CUSTOMIZATION

FIELD

The embodiments discussed herein are related to visualization customization.

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 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 visualizations on a computer based on the structured source data. For example, a user may manually create from the structured source data a diagram that depicts data objects and their relationships, e.g., by manually creating in a graphical drawing canvas graphical objects that represent the data objects and their relationships.

Sometimes the underlying structured source data may be updated and the user may desire the updates to be reflected in the visualization. One existing option to update the visualization includes manually generating the visualization again from scratch based on the updated structured source data and/or manually modifying the visualization to reflect the updates if the updates are relatively few, which can be bothersome and/or time-consuming when the visualization includes many nodes.

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.

BRIEF SUMMARY OF SOME EXAMPLE EMBODIMENTS

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.

Some embodiments described herein generally relate to visualization customization.

In an example embodiment, a method to customize a visualization may include generating a semantic layout of a visualization based on a contract associated with a visualization type. The semantic layout may specify one or more of relative locations, arrangement, and relationships of data objects to represent in the visualization. The data objects
5 may include at least some data of a data source. The method may also include receiving input effective to alter a set of one or more graphical objects to display for each of some but not all of the data objects from a default set of one or more graphical objects to a custom set of one or more graphical objects. The default set is different than the custom set. The method may also include generating a concrete layout of the visualization that specifies the set of one or more
10 graphical objects to display for each of the data objects. The concrete layout may specify the default set for a first data object and the custom set for a second data object. The method may also include displaying the visualization of the at least some data of the data source based on the contract, the semantic layout, and the concrete layout, the visualization having the visualization type.

In another example embodiment, a non-transitory computer-readable storage medium
15 has computer-executable instructions stored thereon that are executable by a processor device to perform or control performance of operations that may include generating a semantic layout of a visualization based on a contract associated with a visualization type. The semantic layout may specify one or more of relative locations, arrangement, and relationships of data objects to represent in the visualization. The data objects may include at least some data of a data
20 source. The operations may also include receiving input effective to alter a set of one or more graphical objects to display for each of some but not all of the data objects from a default set of one or more graphical objects to a custom set of one or more graphical objects. The default set is different than the custom set. The operations may also include generating a concrete
25 layout of the visualization that specifies the set of one or more graphical objects to display for each of the data objects. The concrete layout may specify the default set for a first data object and the custom set for a second data object. The operations may also include displaying the visualization of the at least some data of the data source based on the contract, the semantic layout, and the concrete layout, the visualization having the visualization type.

Additional features and advantages of the invention will be set forth in the description
30 which follows, and in part will be obvious from the description, or may be 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.

BRIEF DESCRIPTION OF THE DRAWINGS

5 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 therefore not to be considered limiting of its scope. The invention will be described and explained with additional
10 specificity and detail through the use of the accompanying drawings in which:

Figure 1 is a block diagram of an example architecture to generate, customize, and/or interact with visualizations of data from data sources based on contracts;

Figure 2 illustrates various example visualizations with various example visualization types;

15 Figure 3 illustrates two example contracts;

Figure 4 illustrates an example visualization file from which a visualization may be generated;

Figure 5 is a block diagram of an example operating environment in which the architecture of Figure 1 may be implemented;

20 Figure 6 is a flowchart of a method to generate a visualization of data from a data source based on a contract;

Figure 7 is a flowchart of a method to customize a visualization of data from a data source based on a contract;

25 Figure 8 is a flowchart of a method to interact with a visualization of data from a data source based on a contract;

Figures 9A-9F include graphical representations of a user interface to generate, customize, and/or interact with visualizations; and

Figure 10 is a block diagram illustrating an example computing device that is arranged for generating, customizing, and/or interacting with visualizations,

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

DETAILED DESCRIPTION OF SOME EXAMPLE EMBODIMENTS

Diagrams representing source data may be generated manually or automatically. Manually generating diagrams and other visualizations can be time-consuming, tedious, and/or prone to user error. Automatically generating diagrams can be less time-consuming, less tedious, and/or less prone to user error than manually generating diagrams. However, applications to automatically generate diagrams from source data may be source-dependent, e.g., such applications may be limited to use with a particular type or format of source data, and/or may be limited to generating only one specific type of diagram.

In addition, while some automatic diagram generation applications may allow a user to choose a particular theme to apply to an entire diagram, such applications generally do not permit the user to customize specific objects in or represented in the diagram.

Further, any changes the user makes to the diagram typically do not affect the underlying source data. Thus, even if some applications permit the rearrangement of objects in the diagram and/or the relationships between objects in the diagram, such changes may be limited to the diagram and may not be propagated back to the source data.

Some embodiments described herein may address one or more of the foregoing problems with some visualization applications. For example, at least one embodiment may automatically generate diagrams based on source data from any source, which source data may be in virtually any format at the source. In this and other embodiments, the source data may be translated from its original format to a standardized format. The translation may be implemented by an extract, transform, and load (ETL) process. Alternatively or additionally, various portions of the same source data may be used to generate visualizations of various visualization types that may be associated with different data contracts (hereinafter “contracts”). Each contract may generally specify a structure of data necessary to generate a visualization with a specific visualization type associated with the contract.

In at least one embodiment, each data object may be represented in a visualization by a set of one or more graphical objects. The set of one or more graphical objects may include a default set that may be customized for a given data object. For example, the default set of one or more graphical objects for a given data object may include a default outline shape with a default outline line type and a default font style (font type, font size, font color). For one or more of the data objects, the default set of one or more graphical objects may be customized as a custom set of one or more graphical objects, e.g., to include more or fewer graphical objects than in the default set, to include graphical objects with a custom outline shape, a custom outline line type, a custom font style, or other custom aspects. The customization may be applied, e.g., manually to one or more specific data objects, or may be applied, e.g.,

automatically through conditional customization to any data object that satisfies a corresponding condition.

In at least one embodiment, any changes made to a visualization may implicate changes to the underlying source data, and/or may be propagated back to the source data to update the source data consistent with the changes to the visualization. In this and other embodiments, data of the data source may be mapped to a contract such that the contract is satisfied. The visualization may be displayed with multiple graphical objects that graphically represent the data of the data source. One or more graphical objects in the visualization and/or relationships between graphical objects may be altered. Semantically these alterations may be understood as implicating changes to data objects in the data of the data source. Accordingly, the data objects may be altered consistent with the alterations to the graphical objects. In some embodiments, all changes to the visualization and/or implicated changes to the underlying source data may be saved locally, e.g., while a user has the visualization open and is editing it. At some point, e.g., when the user is closing the visualization, all of the locally saved changes may be presented to the user and the user may, e.g., choose to propagate a given implicated change back to the source data, save a given change locally without propagating it back to the source data, and/or completely discard a given change such that it is neither propagated back to the source data nor saved in the visualization. Alternatively or additionally, a user may be prompted to choose what to do about each change every time a changes is made, rather than going through multiple changes at once.

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.

Figure 1 is a block diagram of an example architecture 100 to generate, customize, and/or interact with visualizations of data from data sources based on contracts, arranged in accordance with at least one embodiment described herein. The architecture 100 may include one or more data sources 102, 104, 106 and a data model 108.

Each of the data sources 102, 104, 106 may include one or more collections of data, referred to as data collections. The data collections may be data collections of a corresponding entity, such as a business, a company, an agency, a municipality, an individual and/or other entity. In an example, the data source 102 and/or one or more data collections thereon may be owned by or otherwise associated with one entity, the data source 104 and/or one or more data collections thereon may be owned by or otherwise associated with another entity, and the data

source 106 and/or one or more data collections thereon may be owned by or otherwise associated with still another entity. Alternatively or additionally, two or more data sources may be owned by or otherwise associated with the same entity.

Each of the data sources 102, 104, 106 and/or the data collections may include, for example, 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 cloud infrastructure provider such as AWS, GOOGLE CLOUD, or MICROSOFT AZURE; 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 102, 104, 106 and/or the data collections may have a specific format, which may be different and/or the same from one data source 102, 104, 106 to another and/or from one data collection to another. Some example data formats include proprietary formats, Structured Query Language (SQL), and JavaScript Object Notation (JSON).

To allow the same or similar operations (e.g., generation of visualizations) to be performed on data from different ones of the data sources 102, 104, 106, from different data collections, and/or with different data formats at the data sources 102, 104, 106, the data from the data sources 102, 104, 106 may be standardized before being saved in the data model 108. Accordingly, the architecture 100 may include one or more ETL modules 110, 112, 114 to translate data of the data sources 102, 104, 106 from a first format (e.g., proprietary, SQL, JSON, or other data format) to a standardized format in which the data is saved on the data model 108.

By way of example, a service provider that provides and/or supports the data model 108 and/or provides various operations (e.g., generation of visualizations) that can be done on data in the data model 108 may provide the ETL module 110 to translate data of the data source 102 from one or more non-standardized formats to a standardized format that is standardized for the data model 108 and the operations provided by the service provider. Alternatively or additionally, the service provider may provide an application programming interface 116 that permits the entity associated with the data source 104 or some other entity the ability to generate a proprietary ETL module, e.g., the ETL module 112, to translate data of the data source 104 from one or more non-standardized formats to the standardized format. Alternatively or

additionally, the ETL module 114, whether provided by the service provider or a different entity, may translate data of the data source 106 from one or more non-standardized formats to the standardized format, while an API 118 may allow the service provider to update source data of the data source 106 responsive to, e.g., a change made to a visualization of at least some
5 of the source data of the data source 106.

With continued reference to Figure 1, the architecture 100 may further include one or more contracts 120, 122, 124 to generate one or more visualizations 126, 128, 130, 132 of data from the data sources 102, 104, 106. Each of the contracts 120, 122, 124 and other contracts described herein may specify a structure of data necessary to produce one or more
10 visualizations, such as the visualizations 126, 128, 130, 132.

Each of the visualizations 126, 128, 130, 132 may have a visualization type associated with the corresponding contract 120, 122, 124. For example, generating a visualization of a specific visualization type may require a specific contract that specifies the structure of data necessary to produce the visualization with the specific visualization type. In some cases,
15 multiple visualization types may be associated with the same contract. For example, generating a visualization with any of two or more specific visualization types may require the same specific contract that specifies the structure of data necessary to produce the visualization with either of the two or more specific visualization types. For example, the visualization 128 may have one visualization type that is associated with the contract 122, while the visualization 130
20 may have a different visualization type that is also associated with the contract 122. In other cases, some contracts may each be associated with only a single visualization type.

Figure 2 illustrates various example visualizations 200A, 200B, 200C, 200D (hereinafter collectively “visualizations 200”) with various example visualization types, arranged in accordance with at least some embodiments described herein. The visualization
25 200A has a mind map visualization type that organizes information in a hierarchical manner to show relationships among pieces of a whole. The visualization 200B has a container visualization type in which different objects that have a same feature are depicted within a same group or container. The visualization 200C has an abstraction layers visualization type that may, e.g., generalize conceptual models or algorithms with increasing order of complexity in
30 one direction and increasing order of abstraction in the other direction, as indicated by the arrows 202, 204. The visualization 200D has a hierarchical organizational chart visualization type that shows the structure of an organization and the relationships and relative ranks of its parts and positions/jobs, where every entity in the organization, except one, is subordinate to a single other entity. More generally, the visualizations that may be generated according to

embodiments described herein may have any suitable visualization type, such as organizational chart (e.g., hierarchical organizational chart, matrix organizational chart, flat organizational chart), container, mind map, abstraction layer, logical network diagram, flowchart, supply chain process map, concept map, or other suitable visualization type.

Returning to Figure 1, each of the contracts 120, 122, 124 and other contracts described herein may include one or more corresponding schema that describe a corresponding set of and corresponding type of fields required to satisfy the corresponding contract 120, 122, 124. Alternatively or additionally, each schema may specify a set of one or more data fields corresponding to specific data from a corresponding one of the data sources 102, 104, 106, e.g., in a standardized format in the data model 108, and one or more relationships between data fields in the set. Each schema may be casually referred to as a shape of underlying source data. E.g., data of a corresponding one of the data sources 102, 104, 106 that satisfies a particular schema, or has a particular “shape”, may be used to generate a visualization having the visualization type described by the particular schema.

To generate a corresponding one of the visualizations 126, 128, 130, 132 of data from a given one of the data sources 102, 104, 106, a user may select a visualization type that is associated with a given one of the contracts 120, 122, 124 to apply to the visualization 126, 128, 130, 132. At least some data from the corresponding one of the data sources 102, 104, 106, e.g., in the standardized format in the data model 108, may then be mapped to the associated one of the contracts 120, 122, 124 such that the contract 120, 122, 124 and/or the corresponding schema is satisfied. In at least one embodiment, the contract 120, 122, 124 may be considered satisfied if the mapping identifies specific data from the data source 102, 104, 106 that satisfies and/or is compatible with the set of data fields and relationships of the schema of the contract 120, 122, 124.

After the mapping, the corresponding one of the visualizations 126, 128, 130, 132 may then be displayed based on the corresponding contract 120, 122, 124, where the displayed visualization 126, 128, 130, 132 has the selected visualization type.

Figure 3 illustrates two example contracts 300A, 300B (collectively “contracts 300”) and example data collections 302A, 302B (collectively “data collections 302”) from a data source, arranged in accordance with at least one embodiment described herein.

The contract 300A may be or include a simple organizational chart contract to generate visualizations with an organizational chart visualization type, such as one or more of hierarchical organizational chart, matrix organizational chart, or flat organizational chart. The contract 300A includes a schema that defines three data fields—ID, Name, and Parent ID—

that have a specified relationship in which the ID, Name, and Parent ID are associated with a specific entity (e.g., an employee that has an ID and a Name and that reports to another entity (e.g., a manager or supervisor) identified by the Parent ID) as a data object to be represented in the visualization, where the Parent ID for one data object is the ID for another data object.

- 5 The contract 300A may permit one or a few data objects, e.g., corresponding to one or more owners, executives, or other entities of an organization that are at a top of a hierarchy of the organization and that do not report to anyone else, to not have a Parent ID.

In at least one embodiment, a contract such as the contract 300A, and/or a schema included in the contract, may include one or more optional data fields and/or relationships to
10 describe more unusual relationships. For example, some employees may report to multiple managers or supervisors and the contract 300A may have two or more “Parent ID” fields instead of one, where one of the “Parent ID” fields is required and one or more other “Parent ID” fields are optional.

In some embodiments, an organizational chart contract may be more complex or
15 different than the contract 300A of Figure 3, e.g., may include more, fewer, and/or different data fields than illustrated in Figure 3. For example, such organization chart contracts may include an employee photo Uniform Resource Identifier (URI) data field, an office location data field, and/or other data fields.

The contract 300B may be or include a container contract to generate visualizations
20 with a container visualization type. The contract 300B includes a schema that defines two data fields—Tag and Name—that have a specific relationship in which the Tag and Name are associated with a specific entity (e.g., an employee that has a Name and that is on a specific team within an organization where the team is designated by a Tag) as a data object to be represented in the visualization, where one or more multiple data objects may have the same
25 Tag. In some embodiments, a container contract may be more complex or different than the contract 300B of Figure 3, e.g., may include more, fewer, and/or different data fields than illustrated in Figure 3.

The data collection 302A may include an employee data collection that includes employee IDs, First Names, Last Names, and Team designations for each employee of an
30 organization. The data collection 302A may include data from a data source, e.g., an HR database, and/or the data may be in a standardized, tabular format, e.g., in the data model 108 of Figure 1.

The data collection 302B may include a reports to data collection that includes employee IDs for the employees and an identification of, e.g., a manager, a supervisor, or other

individual in the organization each employee reports to. The identification of the manager, supervisor, or other individual in the organization that each employee reports to may include the manager's or supervisor's employee ID in an example embodiment, or other identifying information for the manager or supervisor. The data collection 302B may include data from a data source, e.g., a spreadsheet that identifies each employee and each employee's supervisor, and/or the data may be in a standardized, tabular format, e.g., in the data model 108 of Figure 1.

Figure 3 additionally illustrates contract maps 304, 306 to map data from the data collections 302 to the contracts 300. Each contract map 304, 306 may include one or more pointers to data of a data source, e.g., to data of one or both of the data collections 302 in these examples. In particular, the contract map 304 includes pointers 304A, 304B, and 304C and the contract map 306 includes pointers 306A, 306B. The pointers 304A, 304B, 304C, 306A, 306B point to the appropriate data of the data source(s) 302 to satisfy the corresponding contract 300.

For example, the pointer 304A of the contract map 304 specifies that data for the ID data field in the contract 300A is in the ID field of the data collection 302A. The pointer 304B of the contract map 304 specifies that data for the Name data field in the contract 300A is a concatenation of the data in the First Name field and the Last Name field of the data collection 302A. The pointer 304C of the contract map 304 specifies that data for the Parent ID data field in the contract 300A is in the Reports To field of the data collection 302B.

Analogously, the pointer 306A of the contract map 306 specifies that data for the Tag data field in the contract 300B is in the Team field of the data collection 302A. The pointer 306B of the contract map 306 specifies that data for the Name data field in the contract 300B is a concatenation of the data in the First Name field and the Last Name field of the data collection 302A.

Accordingly, the data source that includes the data collections 302 may include one or more data objects. Each visualization may graphically represent one or more of the data objects using one or more graphical objects. In particular, each data object may be represented by or otherwise associated with a set of one or multiple graphical objects. For example, a data object 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.

Alternatively or additionally, in at least one embodiment, one graphical object may represent multiple data objects. As described below, for example, some portions of an organizational chart or other visualization may be selectively collapsed, in which case multiple employees (e.g., multiple data objects) may be hidden and may instead be represented by a suitable graphical object (such as a user interface element 934 in Figure 9F). As another example, an entire team of multiple employees may be represented by a single shape for the team.

Each visualization described herein may be generated based on a contract, e.g., the contracts 300, from a semantic layout and a concrete layout. The semantic layout may be generated based on a corresponding contract, such as the contracts 300, and may specify one or more of relative locations, arrangement, and relationship of multiple data objects to represent in a visualization. The concrete layout may specify a set of one or more graphical objects to display for each of the data objects to be represented in the visualization. In some embodiments, the concrete layout may specify a default set of one or more graphical objects for each of the data objects. Alternatively or additionally, the concrete layout may specify a custom set of one or more graphical objects for a subset of the data objects. In at least one embodiment, the custom set of one or more graphical objects may be specified in the concrete layout responsive to receiving input, e.g., from a user, effective to alter the default set to the custom set for the subset of the data objects. The customization may be provided for individual data objects and/or for classes or groups of data objects by using conditional customization.

The custom set of one or more graphical objects may be customized and different in at least one respect compared to the default set of one or more graphical objects, although there may be some overlap between the custom set and the default set. For example, the default set of one or more graphical objects may include an outline with a specific shape, color, line type (e.g., solid, dashed, dot-dash, etc.), and line weight and text of a specific font, size, and color. In comparison, at least one of the foregoing outline shape, outline color, outline line type, outline line weight, text font, text size, and/or text color may be different for the custom set of one or more graphical objects.

In some examples, graphical objects in the visualization that represent data objects may be altered, e.g., by a user. The alterations to the graphical objects may implicate changes to the data objects. For example, in a visualization with an organizational chart visualization type, if a connector or other graphical object that connects an employee to the employee's manager or supervisor is modified to connect the employee to a different manager, supervisor, or other entity, this may indicate that the user desires to change the employee's manager or supervisor.

As another example, in a visualization with a container chart visualization type, assume containers represent different groups or teams, with different boxes or text within each container representing members of the corresponding group or team; if a box or text representing a specific member of one group or team is moved by a user from a first container of a first group or team to a different second container of a second group or team, this may indicate the user desires to change the specific member from being on the first group or team to being on the second group or team. As still another example, in a visualization with a logical network diagram visualization type, if a connector between a port of a first device and a first port of a second device is altered to connect the port of the first device to a second port of the second device or to a port of a third device, this may indicate the user desires to change the port(s) over which the first and second devices communicate and/or the device(s) which devices communicate with each other.

At least one embodiment described herein includes updating the underlying source data to reflect changes made to visualizations. In the examples provided above, for instance, source data may be updated by, e.g., changing an employee's manager or supervisor at a corresponding data source, changing an employee or other entity's membership in a group or team at the data source, and/or changing the ports over which and/or the devices with which a device communicates at the data source. Changes to the underlying data may be implemented automatically in response to alteration of the visualization, or after querying the user about whether to change the underlying data and receiving an affirmative response to the query. The user may be queried about whether to update the underlying data every time an alteration is made, or in batches, e.g., periodically or when the user closes the visualization after making the a set of one or more alterations.

Figure 4 illustrates an example visualization file 400 from which a visualization may be generated, arranged in accordance with at least one embodiment described herein. The visualization file 400 may be stored on a user device and/or on a server device or in other storage location. The visualization file 400 may include a contract 402, a corresponding contract map 404, a visualization type 406, a differential log 408, and customizations 410.

The contract 402 may include or correspond to one or more of the contracts described elsewhere herein.

The contract map 404 may include or correspond to one or more of the contract maps described elsewhere herein.

The visualization type 406 may include or correspond to one or more of the visualization types described elsewhere herein and/or may a user's selection of a specific visualization type to apply to a visualization the user desires to generate.

5 The differential log 408 may include a list of one or more alterations the user has made to the visualization, e.g., to one or more graphical objects, and/or a list of one or more implicated changes to the underlying data of the data source used to generate the visualization.

The customizations 410 may include customizations to a set of one of more graphical objects for each of one or more specific data objects and/or conditional customization that customizes the set of one or more graphical objects for each data object that satisfies a
10 predetermined condition.

The visualization file 400 does not include the underlying source data, e.g., data or data objects from a data source. Thus, when a user "opens" the visualization file 400 to display and view the corresponding visualization or syncs to the data source, the visualization may be regenerated from the source data. As such, changes made to the source data may be reflected
15 in the regenerated visualization without the user having to manual update the visualization.

Figure 5 is a block diagram of an example operating environment 500 in which the architecture 100 of Figure 1 may be implemented, arranged in accordance with at least one embodiment described herein. The operating environment 500 may include a server 502 and one or more client devices 504, 506, 508. The server 502 and/or the client devices 504, 506,
20 508 may be configured to generate, customize, and/or interact with visualizations of source data from one or more data sources 510 based on contracts. Alternatively or additionally, the source data of the data sources 510 may be updated based on changes to the visualizations.

Each of the one or more data sources 510 may include or correspond to the data sources described elsewhere herein.

25 Although one server 502 and three client devices 504, 506, 508 are illustrated in Figure 5, the operating environment 500 may more generally include one or more servers 502 and one or more client devices 504, 506, 508. In these and other embodiments, the operating environment 500 may include other servers and/or devices not illustrated in Figure 5.

The operating environment 500 may additionally include a network 512. In general, the
30 network 512 may include one or more wide area networks (WANs) and/or local area networks (LANs) that enable the server 502, the client devices 504, 506, 508, and the data sources 510 to communicate with each other. In some embodiments, the network 512 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 512 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 512 may also include servers that enable one type of network to interface with another type of network.

5 In at least one embodiment, the server 502 may host a web-based visualization application (“application 514”) that allows the client devices 504, 506, 508 and/or users thereof to generate, customize, and/or interact with visualizations as described herein. In other embodiments, the application 514 may include a non-web-based application but may generally be described herein as a web-based application for simplicity. Alternately or additionally, the
10 server 502 may facilitate updates to the data sources 510 based on changes to the visualizations as described elsewhere herein.

The server 502 may additionally include a processor 516 and a storage medium 518. The processor 516 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
15 combination thereof. The processor 516 may be configured to execute computer instructions that, when executed, cause the processor 516 to perform or control performance of one or more of the operations described herein with respect to the server 502.

The storage medium 518 may include volatile memory such as random access memory (RAM), persistent or non-volatile storage such as read only memory (ROM), electrically
20 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 518 may store computer instructions that may be executed by the processor 516 to perform or control performance of one or more of the operations described
25 herein with respect to the server 502.

The storage medium 518 may additionally store one or more data models 520 and/or visualization files 522. The data models 520 may each include or correspond to the data model 108 of Figure 1, for example. In at least one embodiment, one data model 520 may be generated for each data source 510. In at least one embodiment, one data model 520 may be generated
30 for multiple data sources 510.

The visualization files 522 may each include or correspond to the visualization file 400 of Figure 4, for example. Each visualization file may generate a corresponding visualization, each visualization being a visualization of at least some data from a corresponding one of the

data sources 510, which data from the data sources 502 may be available to the server 502 in a standardized format in the data models 520.

Source data at the data sources 510 may undergo changes over time. As such, the server 502 may from time to time update the data models 520 consistent with updates to the data sources 510. For example, the server 502 may include or be communicatively coupled to one or more ETL modules, such as the ETL modules 110, 112, 114 of Figure 1 that perform an ETL process on the data sources 510 from time to time to generate and/or update the data models 520 on the server 502.

Alternatively or additionally, the server 502 may track changes made to data objects in a visualization and/or to graphical objects that represent the data objects in the visualization. The changes may be tracked in, e.g., a differential log of each of the visualization files 522, such as the differential log 408 of Figure 4. A user may then be queried or otherwise prompted about whether to update the source data in view of the changes to the visualization that may implicate changes to the source data.

In some embodiments, the server 502 may be configured to cooperate with the client devices 504, 506, 508 to allow the client devices 504, 506, 508 to generate, customize, and/or interact with visualizations of at least some data of the data sources 510 and/or to update at least some data of the data sources 510 based on alterations to the visualizations. In these and other embodiments, client devices 504, 506, 508 may use a corresponding application or app, such as a browser 524, to generate the visualizations from the visualization files 522, which visualizations may be displayed on a display of the corresponding client device 504, 506, 508.

Accordingly, each of the client devices 504, 506, 508 may execute an application, such as the browser 524, configured to communicate through the network 512 with the server 502. Each of the other client devices 519, 521 may be similarly configured to execute a browser, such as the browser 524. The browser 524 may include an Internet browser or other suitable application for communicating through the network 512 with the server 502. The browser 524 may generate, download and/or interact with visualizations and/or visualization files 522. Each of the client devices 504, 506, 508 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.

Each of the client devices 504, 506, 508 may additionally include a processor and a storage medium, such as a processor 526 and a storage medium 528 as illustrated for the client device 504 in Figure 5. Each of the other client devices 506, 508 may be similarly configured. Similar to the processor 516 of the server 502, the processor 526 may be of any type such as a

CPU, a μ P, a μ C, a DSP, or any combination thereof. The processor 526 may be configured to execute computer instructions that, when executed, cause the processor 526 to perform or control performance of one or more of the operations described herein with respect to the client device 504 and/or the browser 524.

5 Similar to the storage medium 518 of the server 502, the storage medium 528 of the client device 504 may include 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 528 may store
10 computer instructions that may be executed by the processor 526 to perform one or more of the operations described herein with respect to the client device 504 and/or the browser 524. The storage medium 528 may additionally store, at least temporarily, a visualization file 522 and/or other content or data obtained from the server 502 and/or created locally on the client device 504.

15 Embodiments described herein are not limited to using a browser to communicate with the server 502 to generate, customize, and/or interact with visualizations. For example, rather than or in addition to a browser, the client devices 504, 506, 508 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,
20 customizing, and/or interacting with visualizations using a browser, a native app, or another suitable application on the client devices 504, 506, 508.

 Figure 6 is a flowchart of a method 600 to generate a visualization of data from a data source based on a contract, arranged in accordance with at least one embodiment described herein. The method 600 may be programmably performed or controlled by a processor in, e.g.,
25 a computer and/or server that executes a browser, visualization application, and/or other application to generate visualizations based on contracts. In an example implementation, the method 600 may be performed in whole or in part by the server 502 and/or the client device 504 of Figure 5. The method 600 may include one or more of blocks 602, 604, and/or 606. The method 600 may begin at block 602.

30 At block 602 (“Receive Input Effective To Select A Visualization Type Of Multiple Different Visualization Types To Apply To A Visualization Of Data”), input may be received that is effective to select a visualization type of multiple different visualization types to apply to a visualization of data. Various example visualization types from which the visualization type may be selected are described, e.g., with respect to Figure 2. Receiving the input may

include receiving user input. For example, a user may select a specific visualization and/or a specific visualization type from a drop-down menu in a user interface. The selected visualization type may be associated with a contract. Each of the different visualization types may be associated with a corresponding specific contract or multiple contracts. Each contract
5 may include a corresponding schema that specifies a set of data fields and one or more relationships between data fields in the set to generate and display a corresponding visualization having a corresponding visualization type. Block 602 may be followed by block 604.

At block 604 (“Map At Least Some Data Of A Data Source To A Contract Such That
10 The Contract Is Satisfied”), at least some data of a data source may be mapped to a contract such that the contract is satisfied. The data source may include, e.g., any of the data sources 102, 104, 106, 510 or other data sources described herein. The contract may include, e.g., any of the contracts 120, 122, 124, 300, 402 or other contracts described herein. The mapping may be performed by or using a corresponding contract map, such as, e.g., any of the contract maps
15 304, 306, 404 or other contract maps described herein. In at least one embodiment, at least some data of the data source may not have at the data source a schema or shape that matches the corresponding schema or shape of the contract until the at least some data of the data source is mapped to the contract.

Mapping at least some data of the data source to the contract at block 604 may include
20 identifying multiple specific data from multiple different data collections extracted from the data source. At least some of the specific data may satisfy a first data field of the set of data fields specified by the schema of the contract and at least some other data of the specific data may satisfy a second data field of the set of data fields specified by the schema of the contract. When combined, the multiple specific data from the multiple different data collections may
25 form a set of data having a structure necessary to produce the visualization with the selected visualization type.

Alternatively or additionally, mapping at least some data of the data source to the contract at block 604 may include identifying at least first data from a first data collection extracted from the data source and second data from a second data collection extracted from
30 the data source. The first data may satisfy a first data field of the set of data fields specified by the schema of the contract and the second data may satisfy a second data field of the set of data fields specified by the schema of the contract. When combined, the first data and the second data may form at least a part of a set of data having a structure necessary to produce the visualization with the selected visualization type. Block 604 may be followed by block 606.

At block 606 (“Display The Visualization Of The At Least Some Data Of The Data Source Based On The Contract, The Displayed Visualization Having The Selected Visualization Type”), the visualization of the at least some data of the data source is displayed based on the contract, where the displayed visualization has the selected visualization type.

5 Displaying the visualization of the at least some data based on the contract may include displaying graphical objects representing data objects of the data source to reflect relationships specified by the contract.

One skilled in the art will appreciate that, for this and other processes and methods disclosed herein, the functions performed in the processes and methods may be implemented
10 in differing order. Furthermore, the outlined steps and operations are only provided as examples, and some of the steps and operations may be optional, combined into fewer steps and operations, or expanded into additional steps and operations without detracting from the essence of the disclosed embodiments. For instance, the method 600 may be combined with one or more other methods and processes described herein.

15 Alternatively or additionally, the method 600 may further include, in response to an update at the data source resulting in an alteration of a portion of the at least some data of the data source, updating the visualization to reflect the alteration. For example, if the visualization is a logical network diagram that includes, among other devices, components, and/or connections, a server that is renamed at the data source such that the server has a new name,
20 the visualization may be updated to include the server’s new name. As another example, if the visualization is an organizational chart that includes various employees of an organization, and one of the employees is, e.g., deleted from the data source or updated at the data source to have a status of former employee in response to the employee leaving the organization, the organizational chart may be updated to omit the former employee from the organizational chart.

25 In at least one embodiment, the method 600 may further include, prior to displaying the visualization, generating a semantic layout of the visualization based on the contract associated with the selected visualization type. The semantic layout may specify one or more of relative locations, arrangement, and relationships of multiple data objects to represent in the visualization.

30 In at least one embodiment, the selected visualization type may include a first selected visualization type, the visualization may include a first visualization, the contract may include a first contract, and the at least some data of the data source may include a first portion of the data of the data source. In this and other embodiments, the method 600 may further include receiving input effective to select a second visualization type of the multiple different

visualization types to apply to a second visualization of the data from the data source. The second selected visualization type may be associated with a second contract of the multiple contracts, where the second contract is different than the first contract. The method 600 may also include mapping a second portion of the data of the data source to the second contract such that the second contract is satisfied. And displaying the second visualization of the second portion of the data of the data source. The displayed second visualization may have the second selected visualization type that is different than the first selected visualization type.

In at least one embodiment, the selected visualization type may include a first selected visualization type and the visualization may include a first visualization. In this and other embodiments, the method 600 may further include receiving input effective to select a second visualization type of the multiple different visualization types to apply to a second visualization of the data from the data source. The second selected visualization type may be associated with the same contract as the first selected visualization type. The method 600 may also include displaying the second visualization of the at least some data of the data source. The displayed second visualization may have the second selected visualization type that is different than the first selected visualization type.

Alternatively or additionally, the method 600 may further include, prior to the mapping at block 604, translating the data of the data source from a first format to a standardized format. Translating the data to the standardized format may include translating the data to normalized tabular data.

Some or all of the functions or blocks described in this and other processes and methods may be implemented by one or more processors executing computer-readable instructions stored on one or more non-transitory computer-readable media. Execution by the processor(s) of the computer-readable instructions may cause the processor(s) to perform or control performance of the functions or blocks. The one or more processors may include the processor 516 of the server 502 of Figure 5, the processor 526 of the client device 504, and/or one or more other processors of one or more other devices. The non-transitory computer-readable media may include one or more non-transitory computer-readable media included in or accessible to the foregoing computer or server, such as the storage medium 518 of the server 502, the storage medium 528 of the client device 504, and/or one or more other storage media of one or more other devices. The computer-readable instructions may include one or more of the visualization application 514 of the server 502, the browser 524 of the client device 504, and/or one or more other applications of one or more other devices.

Figure 7 is a flowchart of a method 700 to customize a visualization of data from a data source based on a contract, arranged in accordance with at least one embodiment described herein. The method 700 may be programmably performed or controlled by a processor in, e.g., a computer and/or server that executes a browser, visualization application, and/or other application to customize visualizations. In an example implementation, the method 700 may be performed in whole or in part by the server 502 and/or the client device 504 of Figure 5. The method 700 may include one or more of blocks 702, 704, 706, and/or 708. The method 700 may begin at block 702.

At block 702 (“Generate A Semantic Layout Of A Visualization Based On A Contract Associated With A Visualization Type, The Semantic Layout Specifying One Or More Of Relative Locations, Arrangement, And Relationships Of Multiple Data Objects To Represent In The Visualization, The Data Objects Including At Least Some Data Of A Data Source”), a semantic layout of a visualization may be generated based on a contract. The contract may be associated with a visualization type. The semantic layout may specify one or more of relative locations, arrangement, and relationships of multiple data objects to represent in the visualization. The data objects may include at least some data of a data source. Block 702 may be followed by block 704.

At block 704 (“Receive Input Effective To Alter A Set Of One Or More Graphical Objects To Display For Each Of Some But Not All Of The Data Objects From A Default Set To A Custom Set”), input may be received that is effective to alter a set of one or more graphical objects to display for each of some but not all of the data objects from a default set of one or more graphical objects to a custom set of one or more graphical objects. The default set is different than the custom set. Receiving the input may include receiving input from a user effective to specify the customization. Block 704 may be followed by block 706.

At block 706 (“Generate A Concrete Layout Of The Visualization That Specifies The Set Of One Or More Graphical Objects To Display For Each Of The Data Objects, Wherein The Concrete Layout Specifies The Default Set For A First Data Object And The Custom Set For A Second Data Object”), a concrete layout of the visualization may be generated. The concrete layout may specify the set of one or more graphical objects to display for each of the plurality of data objects. In particular, the concrete layout may specify the default set for a first data object of the multiple data objects and the custom set for a second data object of the multiple data objects. Block 706 may be followed by block 708.

At block 708 (“Display The Visualization Of The At Least Some Data Of The Data Source Based On The Contract, The Semantic Layout, And The Concrete Layout, The

Visualization Having The Visualization Type”), the visualization of the at least some data of the data source may be displayed based on the contract, the semantic layout, and the concrete layout. The visualization may have the visualization type associated with the contract.

By way of example, receiving the input at block 704 may include receiving a selection
5 by a user of one of the graphical objects displayed for a data object and receiving an alteration by the user of some aspect of the graphical object. Alternatively or additionally, receiving the input at block 704 may include receiving data object-specific formatting input that specifies the custom set to display for the second data object. For example, the user may input data object-specific formatting by altering one or more of the graphical objects displayed in the
10 visualization for the second data object. In these and other embodiments, displaying the visualization may include displaying the custom set for the second data object and displaying the default set for the first data object. Alternatively or additionally, the method 700 may further include, in response to an update at the data source that alters the second data object, updating the visualization to reflect the alteration while maintaining the custom set for the second data
15 object.

As another example, receiving the input at block 704 may include receiving a conditional customization rule created by the user that customizes one or more graphical objects displayed for any of the data objects that satisfy a predetermined condition specified in the conditional customization rule. In more detail, receiving the input at block 704 may include
20 receiving conditional customization input (e.g., a conditional customization rule created by a user) that specifies a predetermined custom set of one or more graphical objects to display for each data object of a subset of the multiple data objects, where each data object of the subset satisfies a predetermined condition specified by the conditional customization input. The second data object may be included in the subset that satisfies the predetermined condition.

In this and other embodiments, displaying the visualization at block 708 may include
25 displaying the predetermined custom set for each data object—including the second data object—of the subset that each satisfies the predetermined condition. Displaying the visualization may also include displaying the default set for each data object of another subset of the multiple data objects, where the other subset includes the first data object and each data
30 object of the other subset fails to satisfy the predetermined condition. As a particular example, the multiple data objects may include employees of an organization and the predetermined condition may include having a job at or above a predetermined level of the organization. In this and other embodiments, displaying the visualization at block 708 may include displaying a hierarchical organizational chart (e.g., a visualization with a hierarchical organizational chart

visualization type). Displaying the hierarchical organizational chart may include displaying the default set for each of the employees below the predetermined level of the organization and displaying the predetermined custom set for each of the employees at or above the predetermined level of the organization.

5 In at least one embodiment, the method 700 may further include changing the graphical objects displayed for a data object that initially satisfies the predetermined condition from the predetermined custom set back to the default set in response to the data object being changed at the data source such that it no longer satisfies the predetermined condition. For example, in response to an update at the data source that alters the second data object to no longer satisfy
10 the predetermined condition, the method 700 may further include updating the visualization to reflect the alteration. Alternatively or additionally, the semantic layout and/or the concrete layout may be updated prior to updating the visualization. Updating the visualization may include changing a display of a set of one or more graphical objects for the second data object from the predetermined custom set to the default set.

15 Alternatively or additionally, the method 700 may be combined, in whole or in part, with one or more of the other methods described herein, or with one or more portions of one or more of the other methods described herein. For example, the method 700 may further include, prior to displaying the visualization, receiving input effective to select the visualization type of multiple different visualization types to apply to the visualization and mapping at least
20 some data of the data source to the contract such that the contract is satisfied. As another example, the method 700 may further include receiving input effective to alter a third graphical object of the multiple graphical objects, the alteration implicating a change to a third data object of the at least some data of the data source; and altering the third data object at the data source consistent with the alteration to the third graphical object.

25 Figure 8 is a flowchart of a method 800 to interact with a visualization of data from a data source based on a contract, arranged in accordance with at least one embodiment described herein. The method 800 may be programmably performed or controlled by a processor in, e.g., a computer and/or server that executes a browser, visualization application, and/or other application to customize visualizations. In an example implementation, the method 800 may
30 be performed in whole or in part by the server 502 and/or the client device 504 of Figure 5. The method 800 may include one or more of blocks 802, 804, 806, and/or 808. The method 800 may begin at block 802.

At block 802 (“Map At Least Some Data Of A Data Source To A Contract Such That The Contract Is Satisfied”), at least some data of a data source may be mapped to a contract

such that the contract is satisfied. The data source may include, e.g., any of the data sources 102, 104, 106, 510 or other data sources described herein. The contract may include, e.g., any of the contracts 120, 122, 124, 300, 402 or other contracts described herein. The mapping may be performed by or using a corresponding contract map, such as, e.g., any of the contract maps 304, 306, 404 or other contract maps described herein. In at least one embodiment, at least some data of the data source may not have at the data source a schema or shape that matches the corresponding schema or shape of the contract until the at least some data of the data source is mapped to the contract. Mapping at least some data of the data source to the contract at block 804 may include identifying multiple specific data from multiple different data collections extracted from the data source. When combined, the multiple specific data from the multiple different data collections may form a set of data having a structure necessary to produce a visualization. Block 802 may be followed by block 804.

At block 804 (“Display A Visualization Of The At Least Some Data Of The Data Source Based On The Contract, The Displayed Visualization Including Multiple Graphical Objects That Graphically Represent The At Least Some Data Of The Data Source”), a visualization of the at least some data of the data source may be displayed based on the contract. The displayed visualization may include multiple graphical objects that graphically represent the at least some data of the data source. Block 804 may be followed by block 806.

At block 806 (“Receive Input Effective To Alter A First Graphical Object Of The Multiple Graphical Objects”), input effective to alter a first graphical object of the multiple graphical objects may be received. Block 806 may be followed by block 808.

At block 808 (“Determine That The Alteration To The First Graphical Object Implicates A Change To A First Data Object Of The At Least Some Data Of The Data Source”), it may be determined that the alteration to the first graphical object implicates a change to a first data object of the at least some data of the data source.

By way of example of the method 800, the visualization may include an organizational chart of an organization. The first graphical object may include a block in the organizational chart. The block may identify a first individual in the organization as the first data object. Receiving input effective to alter the first graphical object at block 806 may include receiving input effective to alter at least one of: a second graphical object above the first graphical object in the organizational chart to which the first graphical object is connected, the second graphical object representing a second individual in the organization; and a third graphical object below the first graphical object in the organizational chart to which the first graphical object is connected, the third graphical object representing a third individual in the organization.

Determining that the alteration implicates the change to the first data object may include determining that the alteration implicates at least one of: a change of an individual to whom the first individual reports to the second individual; and a change of an individual that reports to the first individual to the third individual.

5 Accordingly, some embodiments described herein determine and/or understand implicated changes to underlying source data based on changes to graphical objects that represent the source data. In these and other embodiments, the implicated changes may be automatically made to the underlying source data or the implicated changes may be presented to the user (e.g., in a window on a display or in some other user interface) to manually make
10 the implicated changes to the underlying source data. Alternatively or additionally, the alterations to the graphical objects and/or the implicated changes to the underlying data may be saved apart from the underlying source data, e.g., in a visualization file such as the visualization file 400 of Figure 4, such that the implicated changes may potentially be made later to the underlying source data, if desired. While the alterations and/or implicated changes
15 may be saved apart from the underlying source data for a given visualization, the visualization may still be updated to reflect changes to the underlying source data alongside the alterations to the graphical objects.

 In at least one embodiment, the method 800 may further include, prior to altering the first data object at the data source, mapping the first data object from the contract back to the
20 data source to determine where the first data object is located at the data source.

 Alternatively or additionally, prior to altering the first data object at the data source, the method 800 may further include: determining that the alteration implicates the change to the first data object; presenting a query that identifies the implicated change to the first data object and queries whether to make the implicated change to the first data object at the data source;
25 and receiving an affirmative response to the query, where altering the first data object at the data source consistent with the alteration to the first graphical object comprises implementing the implicated change at the data source in response to receiving the affirmative response.

 Alternatively or additionally, prior to altering the first data object at the data source, the method 800 may further include: receiving input effective to alter multiple graphical objects,
30 the multiple graphical objects including the first graphical object and a second graphical object; determining that the alterations implicate changes to multiple data objects of the at least some data, the multiple data objects including the first data object and a second data object; and locally storing the alterations. For example, the alterations and/or implicated changes may be

stored on or by the client device 504 and/or on or by the server 502 and locally accessible to one or both of the client device 504 and the server 502.

In at least one embodiment, the method 800 may further include presenting the implicated changes to a user. The method 800 may further include receiving first user input effective to affirm a first one of the implicated changes that applies to the first data object. In this and other embodiments, altering the first data object at the data source consistent with the alteration to the first graphical object may include implementing the first one of the implicated changes at the data source in response to receiving the first user input. The method 800 may further include receiving second user input effective to reject a second one of the implicated changes that applies to the second data object. The second data object at the data source may remain unchanged in response to receiving the second user input rejecting the second one of the implicated changes.

In at least one embodiment, the method 600 may further include, in response to altering the first data object at the data source, refreshing the visualization to display the visualization based on updated data at the data source that includes the altered first data object.

Alternatively or additionally, the method 800 may further include providing a set of one or more interactions with the visualization that allow configuring the visualization without editing the at least some data of the data source. For example, drop-down arrows or other user interface elements may be provided to selectively display or hide one or more portions of a visualization. Accordingly, in at least one embodiment, providing the set of one or more interactions with the visualization that allow configuring the visualization without editing the at least some data of the data source may include providing one or more user interface elements that allow a user to select whether one or more portions of the visualization are visible in the visualization.

Alternatively or additionally, the method 800 may be combined, in whole or in part, with one or more of the other methods described herein, or with one or more portions of one or more of the other methods described herein. For example, the method 800 may further include some or all of the method 600 of Figure 6 and/or the method 700 of Figure 7.

Some example implementations of at least some portions of each of the methods 600, 700, 800 of Figures 6-8 will be discussed together with Figures 9A-9F. Figures 9A-9F include graphical representations 900A-900F (collectively “graphical representations 900”) of a user interface to generate, customize, and/or interact with visualizations, arranged in accordance with at least one embodiment described herein. The user interface may be provided by the browser 524 and/or the visualization application 514 of Figure 5. For instance, the browser 524

in cooperation with the visualization application 514 may present the user interface to a user through a display of the client device 504. The graphical representations 900 may include point-in-time representations of the user interface corresponding to different steps of one or more of the methods 600, 700, 800 of Figures 6-8.

5 In Figures 9A-9F, the graphical representations 900 include a graphical drawing canvas 902 and a panel 904 beside the graphical drawing canvas 902. The panel 904 may include a field 906 that identifies the name of an organization that owns and/or is otherwise associated with one or more data sources from which visualizations may be generated.

The panel 904 includes a list 908 of various visualization types that may applied to
10 visualizations of the data source. In the example of Figures 9A-9F, the visualization types include hierarchical organizational chart (“Org Chart – Hierarchical”), matrix organizational chart (“Org Chart – Matrix”), flat organizational chart (“Org Chart – Flat”), container (“Container”), mind map (“Mind Map”), and abstraction layer (“Abstraction Layer”). The ellipses in the list 908 indicate that more generally the list 908 may include more, fewer, and/or
15 different visualization types than illustrated in Figures 9A-9F.

In at least one embodiment, the panel 904 may additionally include one or more diagram shapes 910. The diagram shapes 910 may be dragged and dropped into the diagram canvas 902 by the user to manually generate at least a portion of a diagram.

With combined reference to Figures 6-9F, various example implementations of one or
20 more steps of the methods 600, 700, 800 will be described. For example, block 602 of the method 600 of Figure 6 may include receiving input effective to select a “Container” visualization type 912 (Figure 9B) from the list 908. In more detail, a user may move a mouse pointer to the “Container” visualization type 912 in the list 908 and may select the “Container” visualization type 912, e.g., by double-clicking on the “Container” visualization type 912 in
25 the list or providing other suitable input. In response, the selected visualization type may then be highlighted, e.g., by showing a box 914 around the selected visualization type or providing some other visible feedback or other feedback confirming the selection by the user of the “Container” visualization type 912.

The method 600 may also include, at block 606, displaying a visualization 916 (Figure
30 9B) that includes the selected “Container” visualization type 912. Thus, the visualization 916 may be referred to as a container visualization. In the example of Figure 9B, the container visualization 916 includes two containers 918, 920 that represent two different teams of an organization. The container 918 is labeled “Team 1” and includes various individuals (e.g., Individuals 1-6) of the organization that are part of and/or have been assigned to the team

“Team 1”. Analogously, the container 920 is labeled “Team 2” and includes various individuals (e.g., Individuals 7-12) of the organization that are part of and/or have been assigned to the team “Team 2”.

In at least one embodiment, a user may desire to modify the team assignments, e.g., in the container visualization 916, rather than at the data source form which the container visualization 916 was generated, e.g., according to the method 800 of Figure 8. In this example, the text “Individual 1” and/or the box around the text “Individual 1” may each be a graphical object that represents an employee or other entity of the organization as a data object included in the source data of the data source, which may be referred to as the “Individual 1” data object for simplicity. The text and/or boxes for the other individuals in Figure 9B may similarly be graphical objects that represent other employees or entities of the organization as data objects, with a similar naming convention as used for the “Individual 1” data object.

In this and other embodiments, the method 800 of Figure 8 may include, at block 806, receiving input effective to alter the text and box for the “Individual 8” data object, e.g., by moving the text and box for the “Individual 8” data object from the container 920 to the container 918, as denoted at 922 in Figure 9C. For example, a user may use a mouse (or other input device) to click and drag the text and box for the “Individual 8” data object from the container 920 to the container 918. Such a change may implicate a change to the “Individual 8” data object in the source data, which may be determined at block 808 in the method 800. In particular, moving the text and box for “Individual 8” from the container 920 to the container 908 may indicate that the user desires to change the team assignment for “Individual 8”. The implicated change may be easily understood and/or inferred by, e.g., the visualization application 514 since semantically the visualization application may use the positioning of the text/boxes for the various data objects to represent belonging/assignment of the various data objects to a specific team associated with each container 918, 920, such that repositioning one of the text/boxes for one of the data objects from one of the containers into a different container 918, 920 indicates the data object is being reassigned to a different team.

If it is determined that the change to the graphical object (e.g., the text/box for the “Individual 8” data object) implicates the change to the data object (e.g., the “Individual 8” data object), a query 924 may be presented to the user that identifies the implicated change and queries whether to make the implicated change to the data object at the data source. The query 924 may be presented each time a graphical object is changed in a way that implicates a change to a data object represented by the graphical object. Alternatively or additionally, a list of changes may be stored locally, e.g., in the differential log 408 of Figure 4, and all of the accrued

changes may be presented to the user together and/or in sequence periodically or in response to particular events such as the user choosing to close the visualization.

In response to a query such as the query 924 of Figure 9C, the user may affirm that the implicated change should be implemented at the data source, e.g., by selecting “Yes” in the query 924. If the implicated change is affirmed, the implicated change may be implemented at the data source. Alternatively, the user may reject the implicated change, e.g., by selecting “No” in the query 924.

In Figure 9D, an “Org Chart – Hierarchical” visualization type has been selected by a user, as denoted at 926, to generate a visualization 928 with a hierarchical organizational chart visualization type. The visualization 928 may be referred to hereinafter as an org chart visualization. The org chart visualization 928 may be generated for at least some of the same data as the container visualization 916 of Figures 9B and 9C.

In Figure 9D, a user may provide customization input which may be received at, e.g., block 704 of the method 700 of Figure 7, to alter a set of one or more graphical objects to display for one or more data objects from a default set to a custom set. For example, the user may provide data object-specific customization input by selecting the text and/or box for the “Individual 10” data object and applying italics to the text for the “Individual 10” data object, as illustrated in Figure 9D at 930. Alternatively or additionally, conditional customization input may be received from the user that specifies that all individuals in the organization that satisfy a predetermined condition of being upper management, e.g., of being in the first and second levels of the hierarchical organization in this example, be shown with a surrounding box with a heavy line weight (e.g., a heavier line weight than the default line weight shown for all other individuals in Figure 9D). It can be seen in Figure 9D that “Individual 13”, “Individual 1”, and “Individual 7” (as well as another individual that is partially cut off from view) satisfy the predetermined condition, as indicated by being represented by a box graphical object that has a heavier line weight than the default line weight.

Now suppose the underlying data upon which the org chart visualization 928 is generated is updated at the data source. For example, suppose “Individual 7” leaves the company and “Individual 10” is promoted to replace “Individual 7”. As illustrated in Figure 9E, the org chart visualization 928 is updated to reflect the changes to the underlying data. In particular, “Individual 7” is no longer represented in the org chart visualization 928 of Figure 9E, and the text and box graphical objects for the “Individual 10” data object are now depicted in Figure 9E in the location previously occupied by the graphical objects representing the “Individual 7” data object. Moreover, as illustrated in Figure 9E, since the “Individual 10” data

object now satisfies the predetermined condition of being upper management, the box graphical object for the “Individual 10” data object is displayed with the heavier line weight than the default line weight. On the other hand, if “Individual 7” was still part of the organization but was demoted from upper management or otherwise no longer satisfied the predetermined condition of the conditional customization input, the text graphical object and box graphical object that represent the “Individual 7” data object would still be displayed in the org chart visualization 928 but without the heavier line weight.

Further, the data object-specific customization input of italics text that was previously displayed for the “Individual 10” data object is still displayed in Figure 9E. Thus, data object-specific customization input may stay with a data object—or more particularly, with the graphical object(s) that represent the data object—even if the graphical object(s) is(are) moved around in the org chart visualization 928 in response to some change at the data source and/or in response to being moved manually by the user.

Alternatively or additionally, Figure 9F illustrates various interactions that may be provided for the org chart visualization 928 without substantively editing the underlying source data and/or the org chart visualization 928 itself. In particular, one or more areas of the org chart visualization 928 may be selectively collapsible and/or expandable. In this regard, Figure 9F illustrates user interface elements 932 that may be selected, e.g., by a mouse click or other user input, to expand a portion of the org chart visualization 928, as well as user interface elements 934 that may be selected, e.g., by a mouse click or other user input, to collapse a portion of the org chart visualization 928.

Figure 10 is a block diagram illustrating an example computing device 1000 that is arranged for generating, customizing, and/or interacting with visualizations, arranged in accordance with at least one embodiment described herein. In a basic configuration 1002, the computing device 1000 typically includes one or more processors 1004 and a system memory 1006. A memory bus 1008 may be used to communicate between the processor 1004 and the system memory 1006.

Depending on the desired configuration, the processor 1004 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 1004 may include one or more levels of caching, such as a level one cache 1010 and a level two cache 1012, a processor core 1014, and registers 1016. The processor core 1014 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 1018 may also be used with the processor 1004, or in

some implementations the memory controller 1018 may include an internal part of the processor 1004.

Depending on the desired configuration, the system memory 1006 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 1006 may include an operating system 1020, one or more applications 1022, and program data 1024. The application 1022 may include a visualization application 1026 that is arranged to generate, customize, and/or interact with visualizations, as described herein. The program data 1024 may include visualization files 1028 (which may include or correspond to the visualization file 400 of Figure 4) as is described herein, or other visualization application data. In some embodiments, the application 1022 may be arranged to operate with the program data 1024 on the operating system 1020 such that one or more of the methods 600, 700, 800 of Figures 6-8 may be provided as described herein.

The computing device 1000 may have additional features or functionality, and additional interfaces to facilitate communications between the basic configuration 1002 and any involved devices and interfaces. For example, a bus/interface controller 1030 may be used to facilitate communications between the basic configuration 1002 and one or more data storage devices 1032 via a storage interface bus 1034. The data storage devices 1032 may be removable storage devices 1036, non-removable storage devices 1038, 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 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 1006, the removable storage devices 1036, and the non-removable storage devices 1038 are examples of computer storage media or non-transitory 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 by the computing device 1000. Any such computer storage media or non-transitory computer-readable media may be part of the computing device 1000.

The computing device 1000 may also include an interface bus 1040 to facilitate communication from various interface devices (e.g., output devices 1042, peripheral interfaces 1044, and communication devices 1046) to the basic configuration 1002 via the bus/interface controller 1030. The output devices 1042 include a graphics processing unit 1048 and an audio processing unit 1050, which may be configured to communicate to various external devices such as a display or speakers via one or more A/V ports 1052. Diagrams, flowcharts, organizational charts, connectors, and/or other graphical objects generated by the diagram application 1026 may be output through the graphics processing unit 1048 to such a display. The peripheral interfaces 1044 include a serial interface controller 1054 or a parallel interface controller 1056, 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 1058. Such input devices may be operated by a user to provide input to the diagram application 1026, 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 1026. The communication devices 1046 include a network controller 1060, which may be arranged to facilitate communications with one or more other computing devices 1062 over a network communication link via one or more communication ports 1064.

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 1000 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 1000 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, and not limitation, such computer-readable media may include non-transitory computer-readable storage media including Random Access Memory (RAM), Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), Compact Disc Read-Only Memory (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.

One skilled in the art will appreciate that, for this and other processes and methods disclosed herein, the functions performed in the processes and methods may be implemented in differing order. Further, the outlined steps and operations are only provided as examples, and some of the steps and operations may be optional, combined into fewer steps and operations, or expanded into additional steps and operations without detracting from the essence of the disclosed embodiments.

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 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.

As used herein, the terms “module” or “component” may refer to specific hardware implementations configured to perform the operations of the module or component and/or software objects or software routines that may be stored on and/or executed by general-purpose hardware (e.g., computer-readable media, processing devices, etc.) of the computing system. In some embodiments, the different components, modules, engines, and services described

herein may be implemented as objects or processes that execute on the computing system (e.g., as separate threads). While some of the system and methods described herein are generally described as being implemented in software (stored on and/or executed by general-purpose hardware), specific hardware implementations or a combination of software and specific hardware implementations are also possible and contemplated. In this description, a
5 “computing entity” may be any computing system as previously defined herein, or any module or combination of modules running on a computing system.

A phrase such as “an aspect” does not imply that such aspect is essential to the subject technology or that such aspect applies to all configurations of the subject technology. A
10 disclosure relating to an aspect may apply to all configurations, or one or more configurations. An aspect may provide one or more examples of the disclosure. A phrase such as “an aspect” may refer to one or more aspects and vice versa. A phrase such as “an embodiment” does not imply that such embodiment is essential to the subject technology or that such embodiment applies to all configurations of the subject technology. A disclosure relating to an embodiment
15 may apply to all embodiments, or one or more embodiments. An embodiment may provide one or more examples of the disclosure. A phrase such as “an embodiment” may refer to one or more embodiments and vice versa. A phrase such as “a configuration” does not imply that such configuration is essential to the subject technology or that such configuration applies to all configurations of the subject technology. A disclosure relating to a configuration may apply to
20 all configurations, or one or more configurations. A configuration may provide one or more examples of the disclosure. A phrase such as “a configuration” may refer to one or more configurations and vice versa.

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.” Pronouns in the masculine (e.g., his)
25 include the feminine and neuter gender (e.g., her and its) and vice versa. The term “some” refers to one or more. Moreover, nothing disclosed herein is intended to be dedicated to the public regardless of whether such disclosure is explicitly recited in the above description.

The present disclosure is not to be limited in terms of the particular embodiments described herein, which are intended as illustrations of various aspects. Many modifications and variations can be made without departing from its spirit and scope. Functionally equivalent
30 methods and apparatuses within the scope of the disclosure, in addition to those enumerated herein, are possible from the foregoing descriptions. Such modifications and variations are intended to fall within the scope of this disclosure. Also, the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to be limiting.

With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

5 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 is not limited to,” etc.). It will be further understood by those within the art that if a specific number of an introduced claim recitation is intended, such an intent will be explicitly recited in the claim, and in the
10 absence of such recitation, no such intent is present. For example, as an aid to understanding, the following appended claims may contain usage of the introductory phrases “at least one” and “one or more” to introduce claim recitations. However, the use of such phrases should not be construed to imply that the introduction of a claim recitation by the indefinite articles “a” or
15 “an” limits any particular claim containing such introduced claim recitation to embodiments containing only one such recitation, even when the same claim includes the introductory phrases “one or more” or “at least one” and indefinite articles such as “a” or “an” (e.g., “a” and/or “an” should be interpreted to mean “at least one” or “one or more”); the same holds true for the use of definite articles used to introduce claim recitations. In addition, even if a specific
20 number of an introduced claim recitation is explicitly recited, those skilled in the art will recognize that such recitation should be interpreted to mean at least the recited number (e.g., the bare recitation of “two recitations,” without other modifiers, means at least two recitations, or two or more recitations). 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
25 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, and/or A, B, and C together, etc.). In those instances where a convention analogous to “at least one of A, B, or C, etc.” is used, in general, such a construction is intended in the sense one having skill in the art
30 would understand the convention (e.g., “a system having at least one of A, B, or 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, and/or A, B, and C together, etc.). It will be further understood by those within the art that virtually any disjunctive word and/or phrase presenting two or more alternative terms, whether in the description, claims, or drawings, should be

understood to contemplate the possibilities of including 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.”

In addition, where features or aspects of the disclosure are described in terms of
5 Markush groups, those skilled in the art will recognize that the disclosure is also thereby described in terms of any individual member or subgroup of members of the Markush group.

For any and all purposes, such as in terms of providing a written description, all ranges disclosed herein also encompass any and all possible sub ranges and combinations of sub ranges thereof. Any listed range can be easily recognized as sufficiently describing and
10 enabling the same range being broken down into at least equal halves, thirds, quarters, fifths, tenths, and/or others. As a non-limiting example, each range discussed herein can be readily broken down into a lower third, middle third and upper third, etc. All language such as “up to,” “at least,” and the like include the number recited and refer to ranges which can be subsequently broken down into sub ranges as discussed above. Finally, a range includes each individual
15 member. Thus, for example, a group having 1-3 cells refers to groups having 1, 2, or 3 cells. Similarly, a group having 1-5 cells refers to groups having 1, 2, 3, 4, or 5 cells, and so forth.

From the foregoing, various embodiments of the present disclosure have been described herein for purposes of illustration, and various modifications may be made without departing from the scope and spirit of the present disclosure. Accordingly, the various embodiments
20 disclosed herein are not intended to be limiting.

CLAIMS

What is claimed is:

1. A method to customize a visualization, the method comprising:

receiving first input effective to select a visualization type to apply to a visualization of data of a data source;

generating a semantic layout of the visualization based on a contract associated with the visualization type, the semantic layout specifying one or more of relative locations, arrangement, and relationships of a plurality of data objects to represent in the visualization, the plurality of data objects including data of the data source, the contract including a schema that specifies a set of data fields and one or more relationships between one or more data fields of the set of data fields to generate and display the visualization;

receiving second input effective to alter a set of one or more graphical objects to display for each data object of a subset of the plurality of data objects from a default set of one or more graphical objects to a custom set of one or more graphical objects, wherein the default set is different than the custom set and wherein the second input comprises conditional customization input or data object-specific customization input;

generating a concrete layout of the visualization that specifies the set of one or more graphical objects to display for each of the plurality of data objects, wherein the concrete layout specifies the default set for a first data object of the plurality of data objects that is not included in the subset and the custom set for a second data object of the plurality of data objects that is included in the subset; and

displaying the visualization of the data of the data source based on the contract, the semantic layout, and the concrete layout, the visualization having the visualization type.

2. The method of claim 1, wherein receiving the second input effective to alter the set of one or more graphical objects to display for each data object of the subset from the default set to the custom set comprises receiving the conditional customization input that specifies a

predetermined custom set of one or more graphical objects to display for each data object of the subset, each data object of the subset satisfying a predetermined condition specified by the conditional customization input.

3. The method of claim 2, wherein the plurality of data objects includes a plurality of employees of an organization, wherein the predetermined condition comprises having a job at or above a predetermined level of the organization, and wherein displaying the visualization comprises displaying a hierarchical organizational chart, including:

displaying the default set for each of the employees below the predetermined level of the organization; and

displaying the predetermined custom set for each of the employees at or above the predetermined level of the organization.

4. The method of claim 2, wherein displaying the visualization comprises:

displaying the default set for each data object of an other subset of the plurality of data objects, the other subset including the first data object, wherein each data object of the other subset of the plurality of data objects fails to satisfy the predetermined condition; and

displaying the predetermined custom set for each data object, including the second data object, of the subset that satisfies the predetermined condition.

5. The method of claim 4, further comprising, in response to an update at the data source that alters the second data object to no longer satisfy the predetermined condition, updating the visualization to reflect the alteration, including changing a display of a set of one or more graphical objects for the second data object from the predetermined custom set to the default set.

6. The method of claim 1, wherein receiving the second input effective to alter the set of one or more graphical objects to display for each data object of the subset from the default set to the custom set comprises receiving the data object-specific formatting input that specifies the custom set to display for the second data object.

7. The method of claim 6, wherein displaying the visualization comprises:

displaying the custom set for the second data object; and

displaying the default set for the first data object.

8. The method of claim 7, further comprising, in response to an update at the data source that alters the second data object, updating the visualization to reflect the alteration while maintaining the custom set for the second data object.

9. The method of claim 1, further comprising, prior to displaying the visualization mapping data of the data source to the contract such that the contract is satisfied.

10. The method of claim 1, further comprising:

receiving third input effective to alter a third graphical object of the plurality of graphical objects, the alteration implicating a change to a third data object of the data of the data source; and

altering the third data object at the data source consistent with the alteration to the third graphical object.

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

receiving first input effective to select a visualization type to apply to a visualization of data of a data source;

generating a semantic layout of the visualization based on a contract associated with the visualization type, the semantic layout specifying one or more of relative locations, arrangement, and relationships of a plurality of data objects to represent in the visualization, the plurality of data objects including data of the data source, the contract including a schema that specifies a set of data fields and one or more relationships between one or more data fields of the set of data fields to generate and display the visualization;

receiving second input effective to alter a set of one or more graphical objects to display for each data object of a subset of the plurality of data objects from a default set of one or more graphical objects to a custom set of one or more graphical objects, wherein the default set is

different than the custom set and wherein the second input comprises conditional customization input or data object-specific customization input;

generating a concrete layout of the visualization that specifies the set of one or more graphical objects to display for each of the plurality of data objects, wherein the concrete layout specifies the default set for a first data object of the plurality of data objects that is not included in the subset and the custom set for a second data object of the plurality of data objects that is included in the subset; and

displaying the visualization of the data of the data source based on the contract, the semantic layout, and the concrete layout, the visualization having the visualization type.

12. The non-transitory computer-readable storage medium of claim 11, wherein receiving the second input effective to alter the set of one or more graphical objects to display for each data object of the subset from the default set to the custom set comprises receiving the conditional customization input that specifies a predetermined custom set of one or more graphical objects to display for each data object of the subset, each data object of the subset satisfying a predetermined condition specified by the conditional customization input.

13. The non-transitory computer-readable storage medium of claim 12, wherein the plurality of data objects includes a plurality of employees of an organization, wherein the predetermined condition comprises having a job at or above a predetermined level of the organization, and wherein displaying the visualization comprises displaying a hierarchical organizational chart, including:

displaying the default set for each of the employees below the predetermined level of the organization; and

displaying the predetermined custom set for each of the employees at or above the predetermined level of the organization.

14. The non-transitory computer-readable storage medium of claim 12, wherein displaying the visualization comprises:

displaying the default set for each data object of an other subset of the plurality of data objects, the other subset including the first data object, wherein each data object of the other subset of the plurality of data objects fails to satisfy the predetermined condition; and

displaying the predetermined custom set for each data object, including the second data object, of the subset that satisfies the predetermined condition.

15. The non-transitory computer-readable storage medium of claim 14, the operations further comprising, in response to an update at the data source that alters the second data object to no longer satisfy the predetermined condition, updating the visualization to reflect the alteration, including changing a display of a set of one or more graphical objects for the second data object from the predetermined custom set to the default set.

16. The non-transitory computer-readable storage medium of claim 11, wherein receiving the second input effective to alter the set of one or more graphical objects to display for each data object of the subset from the default set to the custom set comprises receiving the data object-specific formatting input that specifies the custom set to display for the second data object.

17. The non-transitory computer-readable storage medium of claim 16, wherein displaying the visualization comprises:

displaying the custom set for the second data object; and

displaying the default set for the first data object.

18. The non-transitory computer-readable storage medium of claim 17, the operations further comprising, in response to an update at the data source that alters the second data object, updating the visualization to reflect the alteration while maintaining the custom set for the second data object.

19. The non-transitory computer-readable storage medium of claim 11, the operations further comprising, prior to displaying the visualization

mapping data of the data source to the contract such that the contract is satisfied.

20. The non-transitory computer-readable storage medium of claim 11, the operations further comprising:

receiving third input effective to alter a third graphical object of the plurality of graphical objects, the alteration implicating a change to a third data object of the data of the data source; and

altering the third data object at the data source consistent with the alteration to the third graphical object.

21. The method of claim 1, further comprising, prior to displaying the visualization, translating data of the data source from a first format to a standardized format.