



- (51) **International Patent Classification:**
G06F 17/30 (2006.01) *G06Q 10/00* (2012.01)
- (21) **International Application Number:**
PCT/US2015/066335
- (22) **International Filing Date:**
17 December 2015 (17.12.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
62/093,666 18 December 2014 (18.12.2014) US
14/717,633 20 May 2015 (20.05.2015) US
- (71) **Applicant:** AB INITIO TECHNOLOGY LLC [US/US];
201 Spring Street, Lexington, Massachusetts 02421 (US).
- (72) **Inventors:** RADIVOJEVIC, Dusan; 140 Weyland Circle,
North Andover, Massachusetts 01845 (US). MOSS, Peter;
59 Sweetwater Ave., Bedford, Massachusetts 01730 (US).
- (74) **Agent:** MCDONOUGH, Christina V.; Fish & Richardson
P.C., P.O. Box 1022, Minneapolis, Minnesota 55440-1022
(US).

- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) **Title:** BUILDING REPORTS

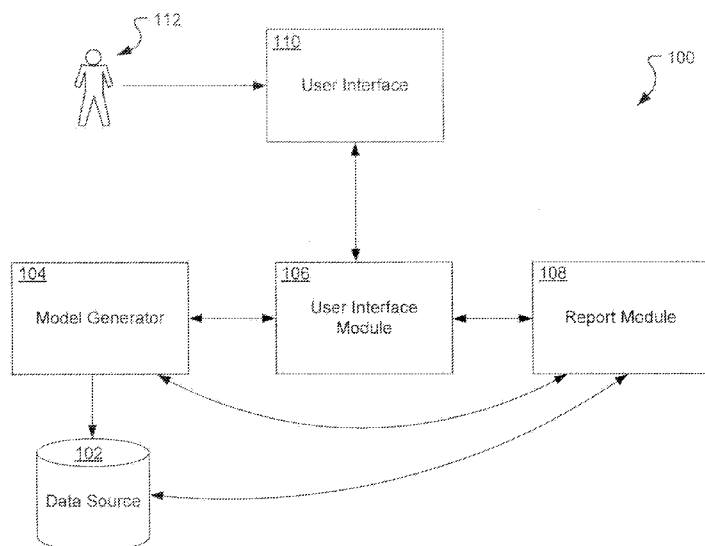


FIG. 1

(57) **Abstract:** Methods, systems, and apparatus, including computer programs encoded on computer storage media, for building reports. One of the methods includes creating a model based on relational structured data, the structured data including data structures, each data structure having data elements, each data element having fields, each field having a name. The method includes generating a hierarchy of objects in model, the hierarchy organizing objects the with respect to a starting object according to relationship fields on the objects. The method includes generating a user interface including elements for one or more of the objects in the hierarchy, wherein the user interface enables a user to create a report and filter the report using the new name. The method includes receiving a user selection of an element from the elements. The method also includes generating a report.

BUILDING REPORTS

BACKGROUND

This description relates to building reports.

5 A database can store large quantities of information. The manner in which the information is stored is not always intuitive. Instead, the structure of the database is determined based on factors such as maintenance of referential integrity, speed of access, and redundancy minimization. Data which may be logically grouped together can be stored in separate entities. Complex relationships may be resolved by adding
10 additional entities which do not have a physical counterpart.

SUMMARY

In general, an aspect 1 of the subject matter described in this specification can be embodied in methods that include the actions of creating a model based on
15 structured data, the structured data including data structures, each data structure having data elements, each data element having fields, each field having a name. Creating the model includes creating an object in the model based on a data structure in the structured data, translating a name of a field in the data structure to a natural language name, creating a field on the object having the new name, and creating a
20 relationship field on the object based on a relationship between data structures in the structured data. The method includes generating a hierarchy of objects in model, the hierarchy organizing objects the with respect to a starting object according to relationship fields on the objects. The aspect includes generating a user interface including elements for one or more of the objects in the hierarchy, wherein the user
25 interface enables a user to create a report and filter the report using the new name. The method includes receiving a user selection of an element from the elements. The method includes generating a report including identifying a data structure corresponding to the element, obtaining data from at least one field in the identified data structure in the data source, and presenting the data to the user.

30 An aspect 2 according to aspect 1, wherein translating a name may include translating the name using a look up table.

An aspect 3 according to any of aspects 1 or 2, wherein the relationship field may be a reference field representing a relationship between one data element in the data structure to one second data element in a second data structure.

5 An aspect 4 according to any of aspects 1, 2, or 3, wherein the relationship field may be a collection field representing a relationship between one data element in the data structure to second data elements in a second data structure.

An aspect 5 according to any aspects 1, 2, 3 or 4, wherein the user interface may include aggregation functions that can be used on the collection field.

10 An aspect 6 according to any of aspects 1, 2, 3, 4, or 5, wherein the relationship field may be a parent-child field representing a relationship between one data element in the data structure to a second data element in the data structure.

An aspect 7 according to any of aspects 1, 2, 3, 4, 5, or 6, wherein the structured data may be a structured database and the data structure is a table in the structured database.

15 An aspect 8 according to any of aspects 1, 2, 3, 4, 5, 6, or 7, wherein the structured data may be one or more files.

An aspect 9 according to any of aspects 1, 2, 3, 4, 5, 6, 7, or 8, wherein generating the report may occur in response to receiving the user selection.

20 An aspect 10 according to any of aspects according to any of aspects 1, 2, 3, 4, 5, 6, 7, 8, or 9, wherein the user interface may enable an interactive creation of reports using a what-you-see-is-what-you-get interface and wherein the user interface may provide visual feedback of the report including data values in near-real-time prior to finalizing the report.

25 An aspect 11 according to any of aspects 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10, wherein generating a user interface may further include receiving a selection of the starting object from a user.

An aspect 12 according to any of aspects 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, or 11, wherein the natural language name may include at least one character in a set of restricted characters for the structured data.

30 An aspect 13 according to any of aspects 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, or 12, wherein creating a relationship field on the object based on a relationship between data structures in the structured data may include creating a relationship object to represent a primary key – foreign key relationship in the structured data.

Other embodiments of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the methods. A system of one or more computers can be configured to perform particular actions by virtue of having
5 software, firmware, hardware, or a combination of them installed on the system that in operation causes or cause the system to perform the actions. One or more computer programs can be configured to perform particular actions by virtue of including instructions that, when executed by data processing apparatus, cause the apparatus to perform the actions.

10 Aspects can include one or more of the following advantages. Reports can be updated in real time, allowing a user to immediately see the impact of the report. Reports can be created quickly.

Other features and advantages of the invention will become apparent from the following description, and from the claims.

15 DESCRIPTION OF DRAWINGS

FIG. 1 illustrates an example of an environment for building reports.

FIG. 2 illustrates an example of mapping a relational database into a meta-model.

FIG. 3 illustrates an example of a user interface for building reports.

20 FIG. 4-6 illustrate examples of a user interface for building reports.

FIG. 7 is a flow chart of an exemplary process for building reports.

FIG. 8 is a flow chart of an exemplary process for building reports.

DESCRIPTION

25 A reporting tools are computer programs whose purpose is to take data from a source such as a database, XML stream or a spreadsheet, and use it to produce a document in a format which satisfies a particular human readership. In traditional reporting tools, an expert user sets up a view that provides access to the data for a particular report or set of reports. Generally, views are set up by technical users who
30 understand the structure of the underlying data (for example, the relational model of the database). The views are used and manipulated by a conventional report builder to generate the report. However, because setting up the views are a manual process,

views typically reflect a small portion of the underlying data schemas or may be particular to a specific business problem.

In addition, these views generally do not present a user-friendly representation of the data. They usually present fields exactly as they are represented in the relational
5 model, which may not be intuitive to end users of the reporting tool.

In contrast, the approach presented here involves two novel elements that make the process of creating reports much more intuitive, interactive and efficient for the end user.

First is the concept of creating a meta-model that describes relational data in
10 business-friendly terms. Relationships are modeled in such a way as to create an intuitive hierarchical meta-model.

Second, a report builder application is used to present the meta-model in a highly interactive user interface (UI) to enable the user to create ad hoc and formal reports quickly and easily. The report builder UI provides instant feedback as the user
15 creates the report. Columns can be added to the report via a drag and drop interface, and filters can be applied in simple straightforward manner. These features enable a user to quickly perform ad hoc investigative actions, and to create more formal reports for execution by a broad set of users.

The meta-model can be generated from either the underlying data source
20 directly or from a specification that describes the underlying data. The meta-model identifies objects, fields of objects, relationships between the objects, and a mapping between the objects in the meta-model and the underlying data.

These relationships are symmetrical and can be navigated in either direction. For example, if a relationship enables the identification of orders based on a customer,
25 then a corresponding relationship enables the identification of a customer based on an order.

FIG. 1 illustrates an example of an environment 100 for building reports. A data source 102 can be, for example, a relational database, an object database, a structured data file (such as a SGML or XML file), a hierarchical database, a data
30 warehouse, or any other data storage system. In general, the data source 102 can be any data store in which data records can be identified and whose data records include values associated with individual fields.

Prior to the generation of a report, a model generator 104 can read the data source 102, or a specification that describes the data sources, and generate a meta-

model, as described further below. The meta-model is generated once and saved for later use by the user interface. In general, the meta-model allows the user interface to present a business-friendly form of the data source. For example, the meta-model may reflect the constraints and relationships established in the data source that
5 correspond to business concepts while eliminating or simplifying the constraints and relationships which are established for technical reasons.

Objects in the meta-model can include different kinds of fields. Ordinary fields are simple fields that provide data to a report, such as name, birth date, quantity, and description. Reference fields are fields that point to other objects in the meta-
10 model. A reference field can be derived from foreign keys in the underlying data source. A meta-model can also include collection fields. Collection fields are logical representations of a collection of objects that are associated with another object. The relationship of the object to the objects in the collection is one-to-many. Generally, a reference field is used when the object is associated with at most one of the target
15 objects and a collection field is used when the object can be associated with many of the target objects.

A user interface module 106 reads the meta-model generated by the model generator 104 and generates a user interface 110 for presentation to a user 112. The user interface module 106 can, for example, generate a webpage or application that is
20 presented to the user 112 on a web browser executed on a client device.

The user 112 interacts with the user interface 110 to generate a report. For example, the user interface 110 may provide user selected report information to the user interface model 106. The user interface module 106 may provide the report information to the report module 108. The report module 108 executes the report
25 against the data source 102. In some implementations, the report module 106 may generate the report using the report information and information from the model generator 104 (for example, a meta-model). The output of the report module 108 can be returned to the user interface 110 via the user interface module 116 for presentation to the user 112. This provides the user 112 with near-real-time (for example, with a
30 latency of less than five seconds) results. This workflow enables the user 112 to see the impact of changes being made to the report as the report is being built.

FIG. 2 illustrates an example of generating a meta-model based on a relational database. In this example, the structure of the data source 102 is reflected in the entity relationship model 200. In general, relationships between rows of data in the tables

may be established by including a foreign key in a row of data of one table which references a primary key in a row of data in a second table. Generally, the foreign key is stored on the side of a relationship that allows for multiple rows in the table with the foreign key to be associated with a single row in the table without a foreign key. The database can describe how one table relates to another. Relationships in a database can describe a number of elements in a relationship between two objects, commonly referred to as the cardinality of the relationship. "One-to-one" indicates that one row in table A relates to one row in table B. "One-to-many" indicates that one row in table A relates to many rows in table B. "Many-to-many" indicates that many rows in table A relate to many rows in table B.

In this example, a row in the customer table 202 can be related to many rows in the order table 210 and each row in the order table 210 can be related to only one row in the customer table 202. As described above, this relationship is created by the introduction of a foreign key in the order table that identifies a row in the customer table 202. Similarly, a row in the order table 210 can be related to many rows in the line item table 212, while a row in the line item table 212 can be related to only one row in the order table 210. A row in the product table 214 can be related to many rows in the line item table 212, while a row in the line item table 212 can be related to only one row in the product table.

A row in the customer table 202 can be related to many rows in the customer address table 204 and a row in the customer address table 204 can be related to one row in the customer table 202. A row in the customer address table 204 can be related to one row in the address table 206 and a row in the address table 206 can be related to many rows in the customer address table 204. A row in the address table 206 can be related to one row in the country table 208, while a row in the country table 208 can be related to many rows in the address table 206.

The creation of a physical database can include the introduction of a database table and relationships that only exist to simplify the management and maintenance of the database. For example, database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy. Normalization usually involves dividing large tables into smaller tables and defining relationships between them. A normalized database isolates data so that additions, deletions, and modifications of a field can be made in just one table and then propagated throughout the rest of the database using the defined relationships. Database normalization can

introduce tables that do not necessarily correspond to concepts readily understood by a business user. For example, normalized databases generally introduce tables to resolve many-to-many relationships. For example, a customer may have many addresses and many customers can reside at the same address; this many-to-many
5 relationship between the customer table 202 and the address table 206 has been resolved through the introduction of a customer address table 204. These tables that serve only to resolve a many-to-many relationship between other tables may be referred to as link tables. In general, a link table may have little meaning to a business user.

10 A meta-model generator generates the meta-model 216. In general, the meta-model 216 describes the relationships between data elements in a manner that is more closely related to the business and less closely related to the physical implementations of the data set. The relationships in the meta-model are bi-directional. For example, a customer 218 has one or more addresses 220 and an address 220 has one or more
15 customers 218. Similarly, a customer 218 has one or more orders 226 and an order 226 has one customer 218.

Relationships in the meta-model can be characterized as reference relationships, where the object is associated with only one other object (for example, the relationship between the order object 226 and the customer 218). Alternatively,
20 relationships in the meta-model can be characterized as collection relationships (for example, the relationship between a customer 218 and orders 226).

The model generator may initially generate an object for tables in the relational database. The model generator can create fields on the object that correspond to fields in the underlying table. The model generator can also identify
25 relationships between objects in the meta-model by analyzing the underlying data structures; for example, the model generator may identify a primary key-foreign key relationship between the customer table and the order table. Based on the presence of a foreign key to the customer table in the order table, the model generator creates a reference relationship between the order object and the customer object and a
30 collection relationship between the customer object and the order object.

The names can be determined based on a predetermined naming convention. For example, a field in a database may be labeled using camel case to separate words (for example, "FirstName") or the field in the database may be named using underscores (e.g., "first_name"). The model generator may create a field "First

Name” for the meta-model based on translating the convention into a user-friendly name. Translations can be performed using heuristics, regular expressions, or explicit conversion routines. In some implementations, the names used in the meta-model may be natural language names. A natural language or ordinary language is any
5 language which arises, unpremeditated, in the brains of human beings. Typically, therefore, these are the languages human beings use to communicate with each other, whether by speech, signing, touch or writing. They are distinguished from constructed and formal languages such as those used to program computers or to study logic. These names may be created from unrestricted character sets that include reserved
10 characters. A reserved character is a letter or symbol that cannot be used because it is being utilized in another location, by the database, or by the operating system. For example, many operating systems reserve the following characters: "\", /, :, *, ?, ", <, >, and |" and disallow these characters from being used. Some databases restrict the use of the space character. In some implementations, the set of restricted characters may
15 be dependent on the underlying database or data store.

In some implementations, the model can identify one or more fields of data that serve as the default field when the object is included in a report. For example, the customer object may have “first name” and “last name” as the two default fields. In some implementations, the default field can be a formula applied to the one or more
20 fields; for example the default field for the customer object may be the concatenation of the first name and the last name. The default fields for the order object 226 may be the order number. The default field for the product object 228 may be the product name. Each of the other objects in the model may also have default fields associated with them.

25 The meta-model 218 can also include information about the fields including the data type and any constraints placed on the field. For example, the customer object 218 may include a name field, which is a string, and a birth date field, which is a date.

A user may add enrichments to the meta-model. For example, represented by
30 a dotted line in FIG. 2, a user may create a link between the customer object 218 and the product object 230. The model generator may determine the cardinality of the relationship based a navigational path between the selected objects. In this example, the customer object 210 to the order object 226 to the line item object 228 to the product object 230. Based on the relationship between these objects the model

generator may determine if the relationship between the two selected objects is a one-to-one, one-to-many, many-to-one, or many-to-many relationship.

FIG. 3 illustrates an example of a user interface for building reports. In this example, on the left side 302 of the user interface are elements of the chosen object
5 organized into a hierarchy. A user selects a starting object as the subject object of the report (in this example, the customer object 218 of FIG. 2). The subject object can be selected before the user interface is displayed or may be selected using a user interface element displayed on the user interface (not shown).

The user interface shows a hierarchy of objects using the subject object as a
10 starting point.

Under the customer element 304 is an addresses element 306 and an orders element 310. Under the address element is a country element 307. Under the orders element is a line items element. Under the line items element 312 is a product element 314. As discussed above, in the meta-model, each relationship has a
15 corresponding reverse relationship that allows the relationships in the meta-model to be navigated in two directions (for example, from the customer object 218 to the order object 226 and from the order object 226 to the customer object 218). In some implementations, the user interface does not display reverse relationships. For example, the customer element 304 has a nested addresses element 306, however,
20 because the customer element 304 immediately precedes the addresses element 306 in the hierarchy, the user interface does not display a customer element under the addresses element 306. In other words, the user interface does not allow the user to navigate from customer to address and back again to customer. Because the user interface displays the address element based on navigating the meta-model from
25 customer to address, the user interface will not navigate the relationship from address to customer. In some implementations, the user interface includes logic that prevents the presentation of circular references.

Icons in the user interface in the hierarchy can indicate the cardinality of the relationship between the two objects in the meta-model. For example, referring to
30 FIG. 2, the model 216 indicates that one customer object 218 may be related to many orders 226; that is, the cardinality of the relationship between the customer object 218 and the order object 226 is one-to-many. Similarly, each order object 226 is associated with one customer object 218, therefore the cardinality of the relationship between the order object 226 and the customer object 218 is many-to-one. Referring

again to FIG. 3, the orders data element 310 nested under the customer data element 304 is presented in the plural (e.g. “orders” as opposed to “order”). Presenting the object name in the plural reflects that a customer (the higher-level data element) is related to many orders. Furthermore, an icon can be used to provide visual indications; for example, the icon present next to the orders element 310 shows multiple folders, indicating that the customer can have multiple orders. In contrast, the product element 314 is described using a singular noun and includes an icon showing a single folder. This would indicate that each line item can have only one product.

The user interface can also display objects based on enrichment relationships in the model. For example, the products 315 element is nested under the customer element 304.

In addition to objects, the user interface can also display fields of the objects (for example, the birthdate field 308). Field names allow the user to add additional fields for an object to the report.

In some implementations, the user interface can load the entire meta-model into memory. Loading the meta-model into memory is more efficient and enables the hierarchy to be navigated in a manner not present in other report builders. For example, the user interface 300 may hold the meta-model 216 of FIG. 2 in memory.

When a user expands the customer element 304, the user interface 300 refers to the meta-model 216 in order to determine how to populate the hierarchy. In this example, referring to FIG. 2, the customer object 218 is related to the order object 226 and the address object 220. Consequently, referring again to FIG. 3, the user interface adds the order element 306 and the addresses element 306 to the hierarchy nested under the customer element. Similarly, when the user expands the orders element 306, the user interface 300 may refer to the meta-model 216 of FIG. 2. In this example, referring to FIG. 2, the order object 226 is related to the line item object 228. Consequently, referring again to FIG. 3, the user interface adds the line item element 312 to the hierarchy nested under the order element 310.

The user interface can also include a right side where a user may create a report, as discussed further below. Referring to FIG. 4, a user can select an element from the left side 302 of the user interface and drag it to or otherwise place the element on the right side of the user interface. Adding an element to the right side 338 of the user interface 300 generates a column in the report based on the added

element. The column may include a header and data from the underlying data. For example, dragging the customer element 304 to the right side 338 of the user interface 300 generates column 340 for the report that displays the name of the customer.

Similarly, dragging the orders element 310 to the right hand side 338 of the user interface 300 causes the order number column 342 to be added to the report.

For example, if a user drags the customer element 304 onto the canvas 322 on the right-hand side of the user interface 300, then the report is populated with the default field from the customer object in the underlying model. In this example, the default field is the customer name. The report can be populated in real-time or near-real-time with the values from a field or fields corresponding to the customer name default field from the rows in the customer table in the underlying data source (for example, the data source 102 of FIG. 1). In this example, the report is populated with the customer names "John Smith" and "Steve Miller."

In some implementations, an initial field may be added to the report based on its report subject.

For example, in one implementation, dragging the customer element 304 onto the canvas 322 can cause the generation and execution of the SQL statement:

```
SELECT CUSTOMER_TABLE.CUSTOMER_NAME FROM CUSTOMER_TABLE;
```

When the user drags the orders element 306 onto the canvas 322 then the report is populated with the default field from the order object; in this example, the default field for the order object is the order number 326. As described above, as soon as the order element is added to the canvas, the corresponding relationships and queries can be generated and obtained for the database. In general, objects and fields can be placed in any position in the report. For example, a user can place a new object before, after, or between existing columns. The user interface also enables the user to reorder the columns.

The values used to populate the report are taken from rows of the order table in the underlying data source, which are related to the customers that are already present in the report. In this case, the row of data in the customer table for John Smith is related to two orders: Order # 12345 and Order # 54321. For example, the row in the order table with the Order # 12345 has a field which is a foreign key that references the primary key associated with the row in the customer table for John

Smith. The row of data in the customer table for Steve Miller is associated with two rows in the order table: Order # 84053 and Order # 12331.

For example, in one implementation, dragging the order element 306, after customer element 304, onto the canvas 322 can cause the generation and execution of the SQL statement:

```
SELECT CUSTOMER_TABLE.CUSTOMER_NAME, ORDER_TABLE.ORDER_NUMBER
FROM CUSTOMER_TABLE, ORDER_TABLE
WHERE CUSTOMER_TABLE.CUSTOMER_ID = ORDER_TABLE.CUSTOMER_ID;
```

10

In this example, the relationship described in the WHERE clause is included between the customer element 304 and the order element 306 based on the hierarchical relationship.

Similarly, if the user drags the products element 308 onto the canvas 322, the report can be updated to display the default field for the product data object (in this case, the product name 328). In this case, the report includes the relationship between the product table and the order table based on the selection of the product element 308, which is located hierarchically under the order element 306.

In this example, Order # 12345 includes two products: the golf ball and iron set. Order # 54321 includes the product golf bag. Order #84053 includes two products: the golf balls and golf tees. Order # 12331 includes the products driver and wood set.

The user interface provides aggregation functions to be added to the report. Aggregation functions automatically perform an operation on the values of a field. Examples of aggregation functions include COUNT, which displays the total number of rows within a field (depending on the attributes you specify). Specifically, COUNT returns the number of non-NULL values in the field; COUNTDISTINCT which displays the number of unique values in a field; MIN displays the minimum value within a field. MAX displays the largest value within a field. SUM provides a total of the value of the fields. AVG displays the average value (arithmetic mean) of a field; it sums the values in a field and divides the sum by the total number of values; and EXISTS displays a boolean value (true or false) indicating whether any non-NULL values exist in a field.

Some aggregation functions are only available for fields of different types. For example, MIN and MAX can be used with both date and numeric fields. AVG and SUM can be used only with numeric fields.

5 In general, aggregation functions perform operations on collection fields. A collection field is composed of a set of values on which the aggregation function can operate. The user interface makes aggregation functions available for collection fields, provided the field is valid as an aggregation function.

The user interface may only display aggregation functions that are appropriate for the field and type. For example, the user interface would not display an AVG
10 aggregation on a string field. Determining whether an aggregation function is appropriate can include analyzing the meta-model for field name, types, and cardinality. For example, the user interface may determine that for an aggregation to be appropriate, there must be a one-to-many relationship between a given object (e.g., customer) and the object and field being aggregated (e.g., order, line item).

15 The user interface 300 includes an aggregation function section 316. The aggregation section can be populated based on the selected report subject of the report. In this example, the aggregation functions 316 includes an order aggregation functions 318. Under the order aggregation functions 318 is a count function of orders 320 and a count distinct of orders 322 (that is, an aggregation function that
20 presents the number of unique values).

Additional aggregation functions can be displayed based on a further navigation of the meta-model. For example, under the orders aggregate functions 318 are line item aggregation functions 324. The line item aggregation functions include a count 326 and a count distinct 328 for the line items.

25 The user interface also enables the user to group columns together and add a name that straddles the grouped columns.

The user interface may also enable users to save reports (for example, by selecting a save icon 330 on the user interface). The save feature can store the selected elements, filters (discussed further below), and/or selected fields in a
30 persistent data store (for example, a flat file system). Saving a report can include providing a name or reference that can be used to identify the saved report.

The user interface may also enable users to load the saved report (for example, by selecting a load icon 332 on the user interface). The user interface may also enable users to refresh or re-execute the report (for example, by selecting the refresh icon

334). While each of these features is described in terms of an icon, other visual representations can be used, including but not limited to, regions on a ribbon, buttons, menu selection items, etc,

FIG. 5 illustrates creating an example report. The user interface 300 can display additional information about a simple field, reference field, or collection. For example, hovering over the country element 307 can cause a pop-up box 502 to appear. The pop-up box can display additional information about the country element 307, including the name of the underlying data set, the name of the corresponding field in the underlying data set, and the data type of field.

The user interface 300 can also allow a user to apply a filter to a field in the report or a field related to an object that is included in the report. For example, the user may elect to filter the customer element 304. The system may present a filter user interface area 504. The filter user interface can include a hierarchy of different fields and objects upon which the report may be filtered. For example, the filter interface area 504 includes an addresses collection 506, a name field 508, and an orders collection. If a user selects, for example, the orders collection, the orders collection may expand to present simple fields, reference fields, and collection fields associated with the order object, as described above.

The user may select an element (for example, the name element 508) and drag the element onto the right side 322 of the user interface 300, in a manner similar to selecting a field to be included in the report. The user interface may display filter information; this can include, for example, the name of the field upon which the report is to be filtered, a selection box 514 that enables the user to select a type of filter, and a value field 516 that allows the user to provide text information into the filter. Types of filters include but are not limited to determining if a field begins with a value, ends with a value, contains a value, is exactly a value, and is not a value. Some filters may be available based on data type; for example, numeric fields may include filters such as determining if the field is greater than a value or less than a value.

In this example, the user has selected the name field 508, a “contains” filter 514 and entered the term “Steve Miller” as the value 516. In response to the selection of a field, the filter operation, and a value, the report is updated to reflect the filter. For example, the report on the canvas 322 only presents orders placed by or associated with John Smith. In some implementations, filtering may occur

immediately when the field and the value are selected. In other implementations, filtering may require a further action by the user. For example, the user may need to select a filter icon presented on the user interface (not shown).

5 Selecting a different top level element can result in the presentation and creation of different reports. For example, referring to FIG. 6, a user interface 600 for building reports. As in the example above, on the left side 602 of the user interface are elements of the chosen object organized into a hierarchy. In the example, the user has selected the product object as a subject object of the report.

10 Under the product element 604 is a line item element 606 and a customers element. Under the line item element 606 is an orders element 607. Under the order element 607 is a customer element 608. Under the customer element 608 is the addresses element 610. Under the customer element 612 is the addresses element 614.

15 A user can select an element from the left side 602 of the user interface and drag it to or otherwise place the element on the right side of the user interface. Adding an element to the right side 638 of the user interface 600 generates a column in the report based on the added element. The column may include a header and data from the underlying data. For example, dragging the product element 604 to the right side 668 of the user interface 600 generates column 640 for the report that displays
20 the name of the product. Similarly, dragging the customers element 612 to the right hand side 638 of the user interface 600 causes the customer column 642 to be added to the report. Alternatively, dragging the customers element 608 onto the right side 668 of the user interface 600 can have the same effect as dragging the customer element 612 to the right-hand.

25 Dragging the addresses element 614 to the right-hand side 668 of the user interface causes the address column 646 to be added to the report. FIG. 7 is a flow chart of an exemplary process 700 for building reports. The process 700 can be performed by a computer system performing the process.

30 A model is obtained (702). The model can include objects representative of data structures in a data source and links between the objects representative of relationships between the data structures, with each object in the model identifying at least one default field in the corresponding data structure. The data source may be, for example, a relational database. The model may be generated from the data structures in the data source. For example, in one implementation, the model may be

generated based on tables and relationships between the tables in a relational database.

For each object in the model, the model can also define a default field associated with the object. For example, a customer object may have “customer
5 name” as a default field. The default field may correspond to one or more fields in the data structures of the data source. In some implementations, the default field may be a transformed or otherwise modified version of the data stored in the data structure. For example, the default field may be a concatenation of two or more other fields or the default field may include a data transformed into a different format.

10 A user interface is generated (704). The user interface includes elements organized into a hierarchy. Each element can correspond to an object in the model. The user interface can be a custom stand-alone application or a web page sent to a client device.

A user selection of an element is received (706). A user may select an element
15 by, for example, clicking on the element, double clicking, dragging the element from the hierarchy onto a canvas, tapping a touch based user interface, or through any other selection mechanism.

A report is generated (708). To generate the report, the process can identify an object of the model corresponding to the element. For example, if a user selects an
20 element labeled “Customer”, the process may identify a customer object as the requested object. The process can identify a data structure corresponding to the object. For example, the process may identify that the CUSTOMER_TABLE is associated with the customer object in the model. The process can obtain data from at least one default field in the identified data structure in the data source. For example,
25 if the customer name is the default field associated with the customer object, the process can obtain the customer name data from customer data structure in the data source.

The data is presented to the user. The data may be presented without requiring a subsequent or intervening action from the client. The data may be presented as a
30 tabular report or may be otherwise transformed or modified. For example, the data may be aggregated, parsed, charted, put into buckets or otherwise modified before being presented.

FIG. 8 is a flow chart of an exemplary process 800 for building reports. The process 800 can be performed by a computer system performing the process.

A model is created (802). The model can be created based on structured data (for example, a database schema of a relational database). The structured data can include data structures having data elements (for example, tables). Each data element can have named fields.

5 Creating the model can include creating an object in the model based on one of the data structures in the structured data. Translating a name of a field in the structure to a natural language name. Translating the name of the field can be performed using a look up table, heuristics, a conversion function, or through any other transformation mechanism.

10 A field is created on the object having the natural language name.

 Creating a model also includes creating a relationship field that represents a relationship between two data structures in the structured data. For example, a relationship field can be created based on a primary key – foreign key relationship in a relational database. The relationship field may be a collection representing that in
15 some relationships multiple objects may be associated with a single object.

 A hierarchy of objects in the model is generated (804). The hierarchy can be generated by organizing the objects with respect to a starting object. In some scenarios, the starting object can be selected by a user interacting with a user interface.

20 A user interface including elements for one or more of the objects in the hierarchy is generated (806). The user interface can include user interface elements representing the objects in the model. The user interface elements organized according to the hierarchy. The user interface can enable a user to create a report and filter the report using the natural language name. The user interface may also include
25 aggregation functions and filtering functions. Aggregation functions enable a user to aggregate information about a field. Filter functions enable a user to restrict values presented in the report.

 A selection of one or more elements from the user can be received (808). The user may select the elements, for example, by dragging and dropping the elements
30 from one portion of a screen to another portion of the screen.

 A report is be generated (808). To generate the report a system may identify a data structure corresponding to a selected element. Obtain data from at least one field in the identified data structure. And present the data to the user. The report can be generated using a WYSIWYG interface. For example, a user may add a field to the

report and the report may refresh and show the values from a data source. This enables the user to get immediate feedback about the quality of the report allowing rapid iterations.

The report building approach described above can be implemented using a computing system executing suitable software. For example, the software may include procedures in one or more computer programs that execute on one or more programmed or programmable computing systems (which may be of various architectures, such as distributed, client/server, or grid), each including at least one processor, at least one data storage system (including volatile and/or non-volatile memory and/or storage elements), and at least one user interface (for receiving input using at least one input device or port, and for providing output using at least one output device or port). The software may include one or more modules of a larger program, for example, that provide services related to the design, configuration, and execution of dataflow graphs. The modules of the program (e.g., elements of a dataflow graph) can be implemented as data structures or other organized data conforming to a data model stored in a data repository.

The software may be provided on a tangible, non-transitory medium, such as a CD-ROM or other computer-readable medium (e.g., readable by a general or special purpose computing system or device), or delivered (e.g., encoded in a propagated signal) over a communication medium of a network to a tangible, non-transitory medium of a computing system where it is executed. Some or all of the processing may be performed on a special purpose computer or using special-purpose hardware (such as coprocessors or field-programmable gate arrays (FPGAs) or dedicated, application-specific integrated circuits (ASICs)). The processing may be implemented in a distributed manner in which different parts of the computation specified by the software are performed by different computing elements. Each such computer program is preferably stored on or downloaded to a computer-readable storage medium (e.g., solid state memory or media, or magnetic or optical media) of a storage device accessible by a general or special purpose programmable computer, for configuring and operating the computer when the storage device medium is read by the computer to perform the processing described herein. The inventive system may also be considered to be implemented as a tangible, non-transitory medium, configured with a computer program, where the medium so configured that it causes a

computer to operate in a specific and predefined manner to perform one or more of the processing steps described herein.

A number of embodiments of the invention have been described.

Nevertheless, it is to be understood that the foregoing description is intended to
5 illustrate and not to limit the scope of the invention, which is defined by the scope of
the following claims. Accordingly, other embodiments are also within the scope of
the following claims. For example, various modifications may be made without
departing from the scope of the invention. Additionally, some of the steps described
above may be order independent, and thus can be performed in an order different
10 from that described.

What is claimed is:

1. A computer implemented method including:

creating a model based on structured data, the structured data including data structures, each data structure having data elements, each data element having fields, each field having a name, wherein creating the model includes:

creating an object in the model based on a data structure in the structured data;

translating a name of a field in the data structure to a natural language name;

creating a field on the object having the natural language name;

creating a relationship field on the object based on a relationship between data structures in the structured data;

generating a hierarchy of objects in model, the hierarchy organizing the objects with respect to a starting object according to relationship fields on the objects;

generating a user interface including elements for one or more of the objects in the hierarchy, wherein the user interface enables a user to create a report and filter the report using the natural language name;

receiving a user selection of an element from the elements;

generating a report including:

identifying a data structure corresponding to the element;

obtaining data from at least one field in the identified data structure in the data source; and

presenting the data to the user.

2. The method of claim 1, wherein translating a name includes translating the name using a look up table.

3. The method of claim 1, wherein the relationship field is a reference field representing a relationship between one data element in the data structure to one second data element in a second data structure.

4. The method of claim 1, wherein the relationship field is a collection field representing a relationship between one data element in the data structure to second data elements in a second data structure.

5. The method of claim 4, wherein the user interface includes aggregation functions that can be used on the collection field.

6. The method of claim 1, wherein the relationship field is a parent-child field representing a relationship between one data element in the data structure to a second data element in the data structure.

7. The method of claim 1, wherein the structured data is a structured database and the data structure is a table in the structured database.

8. The method of claim 1, wherein the structured data is one or more files.

9. The method of claim 1, wherein generating the report occurs in response to receiving the user selection.

10. The method of claim 1, wherein the user interface enables an interactive creation of reports using a what-you-see-is-what-you-get interface and wherein the user interface provides visual feedback of the report including data values in near-real-time prior to finalizing the report.

11. The method of claim 1, wherein generating a user interface further includes receiving a selection of the starting object from a user.

12. The method of claim 1, wherein the natural language name includes at least one character in a set of restricted characters for the structured data.

13. The method of claim 1, wherein creating a relationship field on the object based on a relationship between data structures in the structured data includes creating a relationship object to represent a primary key – foreign key relationship in the structured data.

14. A non-transitory computer storage medium encoded with computer program instructions that when executed by one or more computers cause the one or more computers to perform operations comprising:

creating a model based on structured data, the structured data including data structures, each data structure having data elements, each data element having fields, each field having a name, wherein creating the model includes:

creating an object in the model based on a data structure in the structured data;

translating a name of a field in the data structure to a natural language name;

creating a field on the object having the natural language name;

creating a relationship field on the object based on a relationship between data structures in the structured data;

generating a hierarchy of objects in model, the hierarchy organizing the objects with respect to a starting object according to relationship fields on the objects;

generating a user interface including elements for one or more of the objects in the hierarchy, wherein the user interface enables a user to create a report and filter the report using the natural language name;

receiving a user selection of an element from the elements;

generating a report including:

identifying a data structure corresponding to the element;

obtaining data from at least one field in the identified data structure in the data source; and

presenting the data to the user.

15. The non-transitory computer readable medium of claim 14, wherein translating a name includes translating the name using a look up table.

16. The non-transitory computer readable medium of claim 14, wherein the relationship field is a reference field representing a relationship between one data element in the data structure to one second data element in a second data structure.

17. The non-transitory computer readable medium of claim 14, wherein the relationship field is a collection field representing a relationship between one data element in the data structure to second data elements in a second data structure.

18. The non-transitory computer readable medium of claim 17, wherein the user interface includes aggregation functions that can be used on the collection field.

19. The non-transitory computer readable medium of claim 14, wherein the relationship field is a parent-child field representing a relationship between one data element in the data structure to a second data element in the data structure.

20. The non-transitory computer readable medium of claim 14, wherein the structured data is a structured database and the data structure is a table in the structured database.

21. The non-transitory computer readable medium of claim 14, wherein the structured data is one or more files.

22. The non-transitory computer readable medium of claim 14, wherein generating the report occurs in response to receiving the user selection.

23. The non-transitory computer readable medium of claim 14, wherein the user interface enables an interactive creation of reports using a what-you-see-is-what-you-get interface and wherein the user interface provides visual feedback of the report including data values in near-real-time prior to finalizing the report.

24. The non-transitory computer readable medium of claim 14, wherein generating a user interface further includes receiving a selection of the starting object from a user.

25. The non-transitory computer readable medium of claim 14, wherein the natural language name includes at least one character in a set of restricted characters for the structured data.

26. The non-transitory computer readable medium of claim 14, wherein creating a relationship field on the object based on a relationship between data structures in the structured data includes creating a relationship object to represent a primary key – foreign key relationship in the structured data.

27. A system comprising:

one or more computers and one or more storage devices storing instructions that are operable, when executed by the one or more computers, to cause the one or more computers to perform operations comprising:

creating a model based on structured data, the structured data including data structures, each data structure having data elements, each data element having fields, each field having a name, wherein creating the model includes:

creating an object in the model based on a data structure in the structured data;

translating a name of a field in the data structure to a natural language name;

creating a field on the object having the natural language name;

creating a relationship field on the object based on a relationship between data structures in the structured data;

generating a hierarchy of objects in model, the hierarchy organizing the objects with respect to a starting object according to relationship fields on the objects;

generating a user interface including elements for one or more of the objects in the hierarchy, wherein the user interface enables a user to create a report and filter the report using the natural language name;

receiving a user selection of an element from the elements;

generating a report including:

identifying a data structure corresponding to the element;

obtaining data from at least one field in the identified data structure in the data source; and

presenting the data to the user.

28. The system of claim 27, wherein translating a name includes translating the name using a look up table.

29. The system of claim 27, wherein the relationship field is a reference field representing a relationship between one data element in the data structure to one second data element in a second data structure.

30. The system of claim 27, wherein the relationship field is a collection field representing a relationship between one data element in the data structure to second data elements in a second data structure.

31. The system of claim 30, wherein the user interface includes aggregation functions that can be used on the collection field.

32. The system of claim 27, wherein the relationship field is a parent-child field representing a relationship between one data element in the data structure to a second data element in the data structure.

33. The system of claim 27, wherein the structured data is a structured database and the data structure is a table in the structured database.

34. The system of claim 27, wherein the structured data is one or more files.

35. The system of claim 27, wherein generating the report occurs in response to receiving the user selection.

36. The system of claim 27, wherein the user interface enables an interactive creation of reports using a what-you-see-is-what-you-get interface and wherein the user interface provides visual feedback of the report including data values in near-real-time prior to finalizing the report.

37. The system of claim 27, wherein generating a user interface further includes receiving a selection of the starting object from a user.

38. The system of claim 27, wherein the natural language name includes at least one character in a set of restricted characters for the structured data.

39. The system of claim 27, wherein creating a relationship field on the object based on a relationship between data structures in the structured data includes creating a relationship object to represent a primary key – foreign key relationship in the structured data.

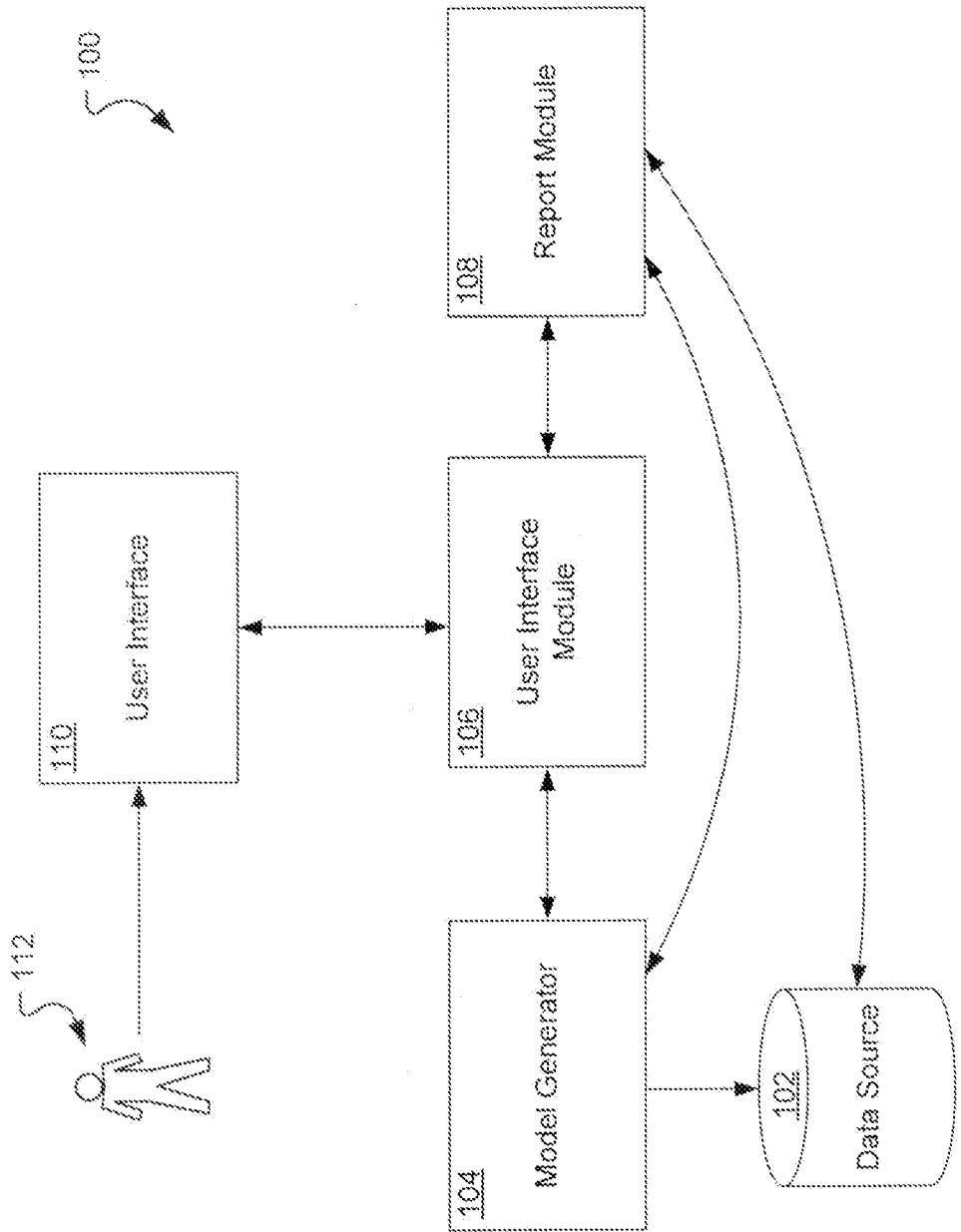


FIG. 1

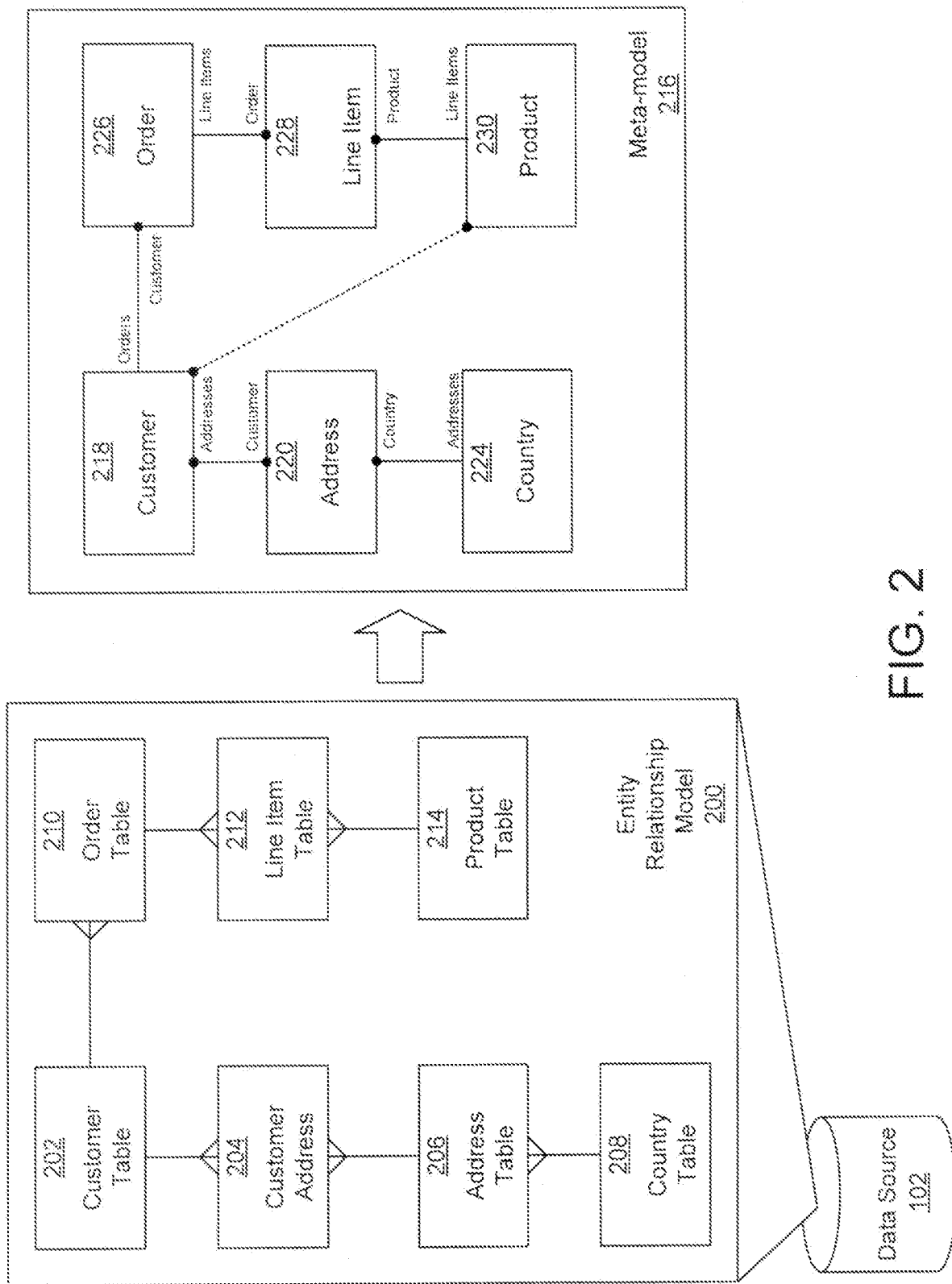


FIG. 2

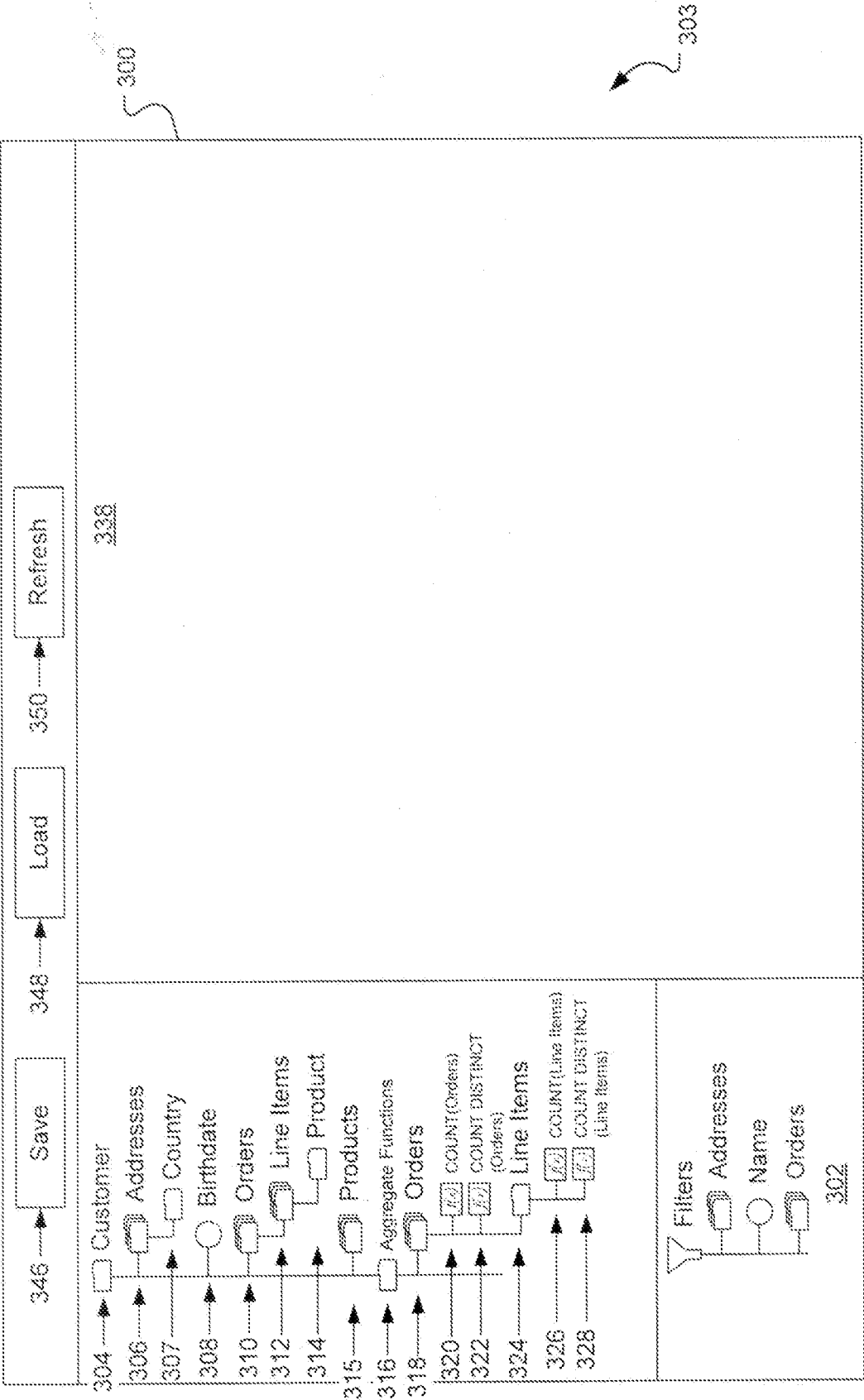


FIG. 3

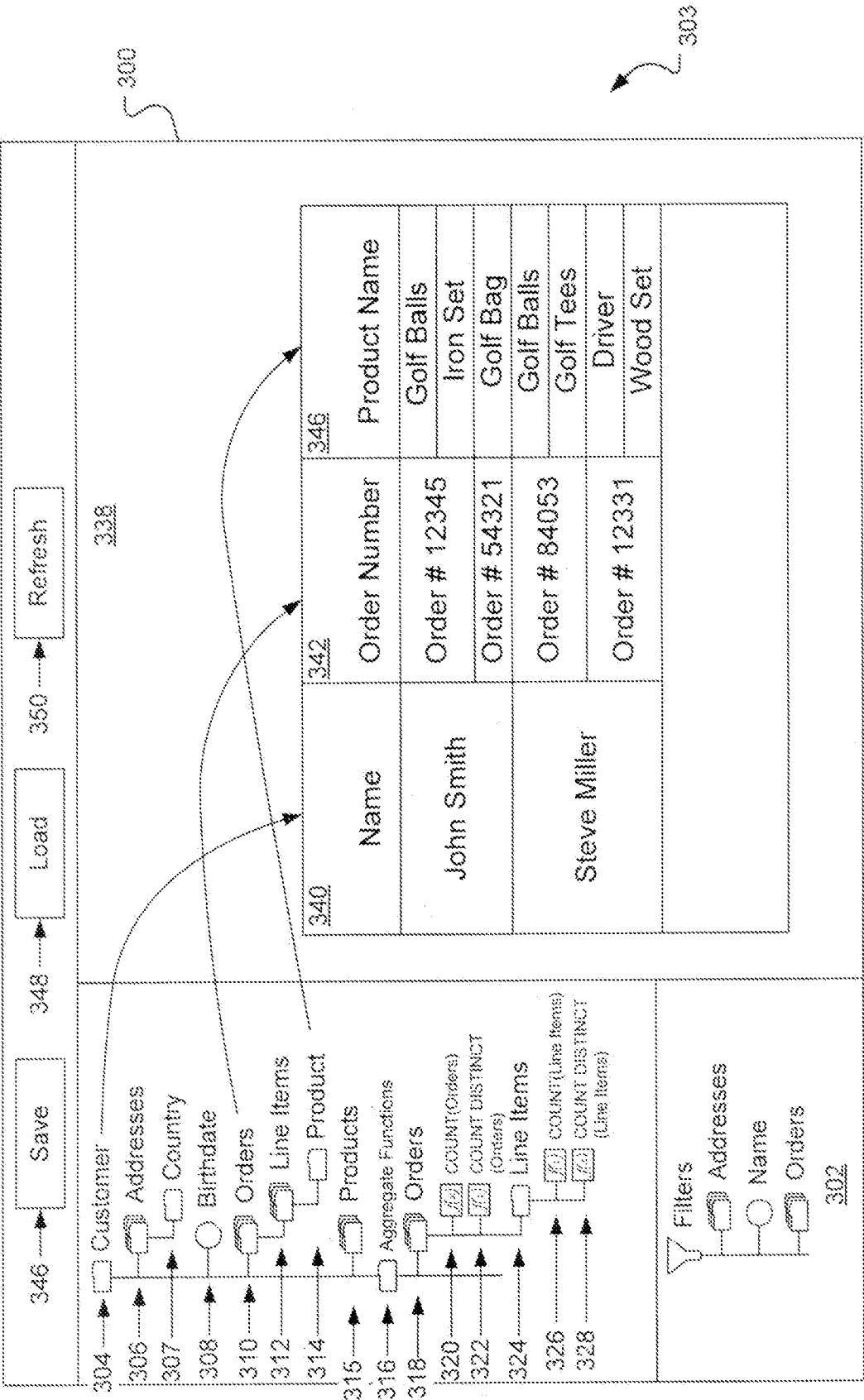


FIG. 4

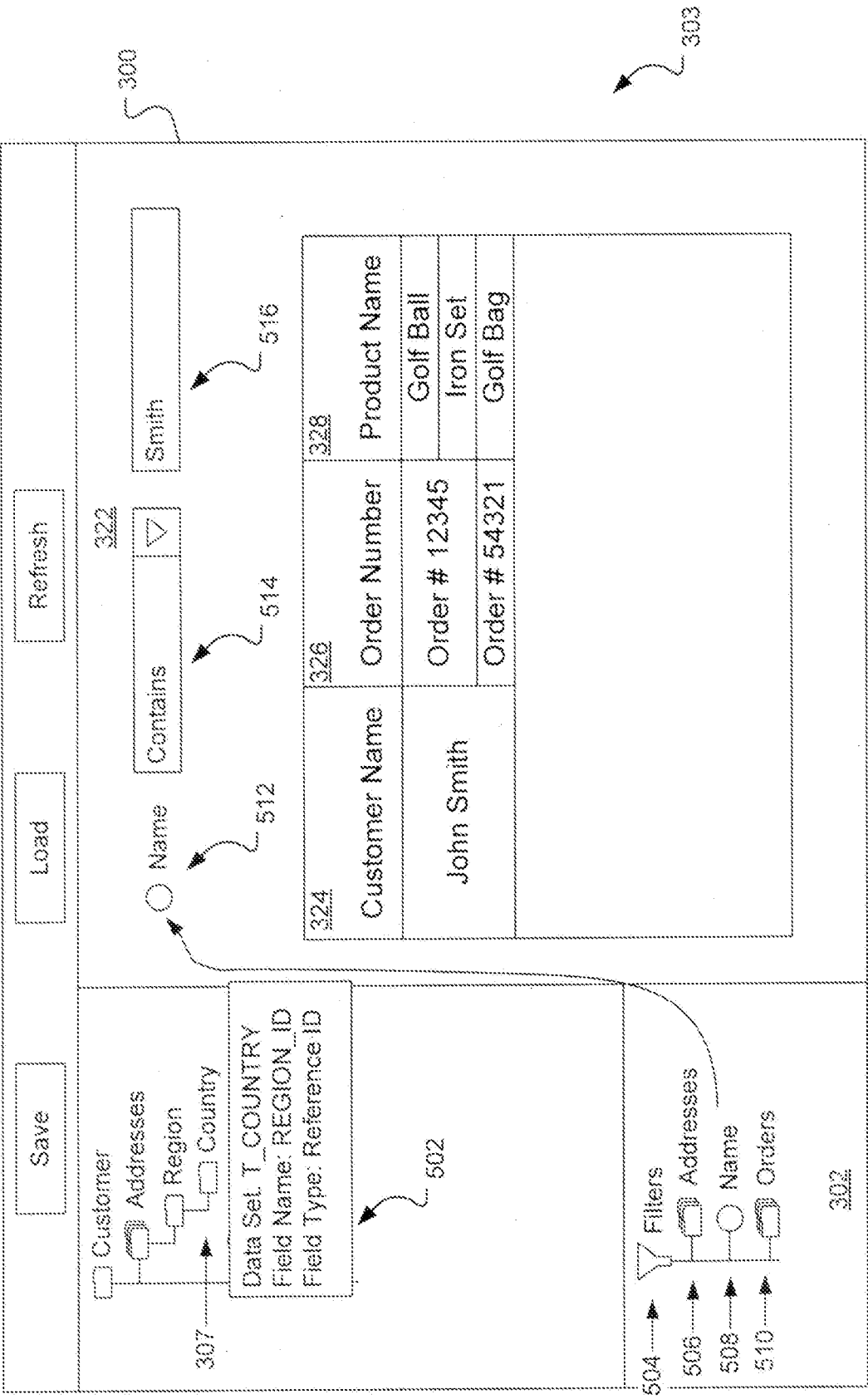


FIG. 5

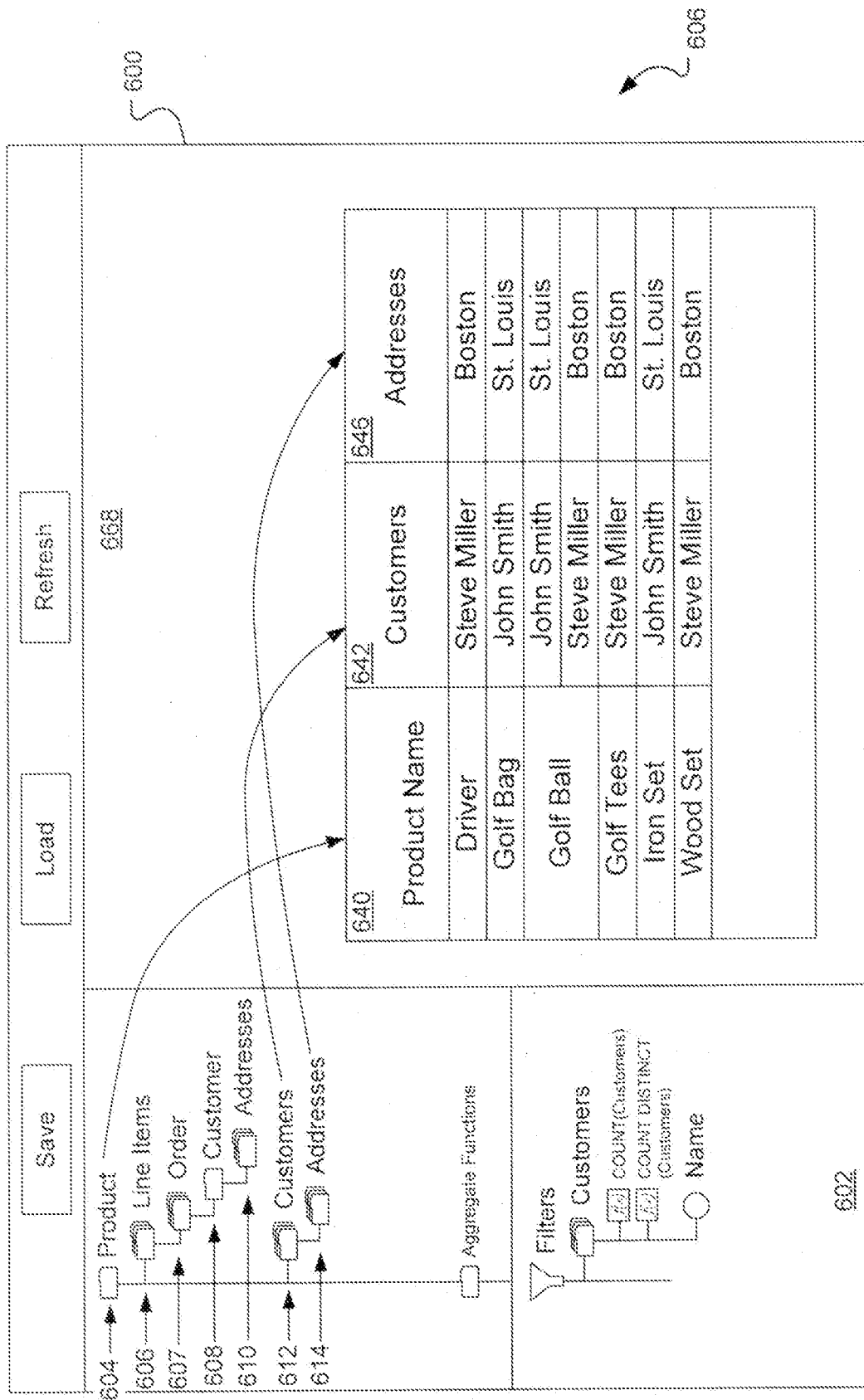


FIG. 6

7/8

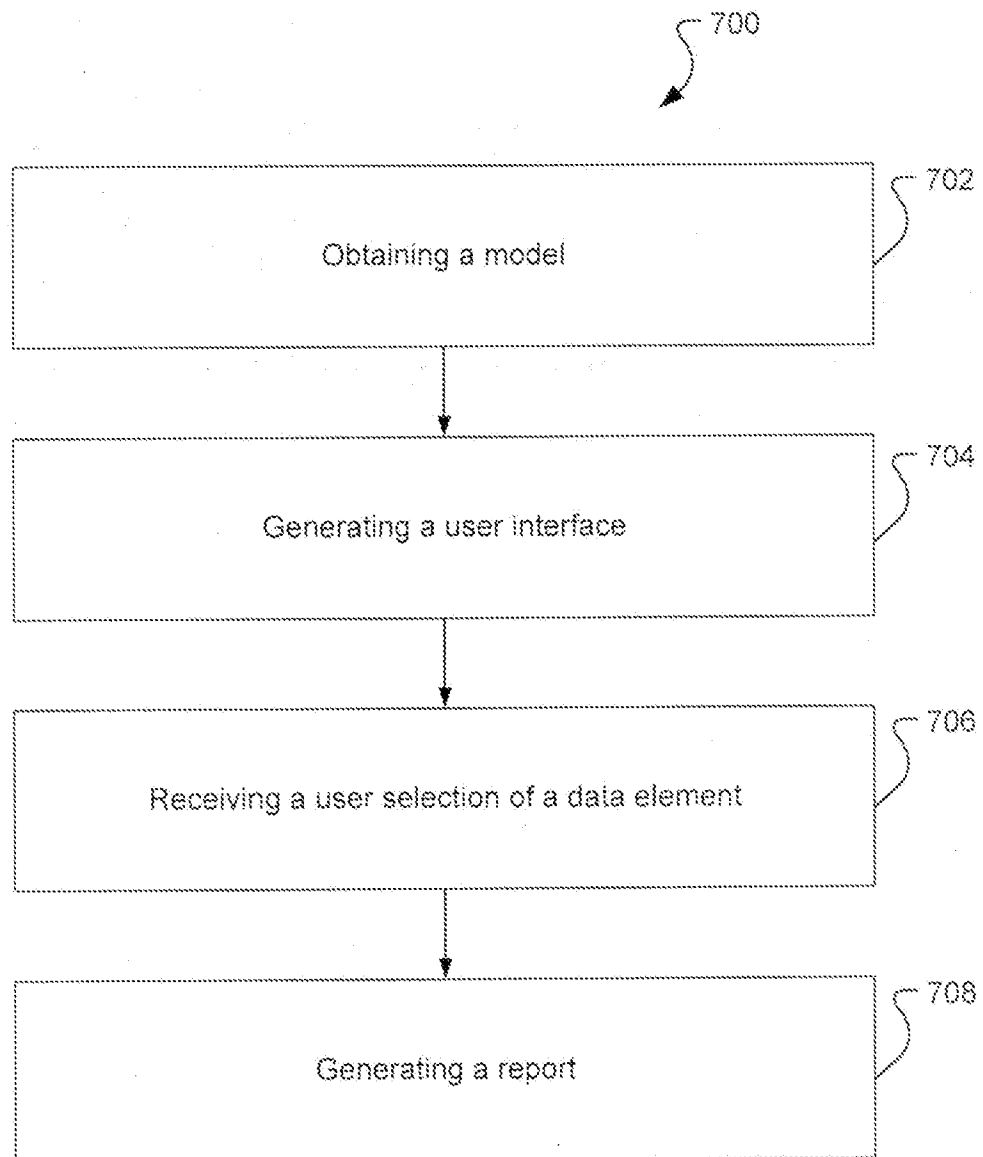


FIG. 7

8/8

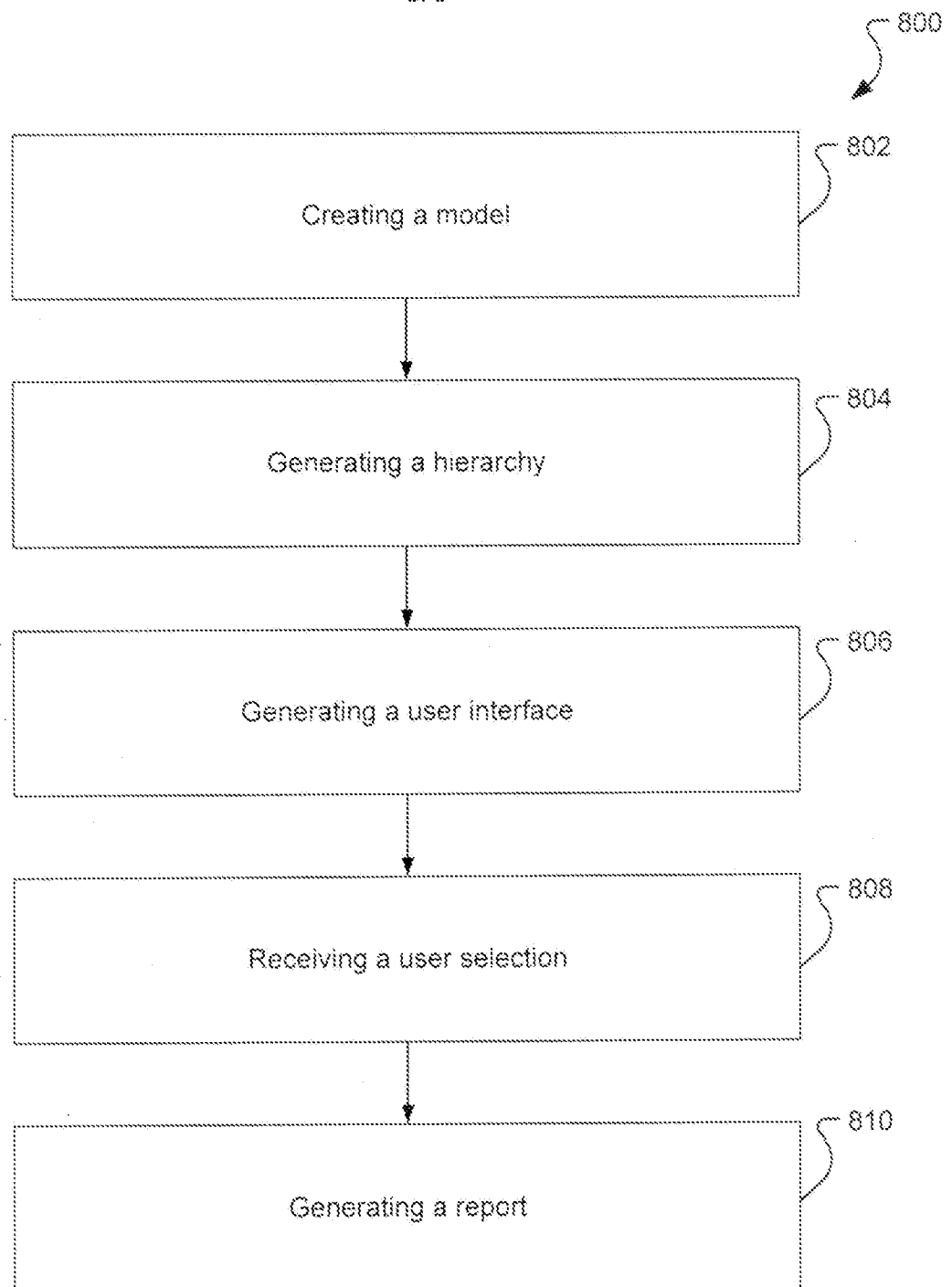


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/066335

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/30 G06Q10/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F G06Q

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

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

EPO-Internal, WPI Data, COMPENDEX, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2006/064383 A2 (COGNOS INC [US]; LEGAULT THOMAS [CA]; ROY DOMINIQUE [CA]; ANTROBUS MAR) 22 June 2006 (2006-06-22) abstract figures 19-21 paragraph [0109] - paragraph [0111] paragraph [0009] claims 7,8 -----	1-39



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

30 March 2016

Date of mailing of the international search report

06/04/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Schmidt, Axel

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2015/066335

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2006064383 A2	22-06-2006	AU 2005315255 A1	22-06-2006
		CA 2588341 A1	22-06-2006
		CN 101111838 A	23-01-2008
		EP 1836620 A2	26-09-2007
		US 2006117057 A1	01-06-2006
		WO 2006064383 A2	22-06-2006
