(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 October 2014 (16.10.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/168961 A1

(51) International Patent Classification:
G06F 19/00 (2006.01)

(21) International Application Number:
PCT/US2014/033356

(22) International Filing Date:
8 April 2014 (08.04.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2013026513 9 April 2013 (09.04.2013) SG

(71) Applicant *(for all designated States except US)*: **EINSIGHTS PTE. LTD** [SG/SG]; 14-01, 20 Cecil Street, Singapore 049705 (SG).

(72) Inventors; and
(71) Applicants *(for US only)*: **RAO, Sandeep, Matade, Shamasundara** [AU/AU]; c/o Einsights Pte. Ltd, 14-01, 20 Cecil Street, Singapore 049705 (SG). **O'RORKE, Paul, Vincent** [US/US]; P.O. Box 494, 105 Redwood Dr, La Honda, CA 94020 (US). **WEERAKOON, Sampath, Bandaralage, Sajith, Vimukthi** [LK/SG]; c/o Einsights Pte. Ltd, 14-01, 20 Cecil Street, Singapore 049705 (SG).

(74) Agents: **HOFFMAN, Brian, M.** et al.; Fenwick & West LLP, Silicon Valley Center, 801 California Street, Mountain View, CA 94041 (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, LT, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,

[Continued on next page]

(54) Title: GENERATING DATA ANALYTICS USING A DOMAIN MODEL

(57) Abstract: A document including a data table is received from a user device, the document being associated with a domain model that is specific to an industry. The data table within the document is identified using the domain model, and a proposed document model is generated using the identified data table and the domain model. The proposed document model s validated using the domain model, and a document model is generated using the validated proposed document model.

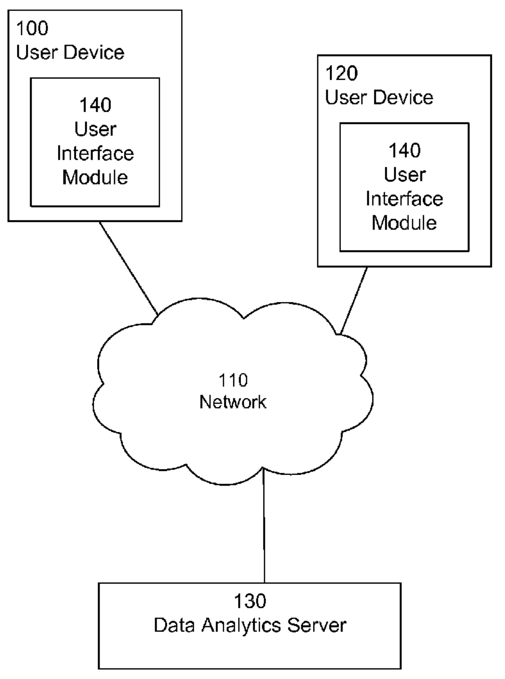

FIG. 1

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))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

GENERATING DATA ANALYTICS USING A DOMAIN MODEL

BACKGROUND

FIELD OF DISCLOSURE

[0001] This disclosure relates to the field of data analytics, and specifically to using an underlying domain model in the generation of data analytics based on natural language queries.

DESCRIPTION OF THE RELATED ART

[0002] Data in any given organization is of utmost value to its decision makers (e.g., the management team). Currently, in order to "interface" with a system for extracting information and/or to perform data analysis, a certain degree of technological expertise is required. Generally, an information technology specialist has to install the system and pre-program queries to extract information from the system for future use by non-technical users (e.g., company decision makers). Such a system may restrict the quality of analysis and the timeliness of the information being made available to a company's decision makers. These have been major deterrents for organizations to derive useful insights into their businesses.

[0003] Moreover, a non-technical person (e.g., without the relevant analytical and/ or IT skills) is usually unable to effectively query a database in a completely ad-hoc manner (e.g., on "non-pre-programmed" parameters, add new sources of data, and conduct complex data analysis). Existing systems that attempt to address these concerns are unsatisfactory. For example, setting up such systems requires strong technical knowledge that generally is present only in outside consultants and/or an inhouse information technology team. Moreover, such systems may require an understanding of specific applications within the firm (e.g., accounting information, sales system, human resources system, etc.) which a company decision maker may not have, and once an existing system has been set up it is generally very difficult to change the parameters of a query, add new data sources, etc.

SUMMARY

[0004] The above and other needs are met by a computer-implemented method, a non-transitory computer-readable storage medium storing executable code, and a device for generating a document model of a document including one or more data tables.

[0005] One embodiment of the computer-implemented method for generating a document model of a document including one or more data tables, comprises receiving a document including a data table from a user device, the document being associated with a domain model that is specific to an industry. The data table within the document is identified using the domain model, and a proposed document model is generated using the identified data table and the domain model. The proposed document model is validated using the domain model, and a document model is generated using the validated proposed document model.

[0006] One embodiment of a non-transitory computer-readable storage medium storing executable computer program instructions for generating a document model of a document including one or more data tables, comprises receiving a document including a data table from a user device, the document being associated with a domain model that is specific to an industry. The data table within the document is identified using the domain model, and a proposed document model is generated using the identified data table and the domain model. The proposed document model is validated using the domain model, and a document model is generated using the validated proposed document model.

[0007] One embodiment of a system for generating a document model of a document including one or more data tables, comprises a processor configured to execute modules, and a memory storing the modules. The modules include a data identification module configured to receive a document including a data table from a user device, the document being associated with a domain model that is specific to an industry, and identify the data table within the document using the domain model. The modules also include a document model module configured to generate a proposed document model using the identified data table and the domain model. The modules also include a model validation module configured to: validate the proposed document model using the domain model, and generate a document model using the validated proposed document model.

BRIEF DESCRIPTION OF DRAWINGS

[0008] FIG. 1 is a high-level block diagram illustrating an embodiment of an environment for generating industry specific document models.

[0009] FIG. 2 is a high-level block diagram illustrating an example computer for implementing the entities shown in FIG. 1.

[0010] FIG. 3 is a high-level block diagram illustrating a detailed view of modules within a data analytics server according to one embodiment.

[0011] FIG. 4 is a flowchart illustrating a process of generating a document model according to one embodiment.

[0012] FIG. 5 is a flowchart illustrating a process of processing a natural language query according to one embodiment

DETAILED DESCRIPTION

[0013] The Figures (FIGS.) and the following description describe certain embodiments by way of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles described herein. Reference will now be made in detail to several embodiments, examples of which are illustrated in the accompanying figures. It is noted that wherever practicable similar or like reference numbers may be used in the figures and may indicate similar or like functionality.

[0014] FIG. 1 is a high-level block diagram illustrating an embodiment of an environment for generating industry specific document models. The environment includes a user device 100 connected by a network 110 to one other user device 120 and a data analytics server 130. Here only two user devices (i.e., 100 and 120) and the data analytics server 130 are illustrated, but there may be multiple instances of each of these entities. For example, there may be thousands or millions of user devices 100 and 120 in communication with multiple data analytic servers 130.

[0015] The network 110 provides a communication infrastructure between the user device 100, the user device 120, and the data analytics server 130. The network 110 is typically the Internet, but may be any network, including but not limited to a Local Area Network (LAN), a Metropolitan Area Network (MAN), a Wide Area Network (WAN), a mobile wired or wireless network, a private network, a virtual private network, or some combination thereof. In some embodiments, the network may have short message service (SMS) capabilities.

[0016] The user device 100 is a computing device that executes computer program modules that allow a user associated with an enterprise (i.e., a business) to upload documents including data tables to the data analytics server 130. A document is a file that includes one or more data tables. A document may be, for example, a MICROSOFT EXCEL file, a PDF, a MICROSOFT WORD document, or any other file that includes at least one data table. A data table is a set of data that is organized using vertical columns and horizontal rows, where an intersection of a row and a column occurs at a data cell. An enterprise is a business within a particular industry. An industry is a group of enterprises that produce/offer a particular type of good and/or service. An industry may be, for example, insurance, health care, automotive, etc. The user device 100 might be, for example, a personal computer, a tablet computer, a mobile phone, a laptop computer, or other type of network-capable device.

[0017] The user device 100 executes a user interface module 140 in one embodiment. The user interface module 140 may be part of a local application specific to the data analytics server 130. Alternatively, the user interface module 140 may be a browser interacting with the data analytics server 130.

[0018] In some embodiments, the user interface module 140 enables the user to interact with documents on the user device 100. For example, the user interface module 140 may present documents to the user, enable the user to create data tables within documents, enable the user to edit data tables within documents, perform some other action relating to data tables, or some combination thereof.

[0019] The user interface module 140 enables a user operating the user device 100 to interact with the data analytics server 130. The user interface module 140 provides a user identifier associated with the user to the data analytics server 130. In some embodiments, the user interface module 140 provides the user identifier as part of a login procedure with the data analytics server 130. The user identifier identifies the user to the data analytics server 130. In some embodiments, the user interface module 140 provides an enterprise identifier (e.g., company name) to the data analytics server 130. The enterprise identifier identifies an enterprise associated with the user.

[0020] The user interface module 140 enables a user to upload documents including data tables to the data analytics server 130. For example, a user associated with an enterprise may upload a document (e.g., a MICROSOFT EXCEL file) that includes information useful to executives of the enterprise. As described below, once this document is converted to a document model, non-technical executives are able to easily query and generate reports using the information in the document model. After a document has been uploaded to the data analytics server 130, the user interface module 140 may receive a proposed document model from the data analytics server 130.

[0021] The user interface module 140 enables the user to validate a proposed document model. A proposed document model is a tentative representation of one or more data tables (or portions thereof) in a document, are associated with an enterprise, and have not been validated. A document model is a proposed document model where the representation of the one or more data tables (or portions thereof) have been validated. The user interface module 140 presents the proposed document model to the user. The user interface module 140 overlays the data tables in the document with one or more graphical indicators that emphasizes a portion of the proposed document model. The graphical indicators may emphasize, for example, portions of the data table to be analyzed, column headers, error notifications, some other portion of the data table, or some combination thereof. A graphical indicator may be, for example, a colored overlay, a change in font size, a change in font color, some other indicator that distinguishes a portion of the proposed document model from the data table as uploaded, or some combination thereof. Additionally, in some embodiments, the graphical indicators may be adjustable by the user. For example, the user may be able to adjust a graphical indicator corresponding to the portion of the proposed document model that the user is selecting as available for analysis.

[0022] In some embodiments, the user interface module 140 enables the user to edit mappings of column headers in a data table in the uploaded document to column headers in the proposed document model. The user interface module 140 enables the user to map column headings in a data table to known column headings and/or create new column headings. This allows the user to easily define new column headers and/or include additional content into the proposed document model.

[0023] The user interface module 140 receives natural language queries from the user that query information in document models associated with the same enterprise as the user and maintained by the data analytics server 130. A natural language query (NLQ) is a query that is phrased in a normal manner according to a given language. For example, a NLQ may be “What are the profits for the fourth quarter of 2013?” The user interface module 140 may receive the NLQ via a user typing out the NLQ. Alternatively, in some embodiments, the user interface module 140 may receive a spoken NLQ via a microphone. Speech to text conversion of the NLQ may occur at the user device 140 or the data analytics server 130. The user interface module 140 provides, via the network 110, the NLQ to the data analytics server 130. Additionally, in some embodiments the NLQ may specify how the results are delivered to the user. For example a NLQ may be “How many total new insurance policies we received this month via SMS?” The user interface module 140 receives from the data analytics server 130 results from the query, and presents the results to the user.

[0024] The user interface module 140 enables a user associated with an enterprise to set access controls associated with one or more document models associated with the enterprise. The access controls determine which users are able to query data from which document models and/or portions of document models. For example, access controls may allow an executive in the San Francisco office of an enterprise to access portions of a document model including salary data for employees in the San Francisco Office, but not portions of the document model that include salary data for employees in other offices.

[0025] The user device 120 is a computing device like the user device 100. The user device 120 includes the user interface module 140 found in the user device 100. In many cases, the technical person uploading the data tables to the data analytics server 130 is not the person who is querying the data analytics server 130 for analyzed data. For example, a user (e.g., accountant) associated with user device 100 may upload a document to the data analytics server 130 to index the document model into the collection of document models associated with her enterprise. Then a different user (e.g., a company manager) who is also associated with the enterprise may use a NLQ to query the collection of document models via the user device 120. Thus, allowing a non-technical manger to easily access and generate reports using the uploaded collection of document models.

[0026] The data analytics sever 130 generates enterprise specific document models that are queryable using NLQs received from user devices (e.g., 100 and/or 120). The data analytics server 130 receives documents from the user devices. The documents may be received from user devices associated with users who are associated with different enterprises. The data analytics server 130 receives a document from a user device 100 associated with a user who is associated with an enterprise. The data analytics server 130 generates a proposed document model of the document based in part on a domain model (as discussed below with regard to FIG. 3) that is specific to the industry associated with the enterprise of the user. The data analytics server 130 validates (e.g., check for errors, obtain user approval of the proposed document model, etc.) the proposed document model using the domain model. The data analytics server 130 generates a document model using the validated proposed document model. The document model is queryable (in accordance with its access controls) by users of the enterprise. The data analytics server 130 may link the validated document model to one or more document models also associated with the enterprise. Additionally, in some embodiments, access controls of the document models control the ability of users associated with an enterprise to search document models associated with the enterprise.

[0027] The data analytics server 130 maintains document models for multiple enterprises, some of which may be associated with same industry. The data analytics server 130 updates a domain model for an industry using document models that are associated with enterprises that are also associated with the industry. For example, as discussed below with reference to FIG. 3, a domain model may identify column headers and update known column headers using one or more domain models.

[0028] The data analytics server 130 processes queries received from user devices (e.g., 100 and 120) in accordance with the access controls. The data analytics server 130 receives a NLQ from a user that is associated with an enterprise (e.g., via a user device 100 or 120). The NLQ request includes a user identifier and/or enterprise identifier which the data analytics server 130 uses to identify a document model associated with the enterprise. The user identifier also allows the data analytics server 130 to determine whether the user meets the access controls associated with the document models being queried. The data analytics server 130 parses the NLQ into one or more commands, and applies the commands to the identified document model to generate one or more results. The data analytics server 130 provides the results to the user device 120.

[0029] The system disclosed herein allows non technical users (e.g., company non-technical executives) to easily perform data analytics using natural language without having to learn a complex query structure. This allows executives to create their own presentations in a very short span of time (e.g., via a simple NLQ) without having to depend on a company resource (e.g., analyst, finance team, technical team, etc.). Additionally, users (including non-technical executives) can add/remove data sources at will thereby enabling them to create new reports using existing document models. Moreover, the documents may be generated using different systems (e.g., human resources, finance, etc.) that are independent from one another. This may result in cost reduction as businesses would not need to implement expensive data solutions that attempt to migrate all of the different systems onto a single platform.

[0030] Turning now to a discussion of the implementation of the entities discussed above, FIG. 2 is a high-level block diagram illustrating an example computer 200 for implementing one or more of the entities shown in FIG. 1. The computer 200 includes at least one processor 202 coupled to a chipset 204. The chipset 204 includes a memory controller hub 220 and an input/output (I/O) controller hub 222. A memory 206 and a graphics adapter 212 are coupled to the memory controller hub 220, and a display 218 is coupled to the graphics adapter 212. A storage device 208, an input interface 214, a speaker 226, and network adapter 216 are coupled to the I/O controller hub 222. Other embodiments of the computer 200 have different architectures.

[0031] The storage device 208 is a non-transitory computer-readable storage medium such as a hard drive, compact disk read-only memory (CD-ROM), DVD, or a solid-state memory device. The memory 206 holds instructions and data used by the processor 202. The input interface 214 is a touch-screen interface, a mouse, track ball, or other type of pointing device, a keyboard, a microphone, or some combination thereof, and is used to input data into the computer 200. The graphics adapter 212 displays images and other information on the display 218. The network adapter 216 couples the computer 200 to one or more computer networks.

[0032] The computer 200 is adapted to execute computer program modules for providing functionality described herein. As used herein, the term "module" refers to computer program logic used to provide the specified functionality. Thus, a module can be implemented in hardware, firmware, and/or software. In one embodiment, program modules are stored on the storage device 208, loaded into the memory 206, and executed by the processor 202.

[0033] The types of computer 200 used by the entities of FIG. 1 can vary depending upon the embodiment and the processing power required by the entity. For example, the data analytics server 130 may include multiple computers 200 communicating with each other through a network such as in a server farm to provide the functionality described herein. Such computers 200 may lack some of the components described above, such as graphics adapters 212 and displays 218.

[0034] FIG. 3 is a high-level block diagram illustrating a detailed view of modules within the data analytics server 130 according to one embodiment. The data analytics server 130 is comprised of modules including a model database 310, a data identification module 320, a document model module 330, a model validation module 340, a query parser module 350, a query processing module 360, and a domain feedback module 370. Some embodiments of the data analytics server 130 have different modules than those described here. Similarly, the functions can be distributed among the modules in a different manner than is described here.

[0035] The model database 310 is a database including one or more models. The models include domain models specific to different industries and document models. In alternative embodiments, the functions of the model database 310 may be performed by two or more databases. For example, a database specific to domain models, and a separate database specific to document models.

[0036] A domain model is an industry specific collection of information that is associated with one or more enterprises. The collection of information in the domain model is used in the creation and querying of document models. The domain model may include, for example, rules indicating expected and/or required content in a domain model for a given industry, one or more header groups, rules indicating expected and/or required column headers in a domain model given particular column headers, rules for identifying data within a domain model, rules defining relationships between data cells in a domain model, common NLQs specific to the industry, analytic tools specific to the industry (e.g., industry specific metrics and/or reports), or some combination thereof.

[0037] The model database 310 also includes document models. A document model is a validated representation of one or more data tables (or portions thereof) in a document. Each document model is associated with a particular enterprise. In some embodiments, a particular enterprise may be associated with multiple document models. The document models for a particular enterprise may be linked such that they are queryable in accordance with the access controls of each document model.

[0038] Additionally, the model database 310 includes a domain lookup table that maps domain models to one or more enterprise identifiers, and an enterprise lookup table that maps user identifiers to their associated enterprise identifiers. Accordingly, given a user identifier and/or an enterprise identifier the model database 310 is able to retrieve an associated domain model.

[0039] The data identification module 320 identifies data tables within the document. The data identification model 320 receives documents from user devices (e.g., 100 or 120). The data identification module 320 may identify data tables within a document by identifying portions of the document that include data in a grid (e.g., data that is organized using columns and rows). Additionally, the data identification module 320 may identify a plurality of data tables within a document. In some embodiments, a document received from a user device (e.g., 100) may specify only a portion of the document be considered by the data identification module 320. For example, the portion may be part of a data table in the document and/or a particular data table within the document. The data identification module 320 identifies data tables within the specified portion of the document.

[0040] The data identification module 320 identifies column headers within identified data tables (or portions thereof) using a domain model. The data identification module 320 identifies a domain model associated with an enterprise using, for example, the enterprise lookup table and/or the domain lookup table. The data identification module 320 retrieves the identified domain model from the domain model database 310. The data identification model 320 identifies column headers using the retrieved domain model and the data table. The data identification module 320 matches column headers in the data table against candidate column headers in different header groups in the domain model. A header group is a collection of one or more known column headers that are associated with the same concept. For example, financial institution, source, XY123, some other identifier, may all be column headers used by users to refer to the concept of a bank. In some embodiments, the header group may include a standard header that is representative of the concept. A standard header may be used to standardize the headers presented in the proposed document model. For example, the data identification module 320 may perform canonicalization on the column headers in the data table using the header groups in the domain model to generate a set of standard headers. The data identification module 320 may provide the standard headers to the document model module 330 for use as column headers in a proposed document model. In alternate embodiments, the column headers in the data table may be provided to the document model module 330 for use as headers in the proposed document model.

[0041] In embodiments where the column header is new (i.e., does not match the known column headers in any of the header groups) the data identification model 320 uses the data table and/or the domain model to identify whether the new column header should be associated with one of the existing header groups, should be associated with a new header group, or should be flagged as a possible error using an error notification. For example, the data identification module 320 may use the cell data in the column of the identified header, other identified column headers, other cell data in the identified column headers, the cell data in the rows, or some combination thereof, to identify whether the new column header should be associated with one of the existing header groups, should be associated with a new header group, or should be flagged using an error notification. The data identification model 320 passes the identified column headers (and any error notifications) to the document model module 330. In alternate embodiments, the data identification module 320 may prompt the user to verify column headings before passing the column headings to the document model module 330.

[0042] The document model module 330 generates proposed document models using identified column headers and associated domain models. The proposed document models represent a tentative understanding of the data tables within a document based on the domain model. For a given data table, the document model module 330 generates a proposed document model composed of a plurality of key value pairs, nested key value pairs, or some combination thereof, that are used to describe the data cells within the data table. A key value pair is a set of two linked data items: a key and a value. A nested key value pair, is a set of key value pairs used to describe a particular data cell and other data associated data cells (e.g., data cells in the same row). Additionally, in some embodiments, the document model module 330 may include one or more error notifications in the proposed document model.

[0043] A key is an identifier that describes a characteristic of data in a data cell (e.g., CITY), and a value is the content of the data cell (e.g., San Francisco). The column header is associated with the same key as that of the data cells within the column. If a data cell is described using a nested key value pair, the nested key value pair may include key value pairs for each data cell in the same row as the data cell. Thus, a single nested key value pair is able to describe all the data associated with a data cell. For example, if a table had a column including two columns titled "Buyer" and "Price Paid" that have respective keys of "NAME" and "DOLLARS," and in the first row the data cells read, respectively, "John Smith," and "1000," a key value pair for the data cell containing John Smith would have a key of NAME and a value of "John Smith." In contrast, a nested key value pair, would additionally include the key value pairs associated with the data cells in the same row, i.e., a key of "DOLLARS" and a value of "1000."

[0044] Key value pairs do not have to specify in advance a format for the data tables that anticipates all possible columns for different kinds of data types (e.g., like quantities). This is useful because different enterprises (even within the same industry) often use different column headers for the same types of information – and different enterprises may include different information in data tables. The keys map to an editable list of column headers (i.e., header group) that provides the data analytics server 130 with flexibility in dealing with additional column headers and/or data that is new to the data analytics server 130.

[0045] The document model module 330 identifies the keys associated with the data cells using the identified columns and the identified domain models. In some embodiments, for a given data table, the document model module 330 looks up the keys that are associated with the column headers. Because a key associated with a column header is also associated with the data cells in the column of the header, the document model module 330 may use the key in the generation of key value pairs and/or nested key value pairs for each of the data cells in the column. The document model module 330 creates the key value pairs by matching the determined key for a data cell with the value contained in the data cell. In embodiments, where there are multiple columns, the document model module 330 creates nested key value pairs for the data cells. The generated key value pairs, nested key value pairs, or some combination thereof represent a proposed document model of the one or more data tables in the document. The document model module 330 passes the generated document model to the document validation module 340.

[0046] In some embodiments, the document model module 330 identifies the keys associated with the data cells using the values of the data cells, and the identified domain models and/or other document models. The document model module 330 compares a value of a data cell in a proposed document model to values that are used in other document models and/or stored in a domain model. Based on the comparison, the document model module 330 selects the key of a matched value, and associates the selected key with the value of the data cell to create a key value pair. Additionally, in embodiments where no match occurs, the document model module 330 may select a key for the key value pair based on other keys associated with other key value pairs in the column, and then update the domain model by including a mapping of the selected key to the value.

[0047] The validation module 340 validates the proposed document model using the domain model. The validation module 340 receives the proposed document model from the document model module 330. The validation module 340 retrieves from the domain model rules indicating expected and/or required content in the domain model given particular column headers in the proposed document model. For example, a domain model specific to the insurance industry may specify that a domain model including a column listing commissions must also include a column listing currencies. The retrieved rules are specific to the industry associated with the data table. The validation module 340 applies the rules to determine if any of the identified columns headers require additional columns or if the proposed document model is missing any expected columns. An expected column includes data that is generally present for a particular industry, but is not required. If an expected and/or required column is not present in the proposed document model, the validation module 340 generates an error notification which may be included in the proposed document model.

[0048] Additionally, in some embodiments, the validation module 340 validates the data type of values in the data cells of the proposed document model against expected data types. A data type is a set of data with values having predefined characteristics. A data type may be, for example, an integer, a floating point number, characters, strings, Boolean, array, etc. The validation module 340 retrieves an expected data type (e.g., integer) for a particular key (e.g., YEAR) from the domain model. The model validation module 340 identifies key value pairs in the proposed document model that include the particular key. The model validation module 340 compares the retrieved data type to the data type of the identified values. If the data types match, the validation module 340 validates the data types of the values within the key value pairs. If the data type does not match, the validation module 340 generates an error notification that is included in the proposed document model.

[0049] The model validation module 340 may provide the proposed document model to the user (e.g., via the user device 100) that uploaded the document for confirmation. Once the confirmation is received from the user device 100, the model validation module 340 validates the proposed document model. Validation of a proposed document model approves the representation of the data tables in the proposed document model. In some embodiments, the model validation module 340 provides the proposed document model to the user if it includes any error notifications. In other embodiments, the model validation module 340 always provides the document model to the user for confirmation, thus providing the user an opportunity to approve and/or edit the proposed document model. In some embodiments, once confirmation is received from the user, a document model is generated.

[0050] The model validation module 340 generates a document model using the validated document model. The model validation module 340 generates the document model by making the validated document model able to be queried by one or more users of the enterprise associated with document model. In some embodiments, a user's ability to query the document model is determined by associated access controls. The model validation module 340 may link the validated document model to one or more document models also associated with the enterprise.

[0051] The query parser module 350 parses NLQs received from a user (e.g., via user device 100 and/or 120) into one or more query commands. The query parser module 350 receives NLQs from user devices. In cases where the received NLQ is an audio file, the query parser module 350 performs speech to text conversion. The query parser parses a received NLQ into one or more query commands. A query command is an instruction that is actionable by the query processor module 360. In some embodiments, the query parser module 350 performs semantics based parsing, syntax based parsing, phonetic based parsing, or some combination thereof, to extract one or more query commands. The query parser module 350 may compare a NLQ to a set of candidate NLQs generated (e.g., as discussed below with regard to the domain feedback module 370) using a domain model associated with the enterprise of the user who submitted the query. Each candidate NLQ is mapped to one or more query commands. The query parser module 350 scores the candidate NLQs based on their similarity to the NLQ. The similarity to the NLQ may be based on, for example, a measure such as estimated probability of correctness or similarity. The query parser module 350 ranks the scored candidate NLQs, and selects the candidate NLQ with the highest score. The query parser module 350 then retrieves the query commands mapped to the selected candidate NLQ. The query parser module 350 provides the query commands to the query processing module 360.

[0052] The query processing module 360 processes query commands using document models. The query processing module 360 retrieves a document model associated with an enterprise associated with the user, and a domain model associated with the enterprise. The query processing module 360 executes the query commands using the retrieved document model. The query processing module 360 extracts information from the key value pairs and/or nested key value pairs within the document model in accordance with the query commands. The query processing module 360 identifies one or more analysis techniques based in part on the query commands, and in some embodiments, the domain model. Analysis techniques may include, for example, regression analysis, cluster analysis, some other analysis technique, analysis techniques specific to the domain, or some combination thereof. The query processing module 360 analyzes the extracted data using the identified one or more analysis techniques. The query processing module 360 generates one or more reports using the analyzed data.

[0053] A report is a graphic or text based answer to a NLQ received from a user. A report may be, for example, a bar chart, a pie chart, a table, a character string, some other method of conveying information to the user, or some combination thereof. In some embodiments, the report is generated in part based on the query commands (e.g., command may specify a pie chart). The query processing module 360 may also use the domain model associated with the enterprise of the user in report generation. For example, the query processing module 360 may retrieve reports that are commonly used in the industry of the user from the domain model. The query processing module 360 may then create the reports using the information extracted from the document model and the domain model. The query processing module 360 provides the generated reports to the users who sent the NLQs. In some embodiments, the query processing module 360 provides the reports via a method indicated in the NLQ (e.g., via SMS). The query processing module 360 may also provide the reports via the same method the NLQ transmitted to the data analytics server 130.

[0054] The domain feedback module 370 updates domain models based on user actions. In some embodiments, the domain feedback module 370 updates the domain model associated with a validated document model based on changes made by the user to the proposed document model. The domain feedback module 370 monitors changes a user makes to the proposed document model. The domain feedback module 370 may update the domain model based on the changes. The domain feedback module 370 may, for example, add new column headings to a header group, specify a new header group, specify a new key, specify new mappings between values of a data cell and keys to a domain model, some change based on user edits to the proposed document model, or some combination thereof, based on changes users make to their respective proposed document models. Thus, the domain feedback module 370 is able to use changes made to the proposed document models that are received from users associated with the same domain model (and possibly different enterprises) to generate proposed document models that require less (and possibly no) changes prior to validation.

[0055] Additionally, the domain feedback module 370 may update the domain model based in part on the queries received from users. For example, the domain feedback module 370 may update the domain model to include NLQs received from users and the NLQs corresponding to query commands. The domain feedback module 370 may generate candidate NLQs based on the NLQs received from users, the NLQs corresponding to query commands, or some combination thereof. The domain feedback module 370 may update one or more domain models with the candidate NLQs. Thus, the domain feedback module 370 is able to use NLQs that are received from users associated with the same domain model (and possibly different enterprises) to better determine the meaning of the NLQs – and generate the appropriate query commands.

[0056] FIG. 4 is a flowchart illustrating a process of generating a document model according to one embodiment. In one embodiment, the process of FIG. 4 is performed by the data analytics server 130. Other entities may perform some or all of the steps of the process in other embodiments. Likewise, embodiments may include different and/or additional steps, or perform the steps in different orders.

[0057] The data analytics server 130 receives 410 a document including a one or more data tables from a user device 100. The document is associated with an enterprise of the user, and the enterprise is associated with a domain model. The data analytics server 130 may retrieve the associated domain model using, for example, an enterprise identifier and the enterprise lookup table.

[0058] The data analytics server 130 identifies 420 the data table within the document using a domain model. The data analytics server 130 identifies the data table within the document by identifying data stored in a grid. The data analytics server 130 identifies column headers using the retrieved domain model and the identified data table.

[0059] The data analytics server 130 generates 430 a proposed document model using the identified data table and the domain model. The data analytics server 130 generates a proposed document model composed of a plurality of key value pairs, nested key value pairs, or some combination thereof, that are used to describe a proposed understanding of the data table.

[0060] The data analytics server 130 validates 440 the proposed document model using the domain model. In some embodiments, the data analytics server 130 verifies expected and/or required content in the proposed document model given the identified column headers using the domain model. The data analytics server 130 may also verify the data type of values in the data cells against expected data types. In some embodiments, the data analytics server 130 provides the document model to the user device (e.g., 100) associated with the user that uploaded the data table for confirmation. The data analytics server 130 receives a confirmation of the proposed document model from the user device 100. The confirmation may include one or more changes to the proposed document model.

[0061] The data analytics server 130 generates 450 a document model using the validated proposed document model. The data analytics server 130 generates the document model by making the validated document model able to be queried by one or more users of the enterprise associated with document model.

[0062] FIG. 5 is a flowchart illustrating a process of processing a NLQ according to one embodiment. In one embodiment, the process of FIG. 5 is performed by the data analytics server 130. Other entities may perform some or all of the steps of the process in other embodiments. Likewise, embodiments may include different and/or additional steps, or perform the steps in different orders.

[0063] The data analytics server 130 receives 510 a NLQ from a user device. The received NLQ may include a user identifier and/or an enterprise identifier. The data analytics server 130 parses 520 the NLQ into one or more query commands. For example, the data analytics server 130 may perform syntax based parsing, phonetic based parsing, or some combination thereof, to extract one or more query commands. In some embodiments, the data analytics server 130 may retrieve a domain model using the user identifier and/or the enterprise identifier. The data analytics server 130 then determines query commands using the retrieved domain model.

[0064] The data analytics server 130 processes 530 the query commands to generate a report. The data analytics server 130 extracts data from the key value pairs and/or nested key value pairs within the document model in accordance with the query commands. The data analytics server analyzes the extracted data, and generates a report using the analyzed information and query commands.

[0065] The data analytics server 130 provides 540 the report to the user. In some embodiments, the data analytics server 130 provides the report in a manner specified by the query commands. In other embodiments, the data analytics server provides the report to the user using the same method the NLQ was provided to the data analytics server 130.

[0066] Some portions of the above description describe the embodiments in terms of algorithmic processes or operations. These algorithmic descriptions and representations are commonly used by those skilled in the data processing arts to convey the substance of their work effectively to others skilled in the art. These operations, while described functionally, computationally, or logically, are understood to be implemented by computer programs comprising instructions for execution by a processor or equivalent electrical circuits, microcode, or the like. Furthermore, it has also proven convenient at times, to refer to these arrangements of functional operations as modules, without loss of generality. The described operations and their associated modules may be embodied in software, firmware, hardware, or any combinations thereof.

[0067] As used herein any reference to "one embodiment" or "an embodiment" means that a particular element, feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

[0068] Some embodiments may be described using the expression “coupled” and “connected” along with their derivatives. It should be understood that these terms are not intended as synonyms for each other. For example, some embodiments may be described using the term “connected” to indicate that two or more elements are in direct physical or electrical contact with each other. In another example, some embodiments may be described using the term “coupled” to indicate that two or more elements are in direct physical or electrical contact. The term “coupled,” however, may also mean that two or more elements are not in direct contact with each other, but yet still co-operate or interact with each other. The embodiments are not limited in this context.

[0069] As used herein, the terms “comprises,” “comprising,” “includes,” “including,” “has,” “having” or any other variation thereof, are intended to cover a non-exclusive inclusion. For example, a process, method, article, or apparatus that comprises a list of elements is not necessarily limited to only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus. Further, unless expressly stated to the contrary, “or” refers to an inclusive or and not to an exclusive or. For example, a condition A or B is satisfied by any one of the following: A is true (or present) and B is false (or not present), A is false (or not present) and B is true (or present), and both A and B are true (or present).

[0070] In addition, use of the “a” or “an” are employed to describe elements and components of the embodiments herein. This is done merely for convenience and to give a general sense of the disclosure. This description should be read to include one or at least one and the singular also includes the plural unless it is obvious that it is meant otherwise.

[0071] Upon reading this disclosure, those of skill in the art will appreciate still additional alternative structural and functional designs for a system and a process for generating document models and querying those document models. Thus, while particular embodiments and applications have been illustrated and described, it is to be understood that the described subject matter is not limited to the precise construction and components disclosed herein and that various modifications, changes and variations which will be apparent to those skilled in the art may be made in the arrangement, operation and details of the method and apparatus disclosed herein.

CLAIMS

1. A computer-implemented method for generating a document model of a document including one or more data tables, comprising:

receiving a document including a data table from a user device, the document being associated with a domain model that is specific to an industry;

identifying the data table within the document using the domain model;

generating a proposed document model using the identified data table and the domain model; and

validating the proposed document model using the domain model; and

generating a document model using the validated proposed document model.

2. The computer-implemented method of claim 1, wherein identifying the data table within the document using the domain model, further comprises:

determining candidate column headers in the document;

matching the candidate column headers against column headers in one or more header groups in the domain model, wherein a header group is a collection of one or more column headers that are associated with a same concept; and

identifying column headers based on the matching.

3. The computer-implemented method of claim 2, wherein a column header does not match the candidate column headers, further comprising:

determining that the column header is associated with the same concept as a header group; and

updating the header group to include the column header.

4. The computer-implemented method of claim 2, wherein validating the proposed document model using the domain model, further comprises:

retrieving from the domain model rules indicating required content in the proposed document model for an identified column header;

determining that the identified column header requires an additional column of required content;

determining that the additional column of required content is not part of the proposed document model; and responsive to the determination that the additional column of required content is not part of the proposed document model, updating the proposed document model with an error notification.

5. The computer-implemented method of claim 1, further comprising:
receiving a change to the proposed document model from the user device; and
updating the domain model based on the change.

6. The computer-implemented method of claim 1, further comprising:
receiving a natural language query from the user device;
parsing the natural language query into one or more query commands, the parsing based in part on the domain model;
processing the one or more query commands using the document model to generate a report; and
providing the report to the user device.

7. The computer-implemented method of claim 6, wherein parsing the natural language query into one or more query commands, the parsing based in part on the domain model, further comprises:
comparing the natural language query to a set of candidate natural language queries generated using the domain model, wherein each candidate natural language query is mapped to one or more candidate query commands;
scoring the candidate natural language queries based on their similarity to the natural language query;
ranking the scored candidate natural language queries;
selecting the candidate natural language query with the highest score; and
retrieving the query commands mapped to the selected candidate natural language query.

8. A non-transitory computer-readable storage medium storing executable computer program instructions for generating a document model of a document including one or more data tables, the instructions executable to perform steps comprising:
receiving a document including a data table from a user device, the document being associated with a domain model that is specific to an industry;
identifying the data table within the document using the domain model;
generating a proposed document model using the identified data table and the domain model; and

validating the proposed document model using the domain model; and

generating a document model using the validated proposed document model.

9. The computer-readable medium of claim 8, wherein identifying the data table within the document using the domain model, further comprises:

determining candidate column headers in the document;

matching the candidate column headers against column headers in one or more header groups in the domain model, wherein a header group is a collection of one or more column headers that are associated with a same concept; and

identifying column headers based on the matching.

10. The computer-readable medium of claim 9, wherein a column header does not match the candidate column headers, further comprising:

determining that the column header is associated with the same concept as a header group; and

updating the header group to include the column header.

11. The computer-readable medium of claim 9, wherein validating the proposed document model using the domain model, further comprises:

retrieving from the domain model rules indicating required content in the proposed document model for an identified column header;

determining that the identified column header requires an additional column of required content;

determining that the additional column of required content is not part of the proposed document model; and

responsive to the determination that the additional column of required content is not part of the proposed document model, updating the proposed document model with an error notification.

12. The computer-readable medium of claim 8, further comprising:

receiving a change to the proposed document model from the user device; and

updating the domain model based on the change.

13. The computer-readable medium of claim 9, further comprising:
receiving a natural language query from the user device;
parsing the natural language query into one or more query commands, the parsing based in part on the domain model;
processing the one or more query commands using the document model to generate a report; and
providing the report to the user device.

14. The computer-readable medium of claim 13, wherein parsing the natural language query into one or more query commands, the parsing based in part on the domain model, further comprises:
comparing the natural language query to a set of candidate natural language queries generated using the domain model, wherein each candidate natural language query is mapped to one or more candidate query commands;
scoring the candidate natural language queries based on their similarity to the natural language query;
ranking the scored candidate natural language queries;
selecting the candidate natural language query with the highest score; and
retrieving the query commands mapped to the selected candidate natural language query.

15. A system for generating a document model of a document including one or more data tables, comprising:
a processor configured to execute modules; and
a memory storing the modules, the modules comprising:
a data identification module configured to:
receive a document including a data table from a user device, the document being associated with a domain model that is specific to an industry, and
identify the data table within the document using the domain model; and
a document model module configured to generate a proposed document model using the identified data table and the domain model; and
a model validation module configured to:
validate the proposed document model using the domain model, and

generate a document model using the validated proposed document model.

16. The system of claim 15, wherein the data identification module is further configured to:

determine candidate column headers in the document;

match the candidate column headers against column headers in one or more header groups in the domain model, wherein a header group is a collection of one or more column headers that are associated with a same concept; and

identify column headers based on the matching.

17. The system of claim 16, wherein a column header does not match the candidate column headers and the data identification module is further configured to:

determine that the column header is associated with the same concept as a header group; and

update the header group to include the column header.

18. The system of claim 16, wherein the model validation module is further configured to:

retrieve from the domain model rules indicating required content in the proposed document model for an identified column header;

determine that the identified column header requires an additional column of required content;

determine that the additional column of required content is not part of the proposed document model; and

responsive to the determination that the additional column of required content is not part of the proposed document model, update the proposed document model with an error notification.

19. The system of claim 8, further comprising:

a query parser module configured to:

receive a natural language query from the user device, and

parse the natural language query into one or more query commands, the parsing based in part on the domain model; and

a query processor module configured to:

process the one or more query commands using the document model to generate a report; and

provide the report to the user device.

20. The system of claim 19, wherein the query parser module is further configured to:

compare the natural language query to a set of candidate natural language queries generated using the domain model, wherein each candidate natural language query is mapped to one or more candidate query commands;

score the candidate natural language queries based on their similarity to the natural language query;

rank the scored candidate natural language queries;

select the candidate natural language query with the highest score; and

retrieve the query commands mapped to the selected candidate natural language query.

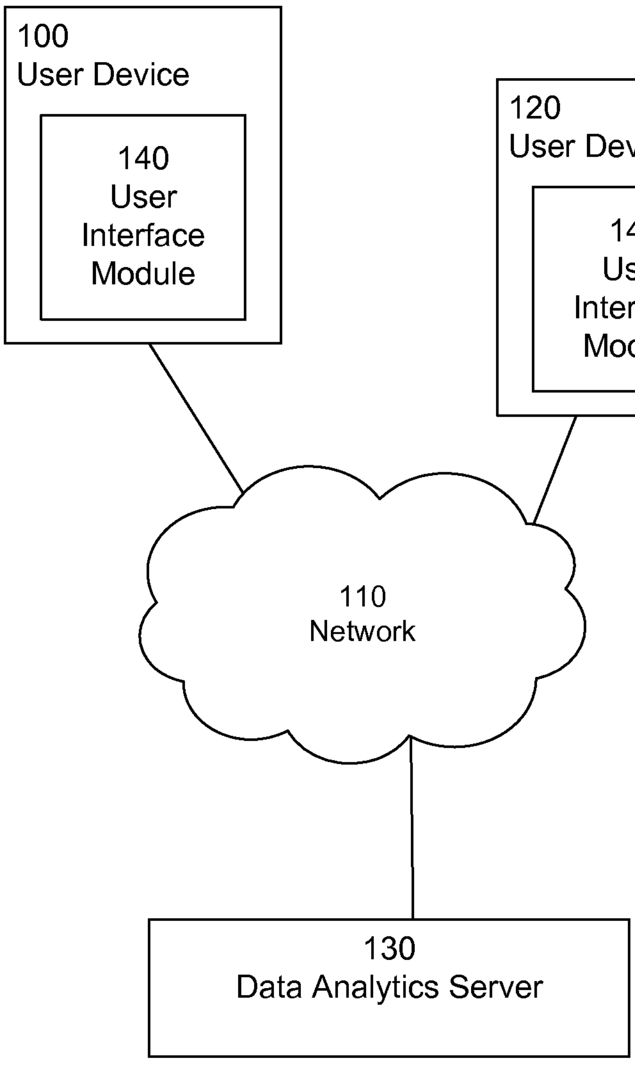
100
User Device
140
User
Interface
Module
120
User Device
140
User
Interface
Module
110
Network
130
Data Analytics Server

FIG. 1

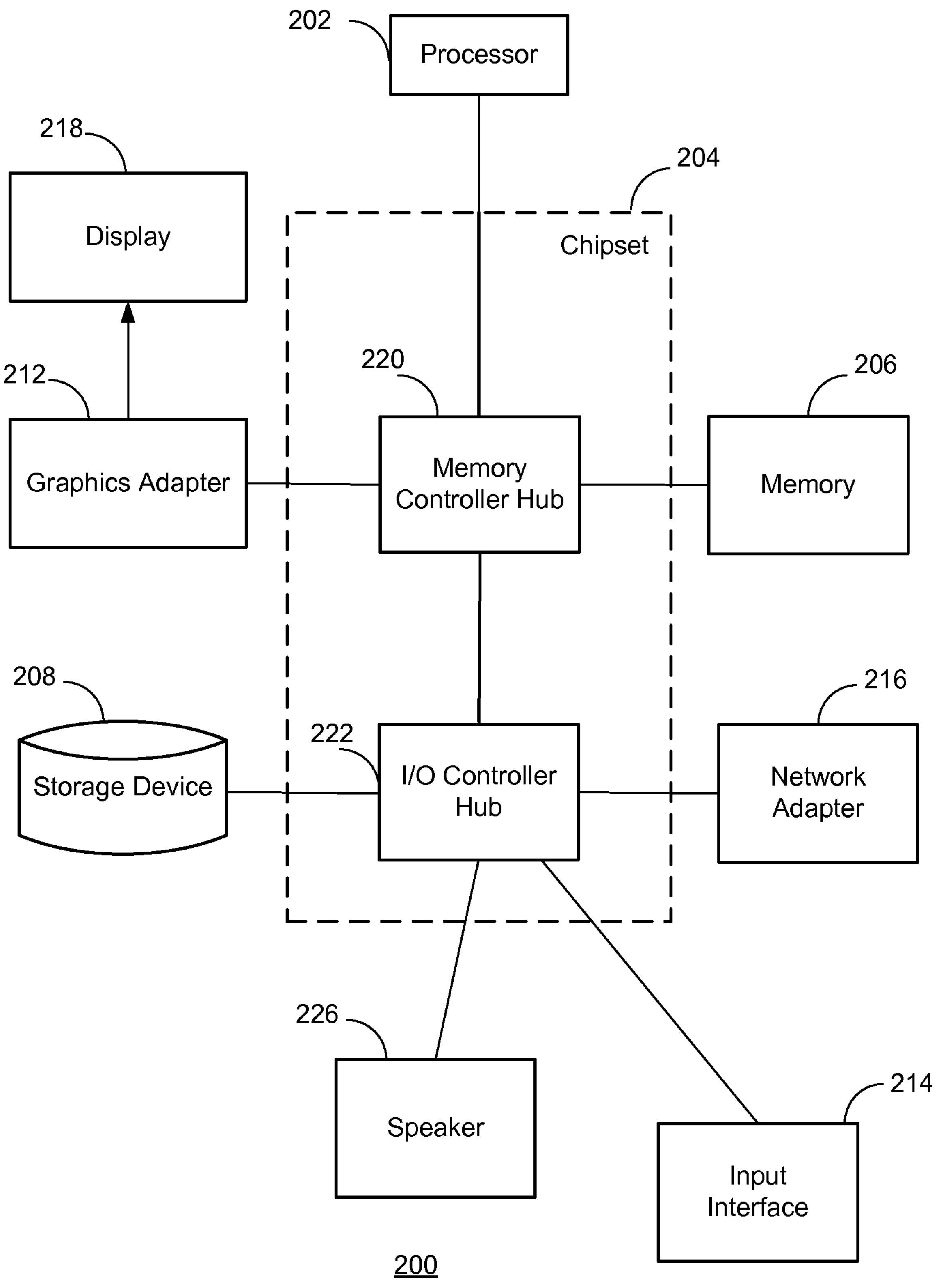
202
Processor
218
Display
204
Chipset
212
Graphics Adapter
220
Memory Controller Hub
206
Memory
208
Storage Device
222
I/O Controller Hub
216
Network Adapter
226
Speaker
214
Input Interface
200

FIG. 2

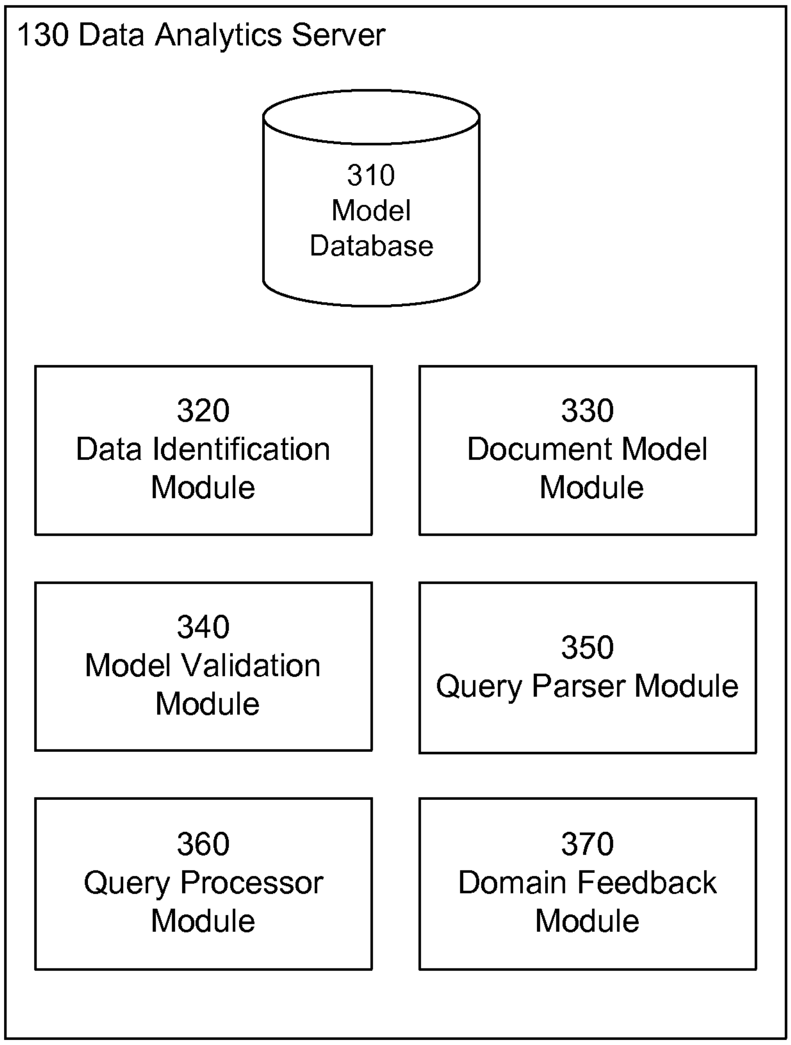
130 Data Analytics Server
310 Model Database
320 Data Identification Module
330 Document Model Module
340 Model Validation Module
350 Query Parser Module
360 Query Processor Module
370 Domain Feedback Module

FIG. 3

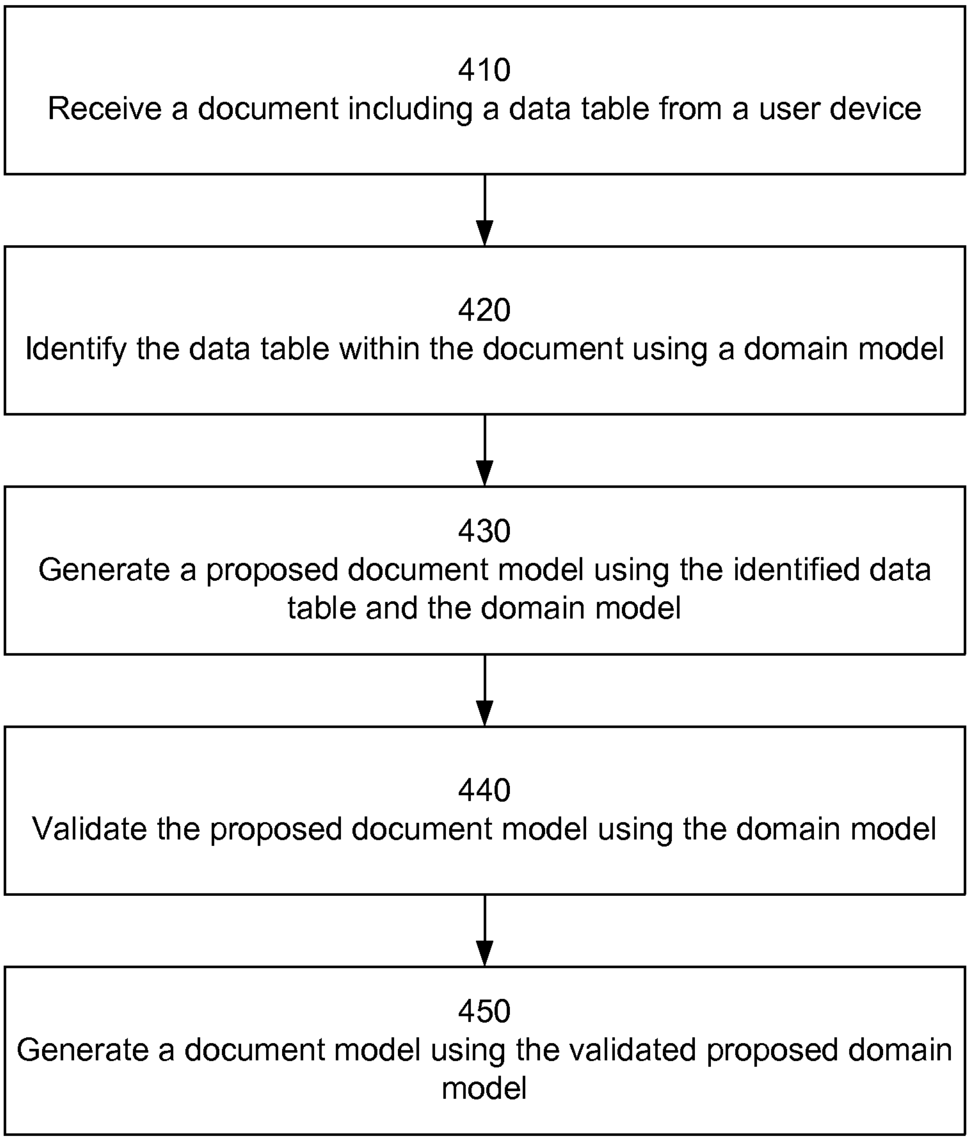
410
Receive a document including a data table from a user device
420
Identify the data table within the document using a domain model
430
Generate a proposed document model using the identified data table and the domain model
440
Validate the proposed document model using the domain model
450
Generate a document model using the validated proposed domain model

FIG. 4

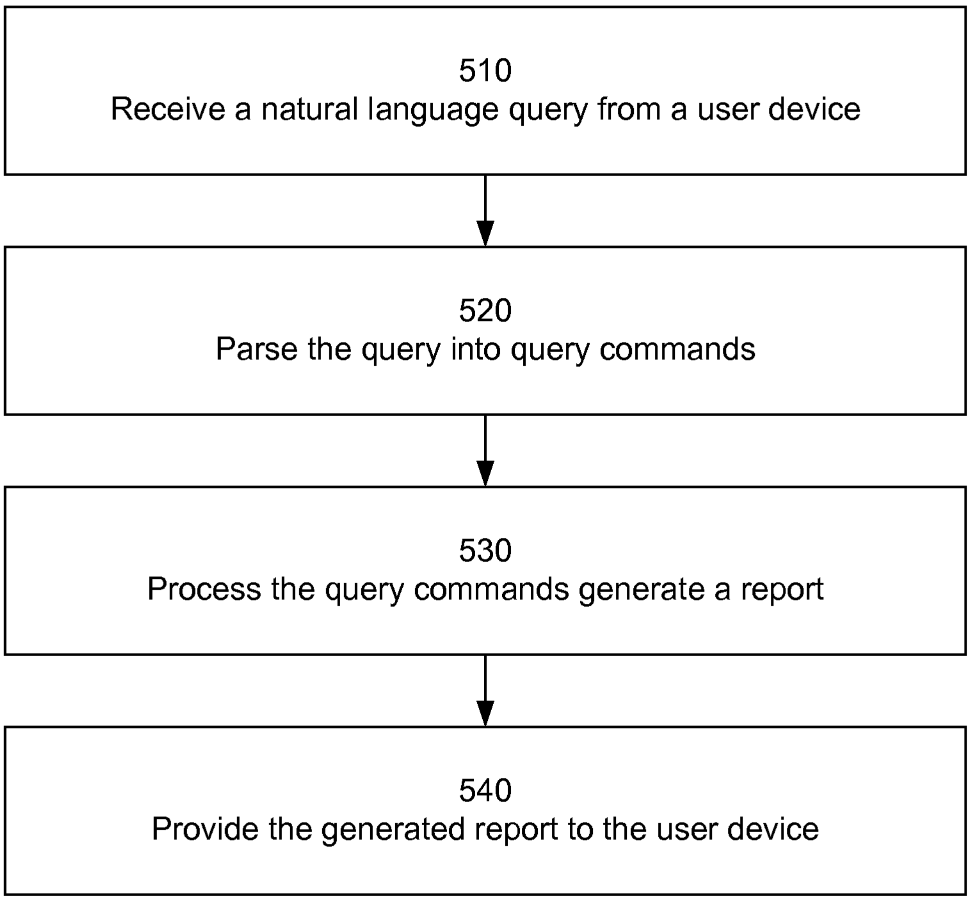
510
Receive a natural language query from a user device
520
Parse the query into query commands
530
Process the query commands generate a report
540
Provide the generated report to the user device

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 14/33356

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 19/00 (2014.01)
USPC - 700/97
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)
USPC: 700/97; IPC(8): G06F 19/00 (2014.01)
G06Q40/04, G06Q10/0633, G06Q30/0601, Y10S707/955, G06F21/6218, G06F17/30696, G06F21/6227 (keyword limited; terms below)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 700/97, 700/95; 715/273, 715/212, 715/267, 715/762, 715/219; 707/805, 707/E17.076; 705/7.27, 705/37, 705/26.1; 709/217 (keyword limited; terms below); CPC: G06Q10/067, G06F17/246, G06Q10/0637, H04L67/02, H04L67/306, G06Q10/10, cont'd.

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PatBase; Google Scholar
Search Terms: document model, domain model, industry, enterprise, business, rule, policy, analytics, metric, statistic, table, column, header, tag, name, topic, concept, subject, generate, create, build, query, search, user, natural language, result, report

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X -- Y	US 2004/0083199 A1 (GOVINDUGARI et al.) 29 April 2004 (29.04.2004), entire document, especially abstract and para [0026], [0029]-[0030], [0050]-[0051], [0073]-[0076], [0078], [0080], [0085], [0090]-[0096], [0103]-[0109], [0125], [0128], [0138], [0167], [0191]- [0192], [0198], [0200]-[0205], [0207]-[0212], claim 1, claim 15, claim 41.	1-5, 8-12, 15-18 -------------------------------- 6-7, 13-14, 19-20
Y	US 2008/0140616 A1 (ENCINA et al.) 12 June 2008 (12.06.2008), entire document, especially abstract and para [0051], [0056], [0058]-[0059], [0063]-[0065], [0111]-[0113].	6-7, 13-14, 19-20
A	US 2010/0100561 A1 (COOPER et al.) 22 April 2010 (22.04.2010), entire document.	1-20

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"&" document member of the same patent family
"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	

Date of the actual completion of the international search 04 August 2014 (04.08.2014)	Date of mailing of the international search report 12 SEP 2014
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774