



Office de la Propriété
Intellectuelle
du Canada

Un organisme
d'Industrie Canada

Canadian
Intellectual Property
Office

An agency of
Industry Canada

CA 2626860 C 2015/12/08

(11)(21) **2 626 860**

(12) **BREVET CANADIEN
CANADIAN PATENT**

(13) **C**

(86) Date de dépôt PCT/PCT Filing Date: 2005/12/13
(87) Date publication PCT/PCT Publication Date: 2007/04/26
(45) Date de délivrance/Issue Date: 2015/12/08
(85) Entrée phase nationale/National Entry: 2008/04/22
(86) N° demande PCT/PCT Application No.: US 2005/045447
(87) N° publication PCT/PCT Publication No.: 2007/046830
(30) Priorité/Priority: 2005/10/23 (US11/257,282)

(51) Cl.Int./Int.Cl. *G06F 7/00* (2006.01),
G06F 17/30 (2006.01)

(72) Inventeurs/Inventors:
REDDY, BINDU, US;
BRUNSMAN, JONATHAN, US;
MOSBERGER, NING, US;
BHAYA, GAURAV RAVINDRA, US;
SIRAJUDDIN, SARAH, US;
KALE, DAVID, US;
KOZENSKI, JENNIFER L., US;
...

(73) Propriétaire/Owner:
GOOGLE INC., US

(74) Agent: GOWLING LAFLEUR HENDERSON LLP

(54) Titre : RECHERCHE DANS DES DONNEES STRUCTUREES
(54) Title: SEARCH OVER STRUCTURED DATA

(57) Abrégé/Abstract:

A user can refine a search over structured data by specifying that a label or an attribute value be used to further filter the results of a query.



(72) Inventeurs(suite)/Inventors(continued): SUNDARARAJAN, ARVIND, US; AGARWAL, PUNEET, US

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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
26 April 2007 (26.04.2007)

PCT

(10) International Publication Number
WO 2007/046830 A3

(51) International Patent Classification:

G06F 7/00 (2006.01) **G06F 17/30** (2006.01)

CA 9403 (US). **AGARWAL, Puneet** [IN/US]; 1200 Dale Avenue, Apt. 42, Mountain View, CA 94040 (US).

(21) International Application Number:

PCT/US2005/045447

(74) **Agents: MAJERUS, Laura, A.** et al.; Fenwick & West LLP, Silicon Valley center, 801 California Street, Mountain View, CA 94041 (US).

(22) International Filing Date:

13 December 2005 (13.12.2005)

(81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

11/257,282 23 October 2005 (23.10.2005) US

(71) **Applicant** (*for all designated States except US*): **GOOGLE INC.** [US/US]; 1600 Amphitheatre Parkway, Building 41, Mountain View, CA 94043 (US).

(84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(72) **Inventors; and**

(75) **Inventors/Applicants** (*for US only*): **REDDY, Bindu** [IN/US]; 1097 Howard Street, #305, San Francisco, CA 94103 (US). **BRUNSMAN, Jonathan** [US/US]; 20 Descanso Drive, #1218, San Jose, CA 95134 (US). **MOSBERGER, Ning** [US/US]; 35706 Runckel Lane, Fremont, CA 94536 (US). **BHAYA, Gaurav, Ravindra** [IN/US]; 925 S. Wolfe Road, #38, Sunnyvale, CA 94086 (US). **SIRAJUDDIN, Sarah** [IN/US]; 2645 California Street, #206, Mountain View, CA 94040 (US). **KALE, David** [US/US]; 3488 South Court, Palo Alto, CA 94306 (US). **KOZENSKI, Jennifer, L.** [US/US]; 1202 Susan Way, Sunnyvale, CA 94087 (US). **SUNDARARAJAN, Arvind** [IN/US]; 1097 Howard Street, Apt. 305, San Francisco,

Published:

— *with international search report*

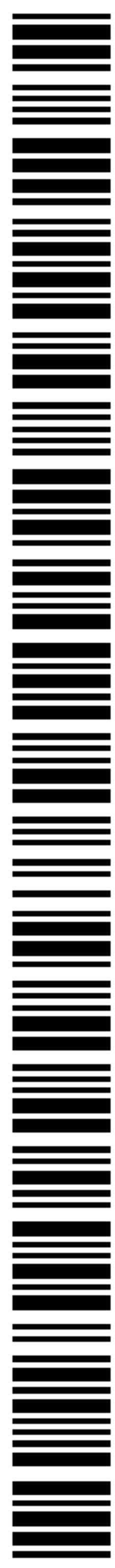
(88) **Date of publication of the international search report:**

15 November 2007

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** SEARCH OVER STRUCTURED DATA

(57) **Abstract:** A user can refine a search over structured data by specifying that a label or an attribute value be used to further filter the results of a query.



WO 2007/046830 A3

Search Over Structured Data

Background of the Invention

[0002] Conventional search engines are capable of searching extremely large collections of information, such as the world wide web or very large databases. As the size of data collections to be searched grows, it is no longer enough to correctly return query results that match query terms entered by a user. Instead, it is desirable to provide a mechanism to help the user sort through the large amount of data returned from a search.

[0003] Several conventional search engines currently use various methods to organize the data returned in query result. The goal of such an organization method is to decide which query result will most interest the user. Conventional search engines generally use a variety of techniques to prioritize the results of a search, but these techniques are not ideal because they must make assumptions about the type of information for which the user is searching. For example, if the user enters "jobs" he might be searching for job postings, information of Steve Jobs, job statistics for a particular country, or any number of other items. Thus, when using a conventional search engine, a user would not enter just "jobs" as a query term. He would probably also enter additional query terms that narrowed the search. Unfortunately, he may also miss relevant listings that do not contain the narrowing terms.

[0004] Currently, it is difficult to search over different types of data that may or may not be stored on the world wide web. Conventional search engines usually operate on data from only a few sources. For example, web-based search engines traditionally allow a user to search pages on the world wide web. Web-search engines often have a “back-end” that indexes the collection of information in order to make it searchable. For example, web-based search engines periodically crawl the world wide web and create indices of the pages and sites crawled. Other search engines allow a user to search existing databases. Such search engines rely on a predetermined organization of the database. For example, if a database has known fields and attributes, the user can search within those attributes. For example, XML databases only accept well-formed XML inputs. If the data to be searched is not so-organized, XML databases are generally not able to accept the data or organize the data for search.

[0005] Other search engines allow a user to search databases or to search text documents having a flat organization. Such search engines must know about the organization of the database and the organization of the documents within it. The variety of locations and formats in which data are stored means that users must often search in multiple locations in multiple databases to find the information that they need.

[0006] It would be desirable for a collection of documents to be searchable via a web-based search engine and thus easily accessible to most people while, at the same time, containing a variety of types of documents and formats of data. Moreover, it would be desirable if the searchable collections of documents were organized in ways that could help users fine-tune their searches.

60412-3964

Summary of the Invention

[0007] The described embodiments of the present invention associate labels and attribute values with data items to be searched. Providers can associate attributes and labels with their data or attributes and labels can be added to existing data. One embodiment allows a content provider to attach his own custom labels and attributes to items or to use predefined labels and attributes. Providers can upload data using a user interface or a bulk upload mechanism. A user can refine a search by specifying that a label or an attribute value be used to further filter the results of a query.

10 **[0007a]** According to one particular aspect of the invention, there is provided a computer-implemented method comprising: extracting attributes from data items in a collection; receiving a search query from a user interacting with a search engine through a client device; receiving a selection of first data items from the collection of data items, the first data items satisfying the search query, and the
15 first data items collectively having first attributes that have been extracted from the first data items; determining an attribute frequency for each first attribute that has been extracted from the first data items, the attribute frequency being a respective frequency that previous users of the search engine have selected individual ones of the first attributes over a predetermined period of time to narrow past query
20 results; selecting, as a core group of attributes, the first attributes which have the highest respective attribute frequency, the core group of attributes including fewer than all of the first attributes; outputting to the client device a search engine results page comprising a respective reference to each of two or more of the first data items, and a respective representation of each attribute of the core group;
25 providing, in response to a user selection of a representation of one or more of the attributes of the core group a search window for receiving a query refinement; and receiving a selection of second data items satisfying the search query and the query refinement.

60412-3964

[0007b] There is also provided a computer readable medium encoded with a computer program product comprising instructions that, when executed, operate to cause a computer to perform operations comprising: extracting attributes from data items in a collection; receiving a search query from a user interacting with a search engine through a client device; receiving a selection of first data items from the collection of data items, the first data items satisfying the search query, and the first data items collectively having first attributes that have been extracted from the first data items; determining an attribute frequency for each first attribute that has been extracted from the first data items, the attribute frequency being a respective frequency that previous users of the search engine have selected individual ones of the first attributes over a predetermined period of time to narrow past query results; selecting, as a core group of attributes, the first attributes which have the highest respective attribute frequency, the core group of attributes including fewer than all of the first attributes; outputting to the client device a search engine results page comprising a respective reference to each of two or more of the first data items, and a respective representation of each attribute of the core group; providing, in response to a user selection of a representation of one or more of the attributes of the core group a search window for receiving a query refinement; and receiving a selection of second data items satisfying the search query and the query refinement.

[0007c] Another aspect of the invention provided a system comprising: one or more computers; and a computer-readable medium coupled to the one or more computers having instructions stored thereon which, when executed by the one or more computers, cause the one or more computers to perform operations comprising: extracting attributes from data items in a collection, receiving a search query from a user interacting with a search engine through a client device, receiving a selection of first data items from the collection of data items, the first data items satisfying the search query, and the first data items collectively having first attributes that have been extracted from the first data items, determining an attribute frequency for each first attribute that has been extracted from the first

60412-3964

data items, the attribute frequency being a respective frequency that previous users of the search engine have selected individual ones of the first attributes over a predetermined period of time to narrow past query results, selecting, as a core group of attributes, the first attributes which have the highest respective attribute frequency, the core group of attributes including fewer than all of the first attributes, outputting to the client device a search engine results page comprising a respective reference to each of two or more of the first data items, and a respective representation of each attribute of the core group; providing, in response to a user selection of a representation of one or more of the attributes of the core group a search window for receiving a query refinement, and receiving a selection of second data items satisfying the search query and the query refinement.

Brief Description of the Drawings

[0008] The teachings of the present invention can be readily understood by considering the following detailed description in conjunction with the accompanying drawings. Like reference numerals are used for like elements in the accompanying drawings.

[0009] Fig. 1(a) is a block diagram showing a data processing system in accordance with a preferred embodiment of the present invention.

[0010] Fig. 1(b) is a block diagram showing another data processing system in accordance with a preferred embodiment of the present invention.

[0011] Fig. 1(c) is an architecture diagram in accordance with a preferred embodiment of the invention.

[0012] Fig. 2(a) is a flowchart showing an overview of creation of a collection of data items searchable in accordance with a preferred embodiment of the present invention.

[0013] Fig. 2(b) is a flowchart showing an overview of searching the collection of documents and refining the search in accordance with a preferred embodiment of the present invention.

[0014] Fig. 3(a) is a flowchart showing a method of extracting labels and attributes from a collections of data items.

[0015] Fig. 3(b) is a flowchart showing a method of receiving a query term and displaying a query result.

[0016] Fig. 3(c) is a flowchart showing a method of determining which attributes to display for a given query result.

[0017] Fig. 3(d) is a flowchart showing a method of allowing the user to refine the displayed query result using labels and/or attribute values.

[0018] Fig. 3(e) shows a method performed periodically to determine whether any new, provider provided attributes should be added to the Core attributes for an information type.

[0019] Fig. 4(a) is an example screen shot of a search engine and a query term entered by a user.

[0020] Fig. 4(b) is an example screen shot showing query result from the query of Fig. 4(a) and also showing labels and attributes relating to the query result for the query term.

[0021] Figs. 4(c)-4(g) are example screen shots showing additional attributes and labels and how a user might narrow his search using attributes and/or labels.

[0022] Fig. 5(a) shows a data format used to store attributes and labels for a collection of searchable data.

- [0023] Fig. 5(b) shows an example of an attribute stored using the format of Fig. 5(a).
- [0024] Fig. 5(c) shows an example of a label stored using the format of Fig. 5(a).
- [0025] Fig. 5(d) shows an example data structure to map information types to their attributes.
- [0026] Fig. 5(e) shows an example of an information type mapped to some example attributes for that information type.
- [0027] Figs. 6(a)-6(e) are example screen shots showing a user interface allowing a provider to edit and enter data into the system.
- [0028] Fig. 7 is an example screen shot showing a user interface for registering a bulk upload file.
- [0029] Figs. 8(a)-8(d) show how a provider does a bulk upload of data and attribute values.
- [0030] The figures depict embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

Detailed Description of Embodiments

[0031] The following paragraphs describe various embodiments of a system to upload and search structured data in accordance with the present invention.

[0032] Fig. 1(a) is a block diagram 100 showing a data processing system in accordance with a preferred embodiment of the present invention. Fig. 1(a) includes a plurality of client data processing systems 110a...110n, a network 130, and a server data processing system 120. In the Figure, an example user data processing system 110a includes a processor 140, a browser 150, and a memory 160. User data processing system 100 or its components can be any appropriate data processing system including but not limited to a personal computer, a wired networked computer, a wireless network computer, a mobile phone or device containing a mobile phone, a handheld device, a thin client device, some combination of the above, and so on. Network 130 can be any network that allows communication between one or more of the user data processing systems 110 and the server data processing system 120. For example, network 130 can be but is not limited to the Internet, a LAN, and WAN, a wired network, a wireless network, a mobile phone network, a network transmitting text messages, some combination of the above.

[0033] In a preferred embodiment of the invention, user data processing system 110a includes browser software 150 in memory 160 that is executed by processor 140 to allow the user to communicate with server system 120. As described below in detail, such a browser 150 allows the user to communicate with server data processing system 120 to send query terms to the server data processing system 120 and to receive query results from the system 120. As further described below, browser 150 allows the user to receive labels and attributes associated with the query result and to use the labels and attributes to further define a query result. Although the embodiments discussed herein are browser-based, the invention is not limited to browser-based searching and any appropriate mechanism for communication between user 110 and sever 120 may be used without departing from the spirit and scope of the invention.

[0034] Some of all of the software and computer-executable instructions discussed herein are capable of being stored as a computer program product on a computer-readable medium, including but not limited to: a memory of a data processing system, a CD ROM, a flash memory, a floppy disk, or capable of being transmitted as signals over a network or between system components.

[0035] Server data processing system 120 includes a processor 170 that executes search and query engine software 185 to enable server system 120 to search a collection of

structured data 190 for a query term. (Search and query engine 185 is also called “search engine”). One example of structured data is fielded data, i.e., data items, each having one or more data fields (such as Name, address, status, etc).

[0036] Memory 180 also includes an attribute repository 195, which stores the attributes (and labels) for some or all of the data items in structured data 190. The repository is discussed below in connection with Fig. 5. Although repository 195 is shown as being a part of collection of structured data 190, repository 195 may also be separate from collection of data 190.

[0037] Search engine 185, repository 195, and collection of structured data 190 are all shown in Fig. 1(a) as being in a single memory 180, although a large search engine and a large collection of data may be stored in many ways, including but not limited to distributed data processing systems, cooperating data processing systems, networks data processing systems, and so on. Search engine 185 can be software, hardware, firmware, or any combination of these.

[0038] In a preferred embodiment, a query term is entered by a user via one or more of the plurality of user systems 110 and transmitted to server data processing system 120 via network 130. Details of methods used by server 120 to receive, index, and search the collections of data are discussed in detail herein.

[0039] Fig. 1(b) is a block diagram 111 showing another data processing system in accordance with a preferred embodiment of the present invention. In Fig. 1(b), the user stores a personal data collection 190 on his machine 110. It is contemplated that a personal search engine will access and organize this data to make it searchable by the user and possibly by other users over network 130. Such a system also would allow databases and other types of data collections to be added to the pool of searchable documents that are accessible by the central search engine.

[0040] In the embodiment of Fig. 1(b), data collection 190 is stored on a user's data processing system 110 or an enterprise server (not shown) and may be made available to a selected group of persons or individuals, such as only to the user, only to a smaller sub-set of users, or to all users who are aware of how to access data collection 190. In such a case, the ability to filter searches via attributes and labels, as described herein, can be part of a personal search engine 185 that runs locally on a computer or on a local network of computers. For example, the Google Desktop Search tool, available from Google, Inc. of Mountain View, CA, is a search tool that runs on a user's desktop and indexes data on his personal computer. An implementation of Google Desktop search incorporating the present invention would give

the user the ability to search databases and other types of data collections stored on or accessible from his desktop.

[0041] It would also give the user the ability to organize his data with useful attributes and labels. For example, a university library can make all of its online collection available to students, faculty, and alumni of the university. In such as case, the information would not be on a publicly available server, but would be stored in a server of the university and would be accessible and searchable only to those persons (and programs) permitted access by the university data provider. In the example, the university would also be able to control which providers had the ability to add to the data collection.

[0042] Fig. 1(c) is an architecture diagram 131 in accordance with a preferred embodiment of the invention. In the described embodiment, providers can use one or more of three ways of inputting data and attributes to the system. A provider-facing front end 132 (see, for example, Fig. 6(b)) allows a provider to enter data items and attributes using a user interface provided for that purpose. The provider can also perform a bulk upload 133 of data items (see, for example, Figs. 8(a)-8(d)). The provider can also upload 134 items from a specific URL (for example, using FTP). Search and Query engine 185 indexes items in the collection of data 190, preferably including the entered attributes and their values for the data items to produce an index of all data 137. Search engine 185 also allows a user to enter a query (see, for example, Fig. 4(a)). The system also includes an Application Program Interface (API) to allow software programs to query the data via search engine 185.

[0043] Fig. 2(a) is a flowchart 200 showing an overview of creation a collection of data items searchable in accordance with a preferred embodiment of the present invention. As is discussed below in connection with Figs. 6(a)-6(e) and Fig. 8(a)-8(d), server 120 receives 202 a collection of data items. This data can be received either as a result of a standard web crawl or can be provided by one or more providers who desire their data to become searchable. The received collection of data items is processed to extract labels, attributes, and attribute values as described below and those labels, attributes, and attribute values are associated with the various information types. In certain circumstances, a user will provide attribute names and/or attribute values for some or all of the entered data. As an example, a user may upload a database that he has created to hold a collection of medical journals. He may have specified attributes to these journals with values reflecting attribute names such as "Journal," "year of publication," "Journal Name." He may also enter zero or more labels for each journal such as "Medical," "Dental," "From Harvard," and so on. Labels are a special kind of attribute

(also called a valueless tag) that do not have values associated with them. Details of element 204 are discussed in connection with Fig. 3(a).

[0044] Fig. 2(b) is a flowchart 210 showing an overview of searching the collection of documents and refining the search in accordance with a preferred embodiment of the present invention. In a described embodiment, the user enters 212 one or more query terms (such as “cancer receptor” 402 in the screen shot 400 of Fig. 4(a)).

[0045] In certain embodiments, the user may also enter attribute names and values as part of a query typed into area 402. For example the user might type the following into area 402:

cancer receptor attr(JournalType: medical)

If the user knows that some items in the query result have an attribute named JournalType but that the attribute is not part of the Core set of attributes, and the user wants to return only medical journals.

[0046] The system determines 213 a query result as discussed in more detail below in connection with Fig. 3(b). In some embodiments, the query result is displayed 213 at this point. In other embodiments, the query result is not displayed yet, but instead the user is asked to further refine his search by selecting labels and/or attributes specific to the query term. As shown in, for example, Fig. 3(d), the user can refine his search 214 by specifying labels and attributes.

[0047] Fig. 3(a) is a flowchart 300 showing a method of extracting labels and attributes from a collections of data items. This method is part of the set-up process used to organize collections of data so that they can be searched.

[0048] Once the data items are received, for each data item having an information type, the system determines 304 labels and attributes for this information type. An attribute is a name/value pair having a name, such as “journal,” which then has one or more possible values of the names of journals.

[0049] In a preferred embodiment, attributes and labels are specified by a provider of data. Thus, determining attributes is merely a matter of identifying user-provided attributes and labels.

[0050] In certain cases, a provider of data does not specify attributes and labels for his items. For example, if the items are web pages located by a web crawler, the owners of the web pages do not have an opportunity to specify attributes or labels for their pages. Thus, in another preferred embodiment, labels and attributes are derived by software for a collection of data. Deriving labels and attributes may involve a purely automated process in which potential values for a predetermined list of labels and attributes are found within the data

collection by software. For example, in a listing of items for sale (e.g., Google's Froogle system) price amounts meeting predetermined criteria are assigned as values of a "Price" attribute for that item. In another preferred embodiment, software performs an interactive process with the provider in which the software proposes attribute/value pairs, which are then accepted or rejected by the provider. In another preferred embodiment, html tags are scanned and information discovered is used to derive attribute values for the pages having the tags. As an example, if a page contains an html comment:

[0051] <! Current price is at <http://www.todayspricesforbigco.com> %id=32423490 !>

[0052] The software would obtain a current price from the indicated URL and make it the value of a Price attribute for that web page.

[0053] Once attributes and labels have been associated 306 with data items, the data items are indexed 309 so that they can be searched. In a first preferred embodiment, attributes and labels and their values also are indexed, although in other preferred embodiments, they are searched separately or indexed separately.

[0054] Fig. 5(a) shows an example of a format 500 used to store labels and attributes in repository 195. Each item is associated with specific attributes and labels appropriate to its type. For example – A job posting may have the attributes, job function – product management, employer - ABC Corporation and job type - Professional. Attributes and labels in a preferred embodiment can have values of the following types:

BOOLEAN

INT

FLOAT

URL

STRING

LOCATION

DATE

DATE RANGE

[0055] Attributes and labels are indicated in storage by metatags as follows:

[0056] <start name>

[0057] name

[0058] </end name>

[0059] <start value>

[0060] value

[0061] </end value>

[0062] Thus, in a preferred embodiment, each attribute is a name/value pair, such as an attribute name of “journal” and a value for the “journal” attribute of “Journal of Inflammation.” (see Fig. 5(b)). Each label has only a name, such as “Medical” which would indicate that a particular journal is a medical journal (see Fig. 5(c)). In a preferred embodiment, the information type of a data item is also the name of one of its labels. Thus, a data item having an information type of “Events and Activities” would also have a label with the same name. That way, a user can search for data having a particular information type by specifying a label with the same name as the data item’s information type.

[0063] Fig. 5(d) shows an example data structure to map information types to their attributes. Thus, if an item in collection of data 190 has an information type of “Product,” the attributes of the item can be determined by accessing the data structure of Fig. 5(c), which includes the attributes and their attribute type for the information type “Product.”

[0064] As shown in Fig. 5(d), each information type has predefined attributes. The values of an attributes are of an attribute type. Fig. 5(e) shows some actual values. Thus, an information type of “Journal” has an attribute of “Journal name” with values of attribute type string and a label of “Medical” with null values. Such attributes would, for example, allow a user to search for a particular journal title or to search for all medical journals. Similarly, the information type of “Product” has an attribute of “NumAvail,” which indicates a number of a particular product that are available for sale and has an attribute type of integer. All attributes are optional. Providers may choose to populate any of the attributes suggested to them or create their own

[0065] Fig. 3(b) is a flowchart 310 showing a method of displaying a query result in response to a received query term or terms. In a preferred embodiment, a query result is determined by search engine 185. For example, a query of “cancer receptor” 402 (see Fig. 4(a)) might return 312 a query result 406 of items having attributes 404 such as those shown in Fig. 4(b). As mentioned earlier, some embodiments of the present invention determine but not display the query result 406 at this point.

[0066] Once a query result is determined for a query (and optionally displayed), at least some of the attribute names and labels for the query result are displayed 322. The data items in the data set 406 have certain information types. Attributes 404 that are initially displayed are some or all of the attributes for the information types of the data items in query result 406. The query result will have data items, each of which have different attributes. The attributes that show up on top of the query result are the attributes are most common in the query result

and the ones that have been clicked on or refined by searchers the most. E.g. Query “housing” has a lot of items with bedrooms and bathrooms as attributes and searchers have always refined by the attributes “bathrooms” and “bedrooms” for the query housing. So bedrooms and bathrooms should show up on the top line above the search results

[0067] Fig. 4(b) shows query result 406 and a plurality of attribute and label names 404 (“journal,” “pubmed,” “news source,” “authors”). The numbers after each attribute indicate the number of items in the query result 406 that have the attribute associated with it. For example, in Fig. 4(b), query result 406 includes 2050 items having an associated “journal” attribute/label. Thus, the number and identity of the attributes shown with a particular query result is query-dependent, and is further dependent on the attributes and labels later chosen to narrow the search.

[0068] Fig. 3(c) is a flowchart 340 showing a method of determining which attributes to display for a given query result 406. When an end-user performs a search, the q most relevant results are determined 341 by search engine 185 and the n most popular attributes are determined 342 for the q most relevant results. For the top n attribute names, the system determines 344 the top m attribute/label values. It then calculates 348 histograms, or offer counts, by counting the number of matching offers in the set of relevant results. The values q, n, and m are all configurable. Example values, which are not to be taken in a limiting sense are: q-1,000 – 100,000 K (q can also be set to ALL results that match a particular query term.) N is in the range of 100s and M is in the range of 20-100.

[0069] In a preferred embodiment, the attributes are normalized 346 before histograms are determined. In certain implementations, a certain amount of data cleanup and normalization is done when the data is initially stored in data collection 190. In the described embodiment, data normalization is done on the fly based on the query term being searched (e.g., when the query term is “autos” it makes sense to normalize all “brand” attributes to “make”, however if the query is “handbag” it makes sense to normalize all make attributes to “brand”) Other embodiments may do more normalization at the time data is received into collection of data 190. Data normalization is accomplished in a preferred embodiment by:

1. Stemming - For example, restaurant = restaurants.
2. Abbreviations - For example, sz = size.
3. Units equivalence - For example, weight = ounces, lbs, etc.
4. Attempted spelling correction

[0070] Stemming is particularly useful in systems where providers specify their own attributes names, allowing variations and misspellings to creep into data collection 190.

Stemming, for example, allows a user to filter by attribute names of “Journal,” “journal,” “Journals” and so on with a single selection of the stemmed attribute “Journals.”

[0071] In certain preferred embodiments, attributes added by providers are type checked. For example, URL, DateTime, Number, String, Location, Boolean attributes are checked to see if they are valid values. Some embodiments ping each URL value to see if it is active, although this is optional for various implementations. For a preferred embodiment, locations are Geocoded so that they can be referenced by an online mapping service such as, for example, GoogleMaps. In certain embodiments, attributes of “location” that cannot be geocoded are considered invalid.

Once popular attributes and labels are determined and displayed 322 (Fig. 3(b)), the user is allowed to specify 324 one or more of the displayed labels and attribute values for the query result (see Fig. 3(d)).

[0072] Fig. 4(c) shows an example in which a user has selected the attribute “journal” from Fig. 4(b) and is preparing to enter in field 408 the names of journals to which he wishes to confine his search. Notice that the query term 402 is now “cancer receptor filter: journal.” The attribute “journal” has disappeared from the listed Core attributes 404.

[0073] Similarly, in Fig. 4(d), the user selects a second attribute “year” 410 and enters a year or range or years over which he wishes to search in the specified journals query term. The attribute year is of an attribute type “range of years”. Notice that the query term 402 is now “cancer receptor filter: journal filter: year.” The attribute “year” has disappeared from the listed attributes 404. If the user selects the GO button 411, the search is performed again using the selected attributes as filters and a display such as that of Fig. 4(e) is displayed. Thus, the user can select one or more popular attributes for displayed query result and can filter an initial search in accordance with the displayed attributes (or labels). If the user leaves an attribute value blank, all attribute values are matched. For example, if the user selects the attribute Journal but does not enter a journal name, all data items with an attribute of journal (and similarly named attributes) are selected as possible candidates for the query result. Data items not having an attribute of Journal are not selected for the query result.

[0074] Fig. 4(d) shows an example where a user has selected more than one attribute or label to narrow the search. In the described embodiment, multiple labels and attributes are selected by clicking on multiple labels and attributes of the attributes and labels 404. Other preferred embodiments allow labels and attributes to be entered into the search window 402. For example, if an attribute Price exists, the user might type the following as a query term:

[0075] Attribute(Price: \$150)

[0076] This query would locate data items in the current query result having an attribute of Price and an attribute value of \$150.

[0077] As another example, the user might type:

[0078] Attribute(Price: \$150) AND Label(SmallerThanABreadBox)

[0079] This query would locate data items in the current query result having an attribute of Price, and Attribute value of \$150, and a label of SmallerThanABreadBox. Other preferred embodiments would use other appropriate user interface elements to allow a user to logically combine attributes and labels.

[0080] Fig. 4(e) shows a query result limited to particular journals of a particular year or range of years as specified in Fig. 4(d). The user is allowed to decide whether he wants to continue to search within journals 412 or whether to search the entire collection of data items (e.g., "Search all of Googlebase") 413. In the example, the user is offered the choice of several labels in area 414 ("biotechnology," "medical," and "photography," which respectively are associated with 30, 15, and 6 items in the query result 406'). In the example, the user is further offered a choice to specify values for the attributes in area 416: Date, author, pubmed, citation. The user is also offered the option of sorting 416 the query result 406' by relevance, date attribute, or any of the attributes that the user has defined by. (e.g. price, location, etc.).

[0081] In Fig. 4(f), the user has selected attribute "Date" from area 414 of Fig. 4(e) and is given a chance to enter a date 420. When the user selects the drop-down operator "between" he is given the opportunity to select a date range (as shown). The attribute "Date" has disappeared from the listed attributes 418. In this example, the attribute of "Author" has disappeared from attributes 414. Attributes disappear if they are no longer relevant to the query and the query result. The fact that the user has not filtered by the journal now assumes that he is looking only at the restricted item set. The user selects the Go button to perform a search and the author term reappears.

[0082] Fig. 4(g) shows the user specifying an author name 422. As the query is being refined new attributes and labels show up since the attributes and labels are based on the query result and the query result constantly changes. When the user presses the Go button 423, another search is performed, further filtering the query result to reflect the attributes and attribute values specified by the user.

[0083] The following paragraphs discuss access of attribute repository 195 during a search or during narrowing of a search using attributes and labels.

[0084] Queries and indexing that reference repository 195 preferably support the following operators:

Number - Is, Between, Greater Than, Less Than, Number Range

suggestions

String - Is, Has

Date- Range, Before, After, Is

Location - Within

[0085] Repository 195 can be queried in at least the following ways:

-Give me all items that match a particular attribute name-type pair

-Sort these items based on the value of the attribute-value

-Sorts for the following types of attributes are supported

-DateTime

-Number - Int, Float

-String

-Location - Distance from user entered location

[0086] This query ability allows the user to enter the following types of attribute queries:

-Give me all items that have a particular name-type pair

-Given these items sort it by attribute value (e.g. Give me all items that have event_date and give it to me sorted in an ascending order)

-Give me all items are in-between value1 and value2 for a particular name-type attribute

[0087] Examples

-Give me all items that have cooking_type as an attribute and have values between 15 and 30 where the unit is minutes

-Give me all items that have size as an attribute and have values 1 and 15 with no unit

-Give me all items that have an event_date and have a value less than today

-Give me all items that have a publication_date and have a value in the year of 1925

[0088] The following operators are supported

-For number - int, float

-Less than

-Greater than

-Between

[0089] For dateTime -

-Is

-Before

-After

-Between

-Scoring of Items

[0090] Currently there are 2 major signals by which items are scored

-Query Dependent Rank - Mainly IR score

-Query Independent Rank - Mixture of page rank and item rank

[0091] Page rank is the provider's website page rank. Page rank does not exist in cases where the items are hosted in a collection of data 190 and/or items are not linked or connected to other items.

[0092] Item rank can be determined by a number of factors. The two main signals are

-Provider specific signals. (e.g. rating)

-Offer specific signals (.e.g. Length of desc, number of attributes, labels, pictures etc.)

-Item Rank can be defined by the following signals

-Length of Desc

-Length of Title

-Number of Labels

-Number of Attributes

-Pictures

-Number of times offer has been reported as spam

-Rating of the provider

-Recency of the offer

Items are scored as - Query Dependent Rank * Query Independent Rank

For default sorts the Rank is the default sort.

[0093] In a preferred embodiment, certain parameters can be set in the system. These parameters include a maximum number of items per provider. This prevents crowding of the page by a specific provider

[0094] When the user selects attributes and/or labels to narrow a search, the system searches labels, titles, description and attribute values. Attribute names should also be searchable as complete names. Phrases are weighted heavily compared to words that occur

far away. Labels are weighted more heavily than titles, which are weighted more heavily than descriptions. Attribute values are weighted the same as labels. Merchant crowding by each provider may be turned on or off by the user to regulate whether a page number of items from an individual provider are or are not displayed as a result of a search. Depending on the search performed, merchant crowding may or may not be desirable.

In a preferred embodiment, the system defines a structure of a particular type of new item based on the attributes associated with other items of the same or similar type (e.g., If most items of information type “Jobs” have attributes of Job function, Job type and Employer than the common attribute structure for the data item of information type “job” will default to be job type, employer and job function). Searchers and other programs can query the data-set with queries such as “Give me all jobs whose employer is ABC Corporation and whose job-type is product management).

[0095] It will be understood that, although the examples described herein refer to a human user, other embodiments of the present invention may be designed to operate with a non-human user such as an artificial intelligence software program or with an entity communicating over the web that could be either human or non-human. If the non-human user is a software program, it may not be necessary to display the results and attributes as described herein. Instead, such an implementation might merely communicate the potential attributes that could be used to narrow the query result. In such an embodiment, a larger option of attributes can be displayed since non-human artificial intelligences are not overwhelmed by a large number of attributes from which to choose. In such an embodiment, elements of the method such as determining histograms may not be needed or they might be used only to rank attribute choices and not to limit a number of available attribute choices.

[0096] It will be understood that periodically, the Core attributes for the various information types in structured data 190 may need to be updated. As data is added to the collection of structured data, certain attributes may become popular that were not initially popular. For example, a “Season” attribute having an attribution type of integer might specify which season of a television show a cast picture is from may not have been initially contemplated by the initial core attributes for the information type “TV shows” but it may become popular as more and more cast pictures are added to the collection of data. In some embodiment, core attributes also are auto-updated based on popularity and seasonality and after passing through a spam filter.

[0097] Fig. 3(e) shows a method 350 performed periodically to determine whether any new, provider-provided attributes should be promoted to the Core attributes for an

information type. The core group of attributes for an item information type is the attributes that are automatically offered whenever a provider adds a new item of the information type. In a preferred embodiment, only Core attributes are offered to decrease the possibility that a provider will spam attributes in order to force his way into the displayed attributes. For each information type, the method looks at the most popular user-added attributes for that information type 322 and promotes the most popular attributes to Core Attributes for that information type.

[0098] “Most popular,” as used to decide which attributes to promote to Core attributes, is defined differently for different embodiments. For example, most popular can be the attribute not in the Core Attributes that is most-often selected 352 by users over a predetermined period of time, such as a week or month, for example. As another example, most popular can be the attribute not in the Core Attributes that has data items appear most often in query result over a predetermined period of time. As another example, most popular can be the attribute not in the Core Attributes that appears in a largest number of providers’ data over a predetermined period of time. Most popular can be determined in any appropriate way as long as it causes attributes that will be useful in narrowing a search to be added to the Core Attributes.

[0099] For example, providers may have started adding an attribute of “blogged” for an item information type of article to indicate that the article has been mentioned in a blog. Such an attribute would have a URL attribute type, indicating the URL of the blog where the item was mentioned. If a threshold number 354 of unique providers or users use a particular new attribute for an information type, the attribute is added 356 to the Core group of attributes for that information type. In a preferred embodiment, the Threshold value will be based on the total number of providers using the system. It will start with something as low as 2-3 and will be increased to larger numbers. A similar method is performed for labels to add popular labels to a core set of labels. In certain preferred embodiments, promoted attributes will be sanity checked by a human being or appropriate software or hardware implemented method.

[00100] The previous paragraphs have generally discussed ways to search and update data entered into a collection of structured data 190. The following paragraphs discuss ways that providers can enter data or add data to a collection of structured data 190. Providers can also, in certain preferred embodiments, specify new attributes for their data.

[00101] Figs. 6(a)-6(e) are example screen shots showing how a provider can edit items in a data collection. A provider is anyone who adds or is capable of adding content to the

collection of data 190. In the described embodiment, collection of data 190 is data owned by one or more providers, such as an individual, a non-profit organization, or a company. The embodiment allows such providers to set-up and populate their own collections of structured data (e.g., databases) via the web and to make those collections searchable via the web or similar network. It is contemplated that providers will be willing to store data in a central repository, either for a fee or in exchange for their permission to allow the data to be searched by others. In such a situation, the data collection can be searched via a web or network based browser, such as the Google browser or Google desktop search engine, in a version that contains some or all of the functionality described herein.

[00102] Figs. 6(a)-6(e) are example screen shots showing a user interface allowing a provider to edit and enter data into the system.

[00103] Fig. 6(a) shows a user interface 600 that allows a provider to view and edit data items in collection of data 190. The user interface can also be used to add items to collection of data 190. An area 602 contains a partial listing of items in the collection of data 190. In the example here, this listing includes item title 601, an item type (also called an information type) 605, Status 603, an Expiration date, a number of impressions (the number of times an item has been displayed), a number of clicks on the object, and the click-through rate, the number of times an item was clicked on in search results. In the example, a subset of all items in the data collection are shown in area 602, but a provider can also search either his personal data collection 620 or search the entire data collection 622. The provider can also view inactive items 616 or upload bulk files 618. Each data item has an associated “edit” link 619. In a preferred embodiment, a provider can only edit his own data items. An area 604 allows the provider to display a selection device such as a dropdown menu showing existing information types (Events and Activities, Housing, etc). If the provider selects an information type, he can add a description of the information type in area 606 for his data.

[00104] Fig. 6(b) shows a user interface that allows a provider to view and edit 610 data items in collection of data 190. The items have an information type of “News and articles.” If the provider had selected a data item in area 602 of Fig. 6(a), that item’s information would be displayed in the fields of area 611. In the example, however, the provider did not select an item, so the provider is free to enter a new data item. In the example, an information type of “News and Articles” 610 contains the following fields: Title, Pictures, Description and a link 614 (e.g., a URL) to be displayed in a query result.

[00105] The user interface of Fig. 6(b) also allows the provider to edit the attributes and labels of the item. Note that, while each information type has associated attributes, not all

data items of a particular type have values for all possible attributes for that information type. In the example, as shown by reference numeral 612, the provider has indicated that a quantity of “1” of the item is available or exists. No value is specified for the Author or News Source attributes for this item. Each of those attributes have an attribute type of “text.” The provider is free to add value for the attributes of individual data items. The provider can also add an attribute using area 613. Here, the provider can add an attribute name and an attribute value.

[00106] The provider can provide attribute values relating to contact information in area 618. The provider can provide attribute values relating to location information in area 619.

[00107] The provider can add labels to the item in area 619. In certain embodiments, the information type is a default attribute name. Here, the information type is News and Articles and this is also a label.

[00108] Fig. 6(c) shows the user interface of Fig. 6(b) that allows a provider to view and edit 610 data items in collection of data 190. In the example, the provider can add a name and value for a new provider-defined attribute 613. While the default attribute type is “text” the provider can choose another attribute type, such as number unit, number, data range, large text, URL, Boolean, and location.

[00109] Fig. 6(d) shows a user interface that allows a provider to view and edit 610 data items in collection of data 190. The items have an information type of “Products” 630. If the provider had selected a data item in area 602 of Fig. 6(a), that item’s information would be displayed in the fields of area 611. In the example, however, the provider did not select an item, so the provider is free to enter a new item using user interface 630. In the example, an information type of “Products” contains the following fields: Title, Pictures, Description and a link 634 (e.g., a URL) to be displayed in a query result.

[00110] The user interface of Fig. 6(d) also allows the provider to edit the attributes and labels of the item. Note that, while each information type has associated attributes, not all data items of a particular information type have values for all possible attributes for that information type. In the example, as shown by reference numeral 632, the provider has indicated that a Price of \$150 per item (as opposed to per pound or per dozen for example). A quantity of “1” is specified. Price type is the type of price the provider is setting (e.g. Best offer, negotiable, fixed etc). No values are specified for Price option, Brand, Condition, and Product Type for this item. Each of those attributes have an attribute type of “text.” In this embodiment, the provider can change the attribute type for those attributes that he specified. The provider is free to add values for the attributes of individual data items. The provider can

also add an attribute using area 613. Here, the provider can add an attribute name and an attribute value.

[00111] In this embodiment, attributes that a provider adds are added to all of his items of the current information type. Here, for example, all of the provider's items of type "Products" are given the newly added attribute 613 once it is defined. The values for each item are normally added individually. Certain embodiments also allow a provider to specify a value for all of his items of a specified information type. As discussed above, it is possible for the new attribute to graduate to the Core set of attributes. In other embodiments, new attributes are not always added to all items of the information type. In other embodiments, providers can agree that a defined group of providers will all have the same attributes, so that when one provider adds an attribute, the others in the group will also have the same attribute.

[00112] The provider can provide attribute values relating to contact information in area 618. The provider can provide attribute values relating to location information in area 619. The provider can provide attribute values relating to Payment methods in area 638.

[00113] The provider can add labels to the item in area 616. In certain embodiments, the information type is a default attribute name. Here, the information type is Products and this is also a label. In this embodiment, labels that a provider adds are not added to all of his items of the current type (except for labels that are the information type). As discussed above, it is possible for a new label to graduate to the Core set of labels. In other embodiments, new labels are always added to all items of the information type.

[00114] Fig. 6(e) shows the user interface of Fig. 6(d) that allows a provider to view and edit 630 data items in collection of data 190. In this example, Contacts, Payments, and Location are all attributes of the Product information type. They are attributes having a complex type (not just integers or simple strings). In the example, the provider can add values relating to contacts 618 for items of information type "Products." Here, the provider specifies some or all of Nickname, phone number, email address (potential values taken from a database of provider information, not shown). In the example, the provider can add values relating to Payments 638 for items of information type "Products." Here, the provider specifies some or all of Payment Method and Notes. In the example, the provider can add values relating to Locations 619 for items of information type "Products." Here, the provider specifies some or all of Text notes (e.g, "Fremont, CA"). In this embodiment, there are also check boxes to indicate whether customers can pick up from this location and delivery radius.

[00115] In the example, Contact, Payment, and Location values are entered separately for each item. Values that a provider adds are not added to all of his items of the current

information type. Here, for example, not all of the provider's items of information type "Products" are given the Contact, Payment, and Location values shown in Fig. 6(e). The values for each item are normally added individually. Certain embodiments also allow a provider to specify values for all of his items of a specified information type. For example, payment information may be the same for all "Products" of a provider.

[00116] Promoters can either enter items through the UI of Fig. 6 or via a bulk upload method shown in Figs. 7 and 8.

[00117] Fig. 7 is an example screen shot 700 showing a user interface for registering a bulk upload file. The bulk upload file is used to create or add to collection of data 190. In this example, a flat file of items that are all of the same information type are to be added. In the example, the File Name 712 is "local inventory". The provider chooses an data type 714 that is a predefined information type or a custom information type. The provider chooses a language 716 for text strings in the data. When provider selects button "Register bulk upload file" 718, the file having file name 712 is registered and the provider will then be allowed to upload the file. A provider can upload files using a web-based uploading interface or using another mechanism such as FTP (File Transfer Protocol) or an RSS.

[00118] Fig. 8(a) shows a format 801 for a tab-delimited file to be bulk uploaded. The following are format requirements for bulk upload files:

- Tab delimited plain text.
- First line of the file is the header - must contain attribute names (described below), separated by tabs.
- One item per line; each attribute should be separated by a tab.
- Do not have trailing tabs at the end of lines.
- File must be saved in LATIN1 or UTF-8 encoding. ASCII is also acceptable, as it is a subset of LATIN1.
- Link and image URLs must be fully qualified. That is, they must include the http:// portion, for example: <http://www.example.com/image.gif>
- Tabs, carriage returns, or new line characters - If any of these appear in an attribute, we will not be able to display that item.
- HTML tags, comments, and escape sequences – No html is removed from a bulk upload, but for best appearance, no HTML should be included.

[00119] In a preferred embodiment, data items are a part of the uploaded file that also contains attributes. In other preferred embodiments, data items and attributes are uploaded in

separate files that are constructed so that it is clear which attribute values belong with which data items.

[00120] Fig. 8(b) is a flowchart 800 of an example method used by a provider to create a bulk upload file. A provider can be a human being, or hardware or software.

[00121] Element 802: Open a new file in a spreadsheet program

The described method uses a spreadsheet program, such as Microsoft Excel, to create a bulk upload file. Using a spreadsheet program like Microsoft Excel makes it easy to create a bulk upload and convert it to the proper format. Other methods can be used that result in an appropriately formatted file.

[00122] Element 804: Create a header row

As an example, the header row for a product bulk upload might look like row 832 in Fig. 8(c). Specify each of the columns in the bulk upload according to the information type of item that the provider would like to submit (see 714 of Fig. 7). On the first row of the spreadsheet 832, enter the name of each of the attributes that the provider would like to include to describe his items. This is the header row. The content of the header row will depend on the information type of information submitted, and whether the provider is sending a defined information type, or one that he created himself.

[00123] Custom Information Types:

Bulk uploads can be used to submit any type of information. If a provider is sending his own information type, he can use any combination of predefined attributes. In a preferred embodiment, it is strongly recommended that providers use the predefined attributes. A provider can also include an unlimited number of custom attributes: A provider should pick a set of attributes that best describes his items

[00124] Defined Information Types:

[00125] A provider can send a bulk upload for one of the defined information types. It is strongly recommended that that a provider include them in his bulk upload. They allow more accurately matching of items to search queries. The more information a provider gives, the easier it will be for users to locate items. In a preferred embodiment, a provider must include these recommended attributes to enable a provider's items to appear in a significant portion of searches done.

[00126] Element 806: Enter item information

On each row 834, a provider enters information for an item in his data collection. Each piece of information should reflect the header of the column it is in. (For example, a product's price

should go under the "price" header). Each row includes only include one item per row. See Fig. 8(c).

[00127] Element 808: Convert bulk upload to tab-delimited plain text

Convert the spreadsheet into a tab-delimited text (.txt) file using the filename previously registered (see Fig. 7). After the provider has entered all items in the spreadsheet, he saves the spreadsheet in the tab-delimited text (.txt) format. The registered filename can be reused for subsequent uploads. If an uploaded file has an unregistered name, the items in the file will not be added to the collection of data 190. In a preferred embodiment, an updated bulk upload must be sent at least once every 30 days to ensure that the items remain in collection of data 190.

[00128] Element 810: Upload File

Fig. 8(d) shows a user interface 840 to upload a file.

[00129] Element 812: Check the bulk upload for errors

After a provider has sent a bulk upload, he can see the bulk upload's status by logging in to a central web site. If the outcome is listed as 'Success', the bulk upload does not need to be altered. Otherwise, the provider can click on the bulk upload's filename to see information on how to correct the error(s).

[00130] After a bulk upload is uploaded, the file will be processed to add the items, attributes, and labels to data collection 190 and the data structure of Fig. 5. Once an upload has been approved, any future updates with the same filename will be processed automatically.

[00131] Although the present invention has been described above with respect to several embodiments, various modifications can be made within the scope of the present invention. For example, certain preferred embodiments include methods and systems for detecting invalid or "spammy" attributes and labels. It is undesirable for a provider to add attributes to his data that will allow the data items to come to the top of a search. Some methods that are used to avoid such attributes include blacklisting, specific histograms distributions, and so on.

[00132] In other preferred embodiments the displayed top attributes and labels are determined based not just on popularity of the attribute key-type tuples and labels but on distribution of values (more discrete the distribution the better and the more the skew the better. e.g. 5 popular values for an attribute are better than 50 values distributed evenly. Example if color is an attribute and we see Red, Blue and Green as the top colors than it would be a good attribute to refine by. On the other hand having 100 values to color each of which occur three times is not so helpful.

[00133] Another preferred embodiment performs sophisticated confidence scores based on the number of providers who use an attribute, the item rank/offer rank of each offer.

[00134] Another preferred embodiment uses click signals from users to determine which attributes to display to the user. Attributes and labels are scored by something defined as popularity rank:

$$PR = \text{Popularity in the Query result} * \text{CTR for that particular query}$$

[00135] In another preferred embodiment, if users ALWAYS 2 attribute restricts for a particular query (e.g. Ipod for the 90% case is always restricted on price and location, the system restricts by price and location going forward when users type ipod) show those restricts already applied to the query result.

[00136] Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

CLAIMS:

1. A computer-implemented method comprising:

extracting attributes from data items in a collection;

5 receiving a search query from a user interacting with a search engine through a client device;

receiving a selection of first data items from the collection of data items, the first data items satisfying the search query, and the first data items collectively having first attributes that have been extracted from the first data items;

10 determining an attribute frequency for each first attribute that has been extracted from the first data items, the attribute frequency being a respective frequency that previous users of the search engine have selected individual ones of the first attributes over a predetermined period of time to narrow past query results;

15 selecting, as a core group of attributes, the first attributes which have the highest respective attribute frequency, the core group of attributes including fewer than all of the first attributes;

outputting to the client device a search engine results page comprising a respective reference to each of two or more of the first data items, and a respective representation of each attribute of the core group;

20 providing, in response to a user selection of a representation of one or more of the attributes of the core group a search window for receiving a query refinement; and

receiving a selection of second data items satisfying the search query and the query refinement.

2. The method of claim 1, further comprising:

receiving the user selection of the representation of fewer than all of the attributes of the core group; and

outputting to the client device a second search engine results page comprising a respective reference to each of two or more of the second data items.

5 3. The method of claim 1, further comprising performing a web crawl of the collection, the collection storing the data items and the associated attributes describing an information type of each data item.

4. A computer readable medium encoded with a computer program product comprising instructions that, when executed, operate to cause a computer to
10 perform operations comprising:

extracting attributes from data items in a collection;

receiving a search query from a user interacting with a search engine through a client device;

receiving a selection of first data items from the collection of data items,
15 the first data items satisfying the search query, and the first data items collectively having first attributes that have been extracted from the first data items;

determining an attribute frequency for each first attribute that has been extracted from the first data items, the attribute frequency being a respective frequency that previous users of the search engine have selected individual ones of
20 the first attributes over a predetermined period of time to narrow past query results;

selecting, as a core group of attributes, the first attributes which have the highest respective attribute frequency, the core group of attributes including fewer than all of the first attributes;

outputting to the client device a search engine results page comprising a respective reference to each of two or more of the first data items, and a respective representation of each attribute of the core group;

5 providing, in response to a user selection of a representation of one or more of the attributes of the core group a search window for receiving a query refinement; and

receiving a selection of second data items satisfying the search query and the query refinement.

5. The computer readable medium of claim 4, the operations further
10 comprising:

receiving the user selection of the representation of fewer than all of the attributes of the core group; and

outputting to the client device a second search engine results page comprising a respective reference to each of two or more of the second data items.

15 6. A system comprising:

one or more computers; and

a computer-readable medium coupled to the one or more computers having instructions stored thereon which, when executed by the one or more computers, cause the one or more computers to perform operations comprising:

20 extracting attributes from data items in a collection,

receiving a search query from a user interacting with a search engine through a client device,

receiving a selection of first data items from the collection of data items, the first data items satisfying the search query, and the first data items collectively having first attributes that have been extracted from the first data items,

determining an attribute frequency for each first attribute that has been
 5 extracted from the first data items, the attribute frequency being a respective frequency that previous users of the search engine have selected individual ones of the first attributes over a predetermined period of time to narrow past query results,

selecting, as a core group of attributes, the first attributes which have the highest respective attribute frequency, the core group of attributes including fewer
 10 than all of the first attributes,

outputting to the client device a search engine results page comprising a respective reference to each of two or more of the first data items, and a respective representation of each attribute of the core group;

providing, in response to a user selection of a representation of one or
 15 more of the attributes of the core group a search window for receiving a query refinement, and

receiving a selection of second data items satisfying the search query and the query refinement.

7. The system of claim 6, further comprising performing a web crawl of the
 20 collection, the collection storing the data items and the associated attributes describing an information type of each data item.

8. The method of claim 1, further comprising:

receiving the collection of data items and the attributes and attributes of the data items from a single data provider.

25 9. The method of claim 1, wherein extracting attributes further comprises:

determining an information type of a data item;

accessing a data structure that maps information types to attributes;

determining, based on applying the information type to the data structure, the attributes associated with the data item.

5 10. The method of claim 1, wherein extracting attributes further comprises:
extracting the attributes before any searching occurs on the data items.

11. The method of claim 1, further comprising:

generating a user interface for allowing a provider of the data items to edit the attributes associated with the data items; and

10 receiving information through the user interface, the information changing an attribute associated with one or more of the data items.

12. The computer readable medium of claim 4, wherein the operations further comprise:

15 receiving the collection of data items and the attributes and attributes of the data items from a single data provider.

13. The computer readable medium of claim 4, wherein extracting attributes further comprises:

determining an information type of a data item;

accessing a data structure that maps information types to attributes;

20 determining, based on applying the information type to the data structure, the attributes associated with the data item.

14. The computer readable medium of claim 4, wherein extracting attributes further comprises:

extracting the attributes before any searching occurs on the data items.

15. The computer readable medium of claim 4, further comprising:

5 generating a user interface for allowing a provider of the data items to edit the attributes associated with the data items; and

receiving information through the user interface, the information changing an attribute associated with one or more of the data items.

16. The system of claim 6, wherein the operations further comprise:

10 receiving the collection of data items and the attributes and attributes of the data items from a single data provider.

17. The system of claim 6, wherein extracting attributes further comprises:

determining an information type of a data item;

accessing a data structure that maps information types to attributes;

15 determining, based on applying the information type to the data structure, the attributes associated with the data item.

18. The system of claim 6, wherein extracting attributes further comprises:

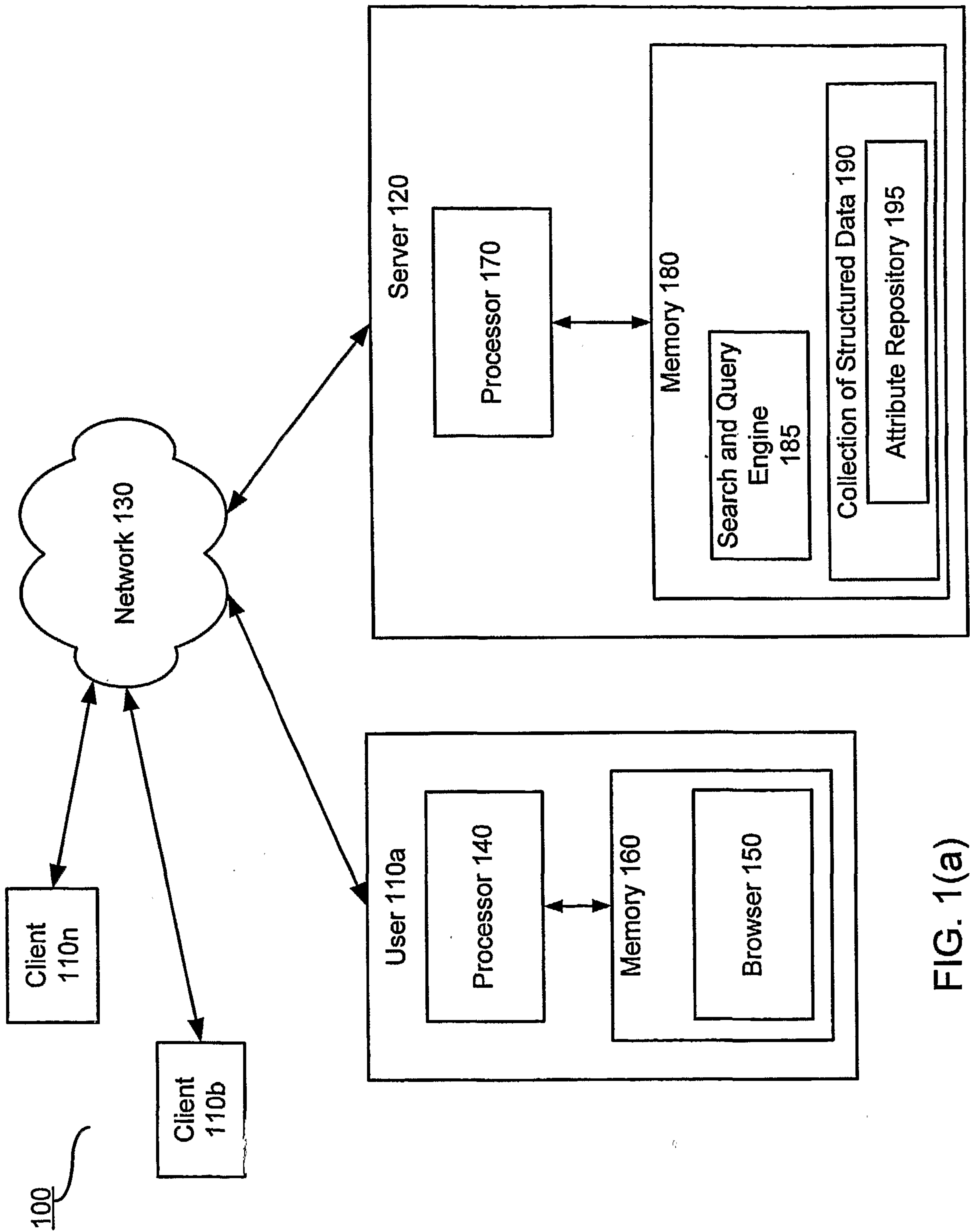
extracting the attributes before any searching occurs on the data items.

19. The system of claim 6, further comprising:

20 generating a user interface for allowing a provider of the data items to edit the attributes associated with the data items; and

receiving information through the user interface, the information changing an attribute associated with one or more of the data items.

1/27



2/27

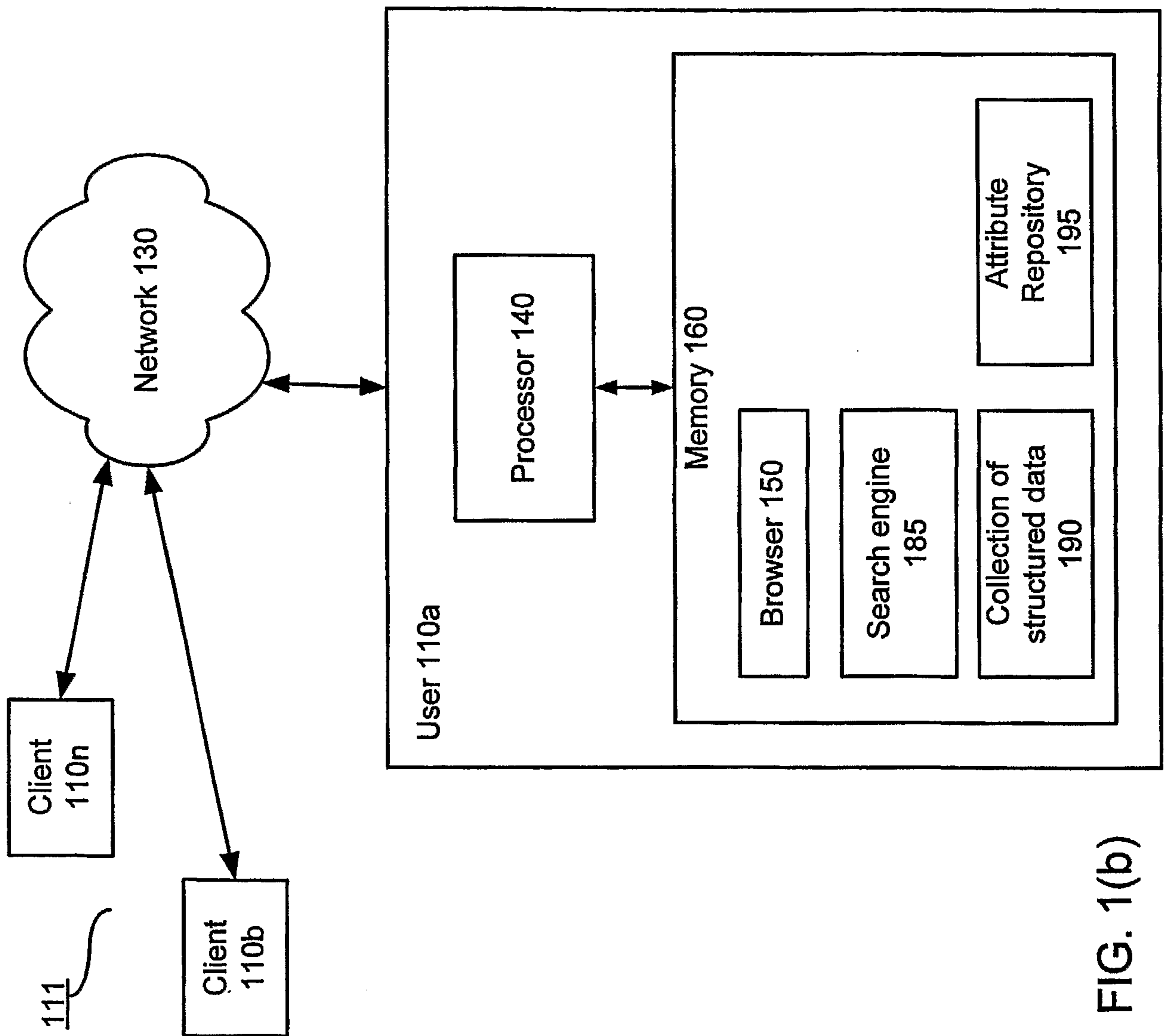


FIG. 1(b)

3/27

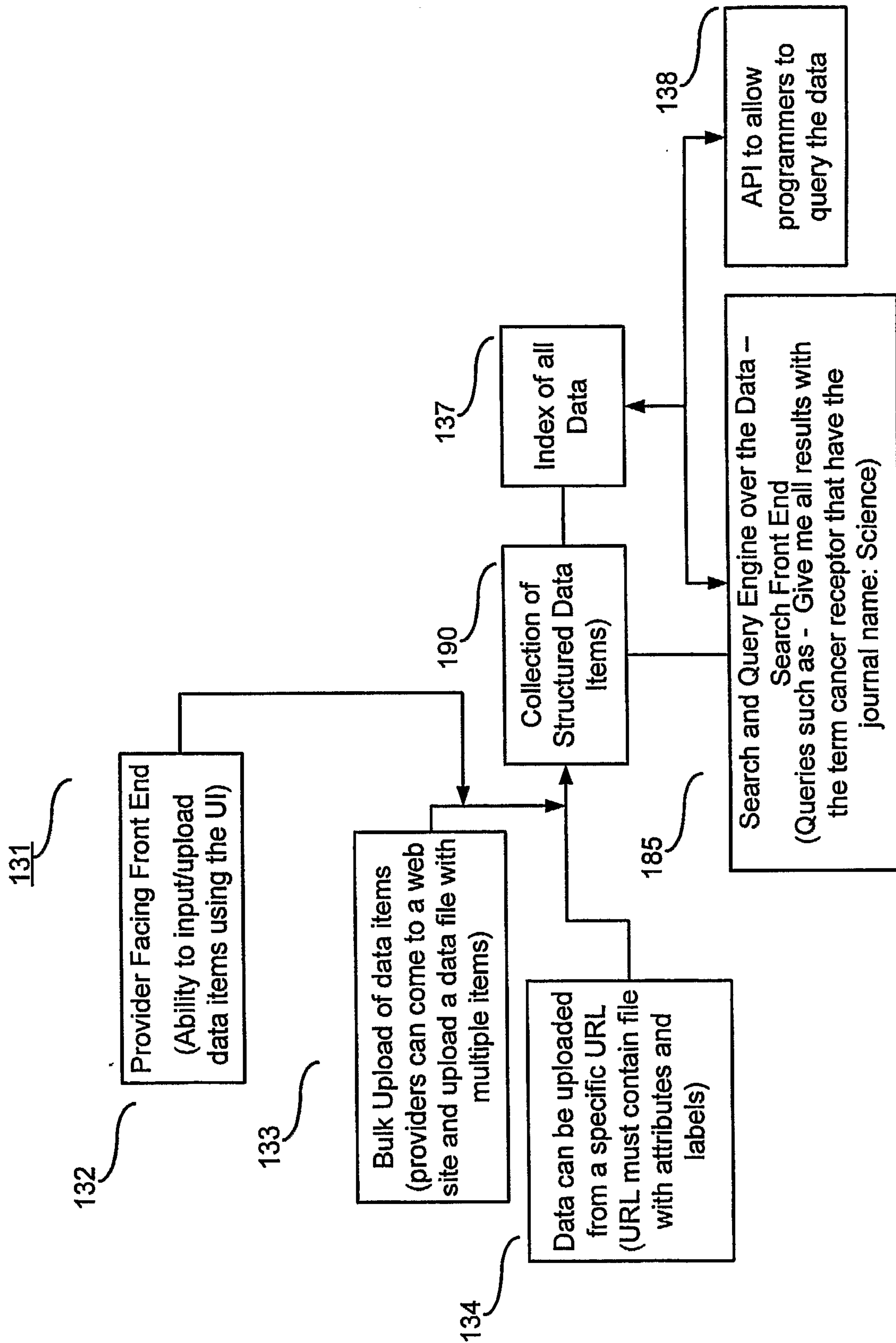


Fig. 1(c)

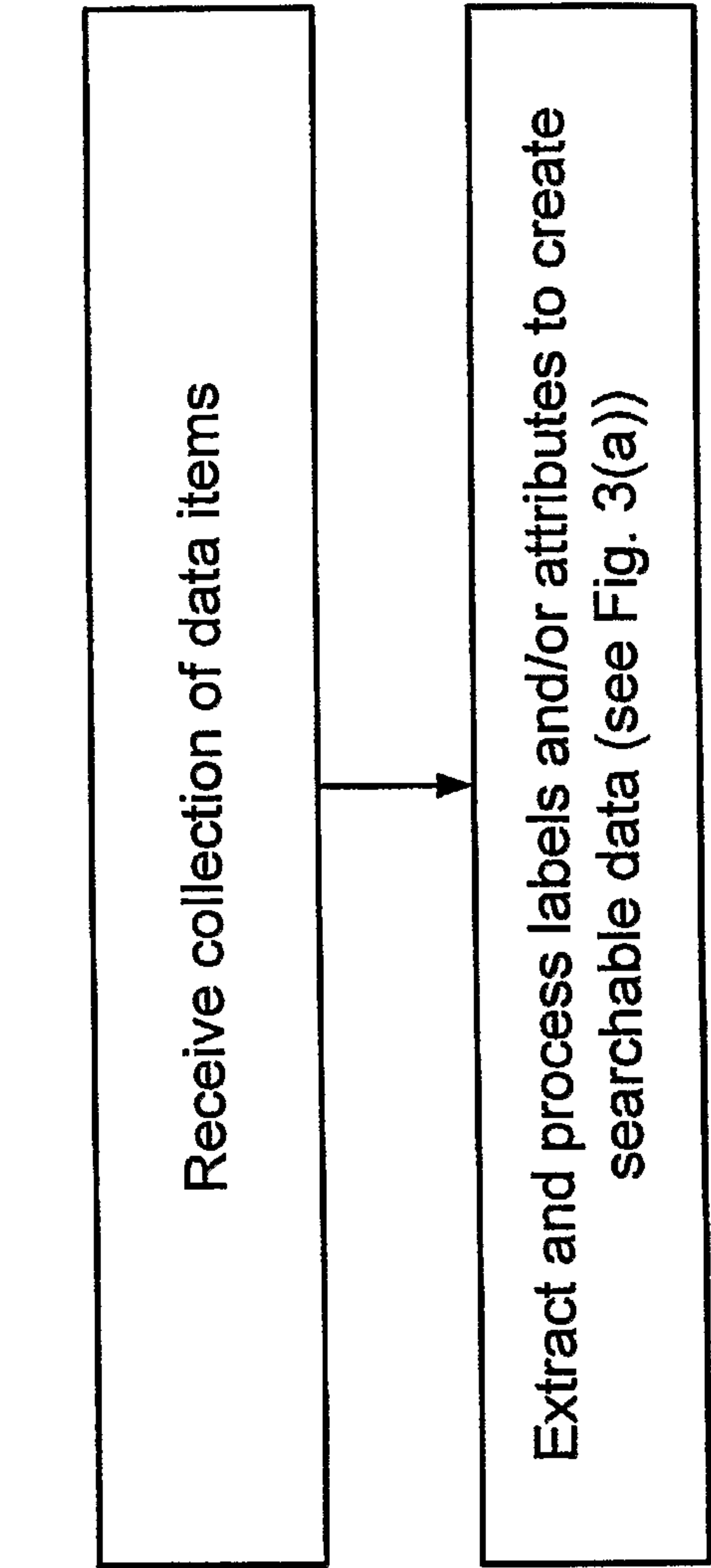


Fig. 2(a)
Overview,
Data
Receipt

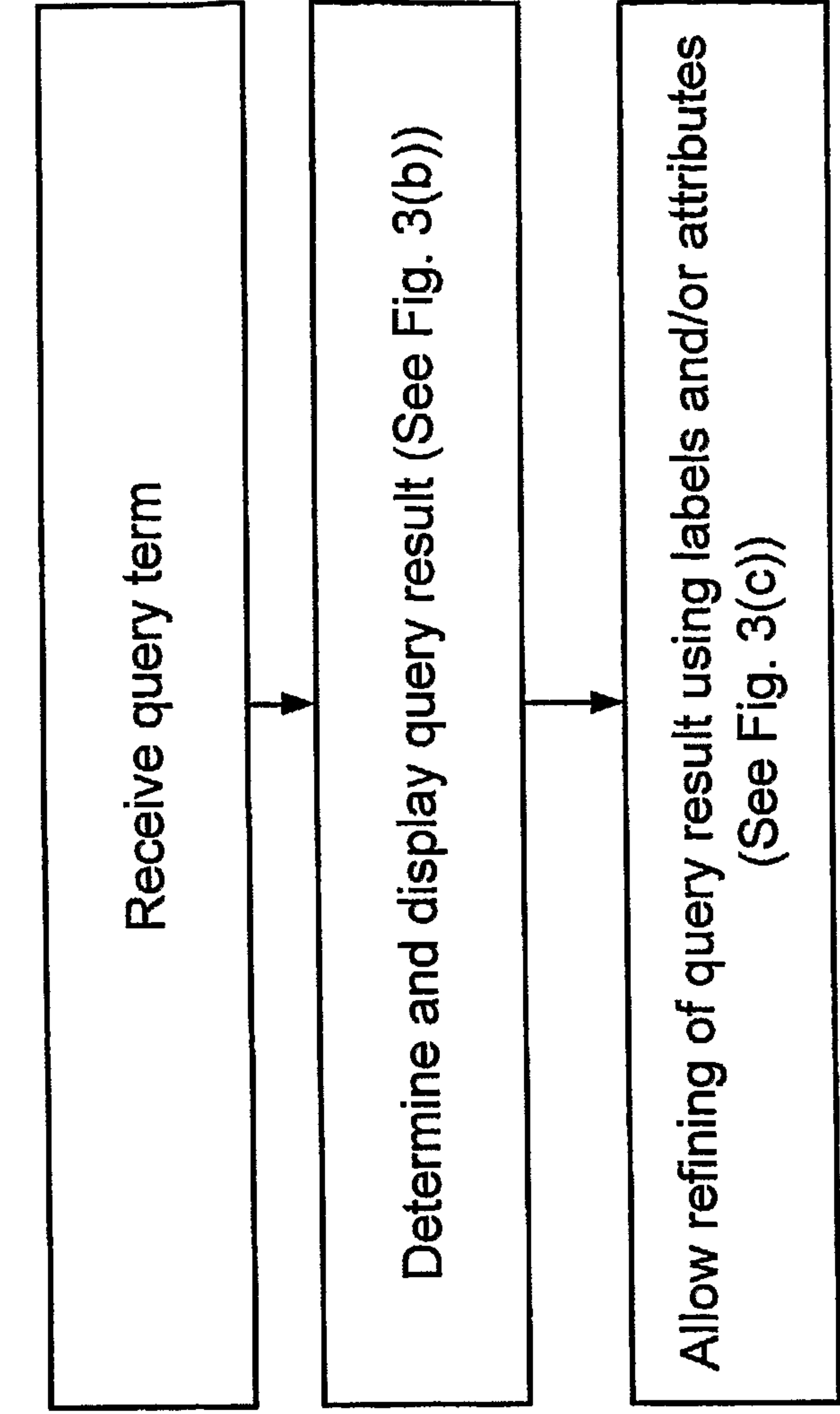


Fig. 2(b)
Overview,
Search

5/27

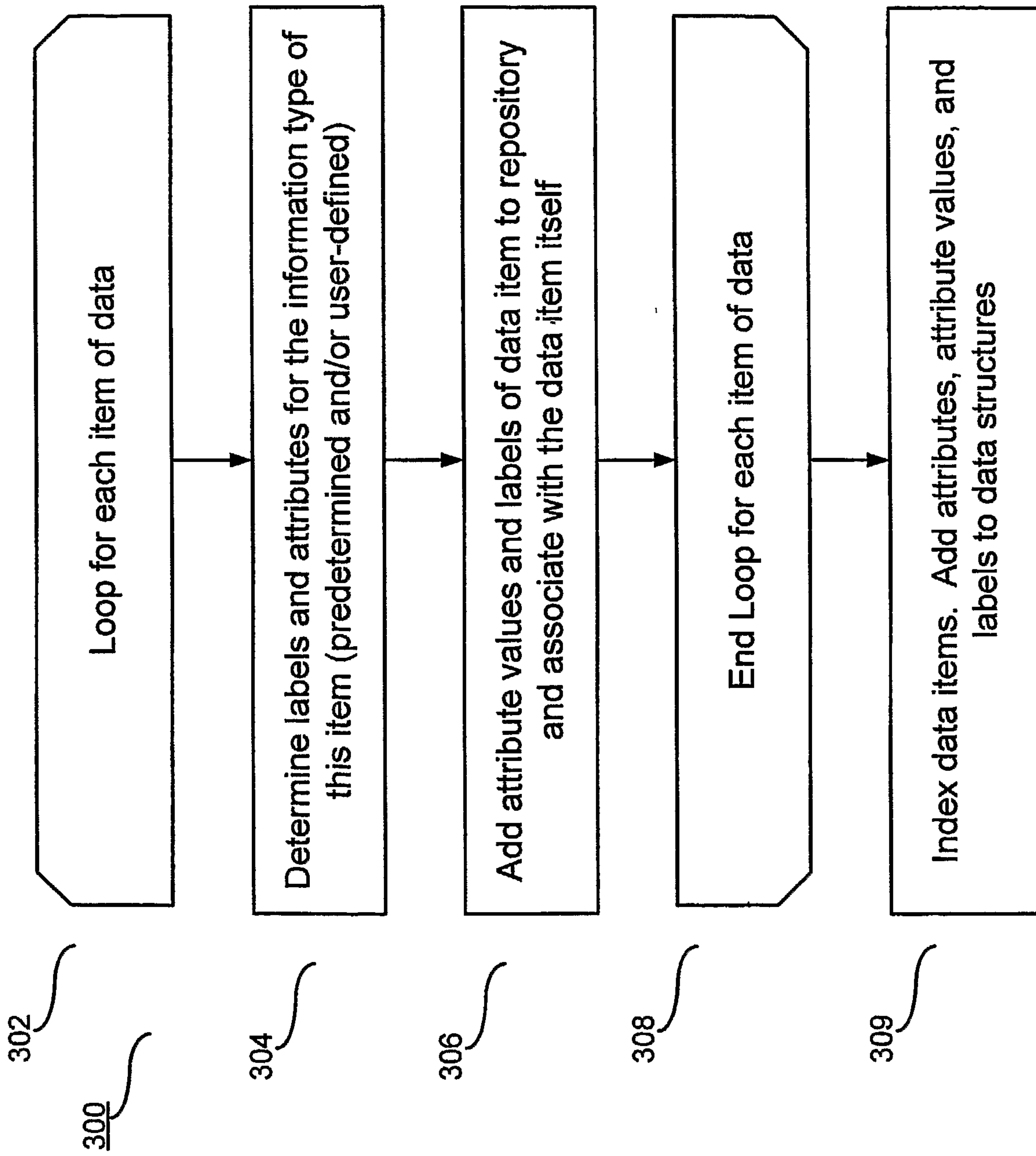
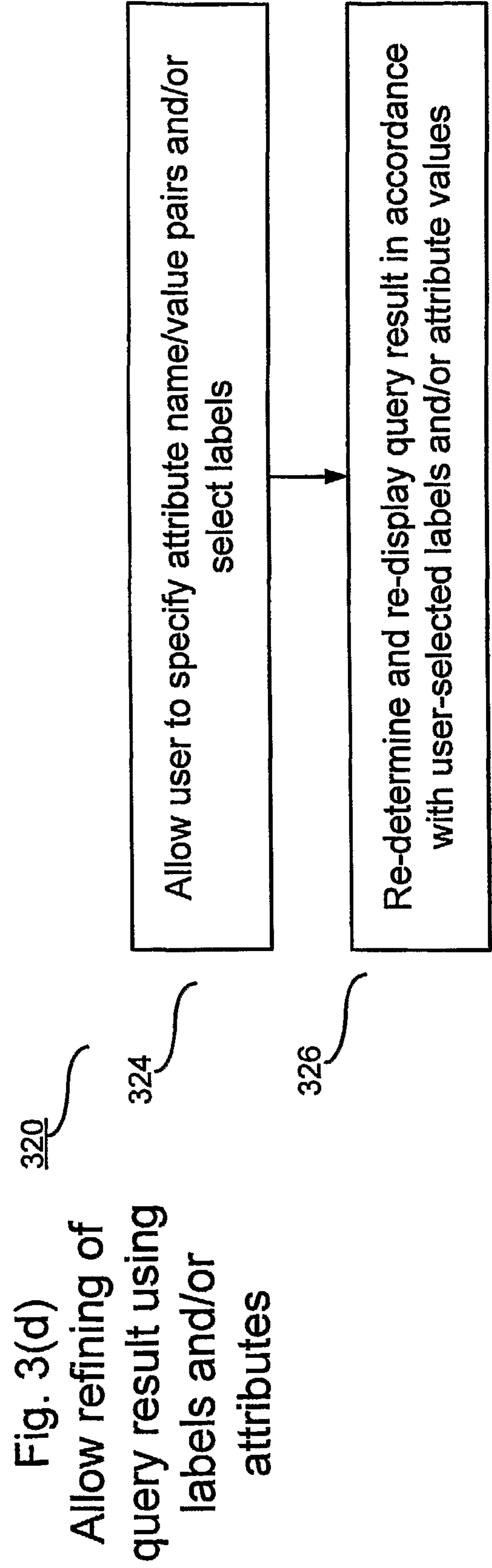
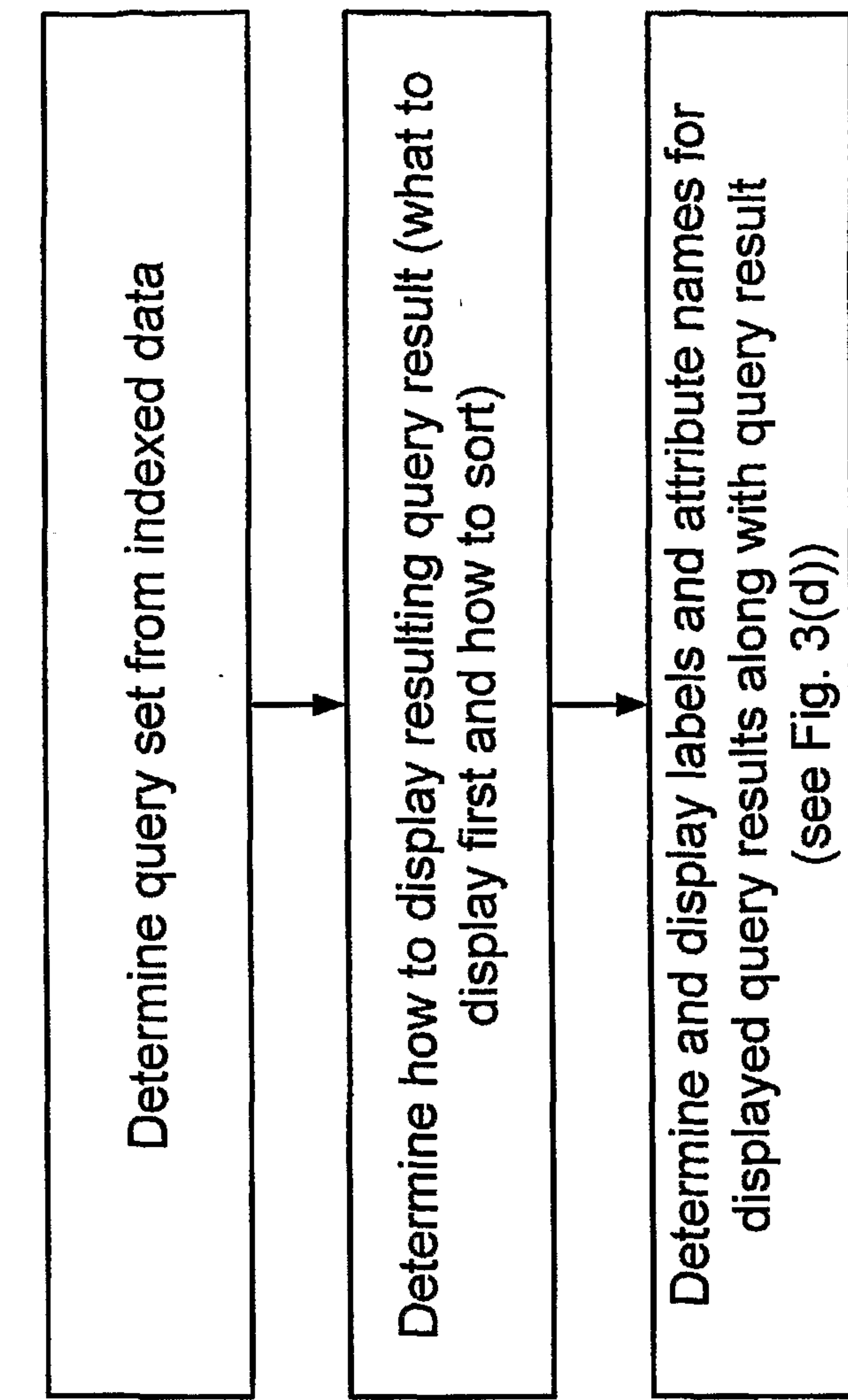
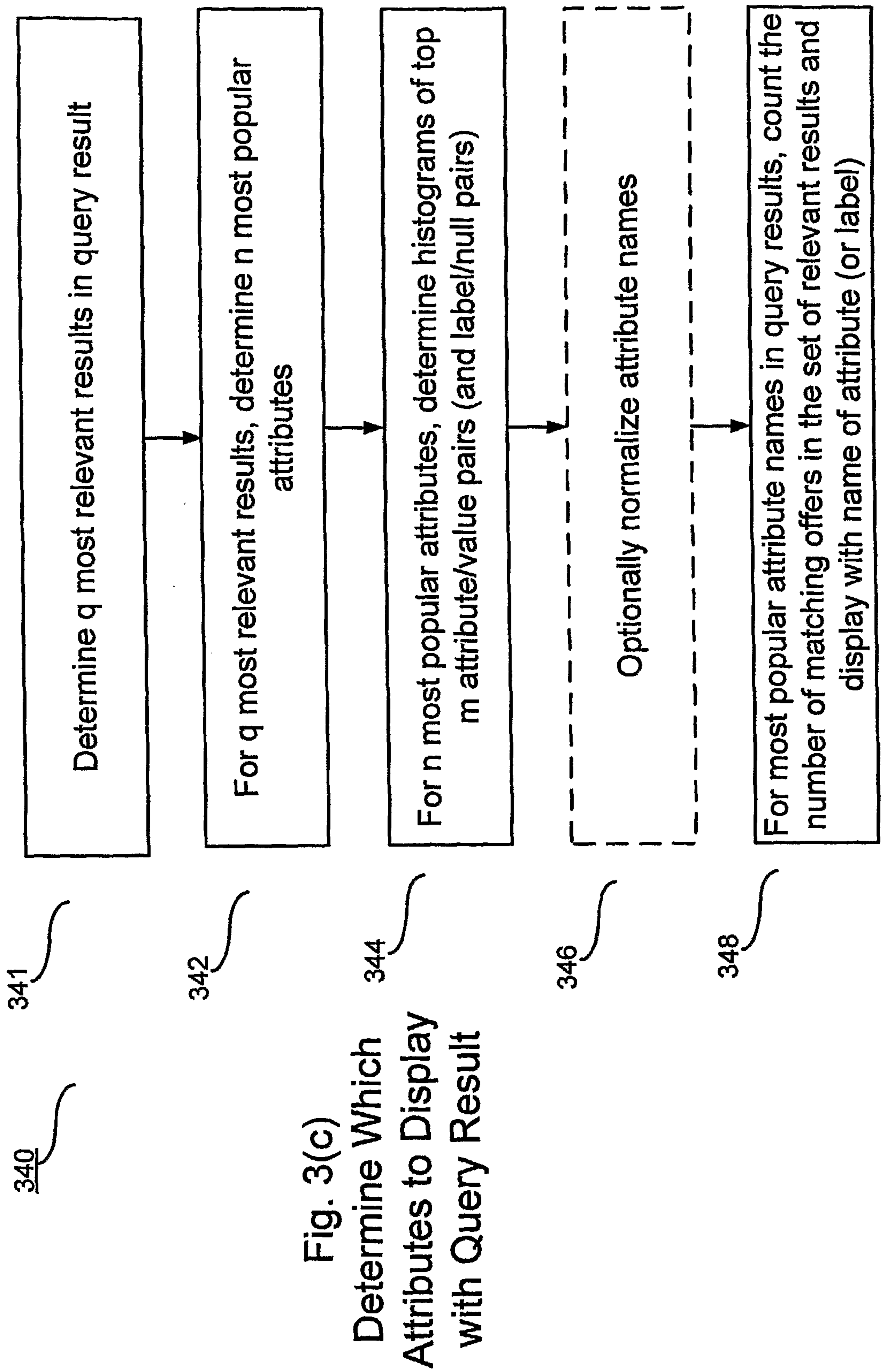


Fig. 3(a)
Extract and process labels and/or attributes (Set-up)

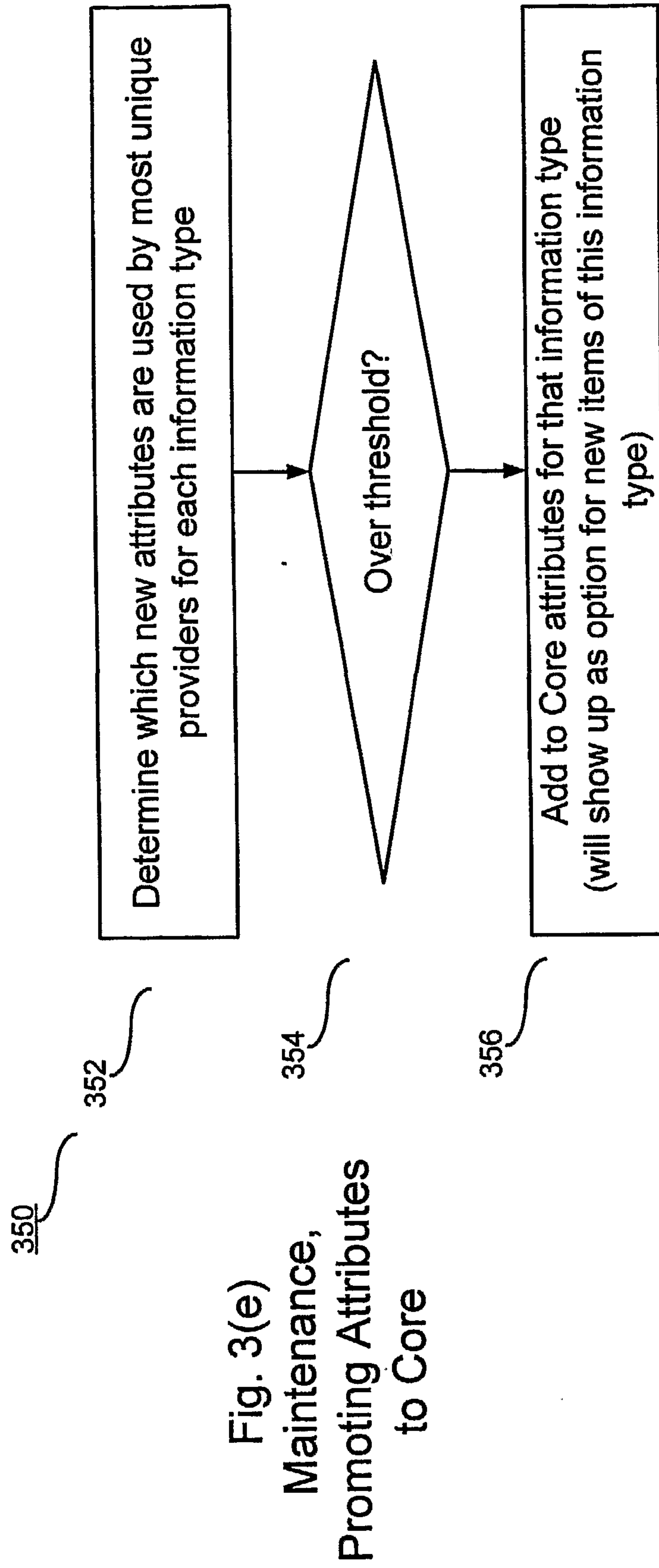
6/27



7/27



8/27



9/27

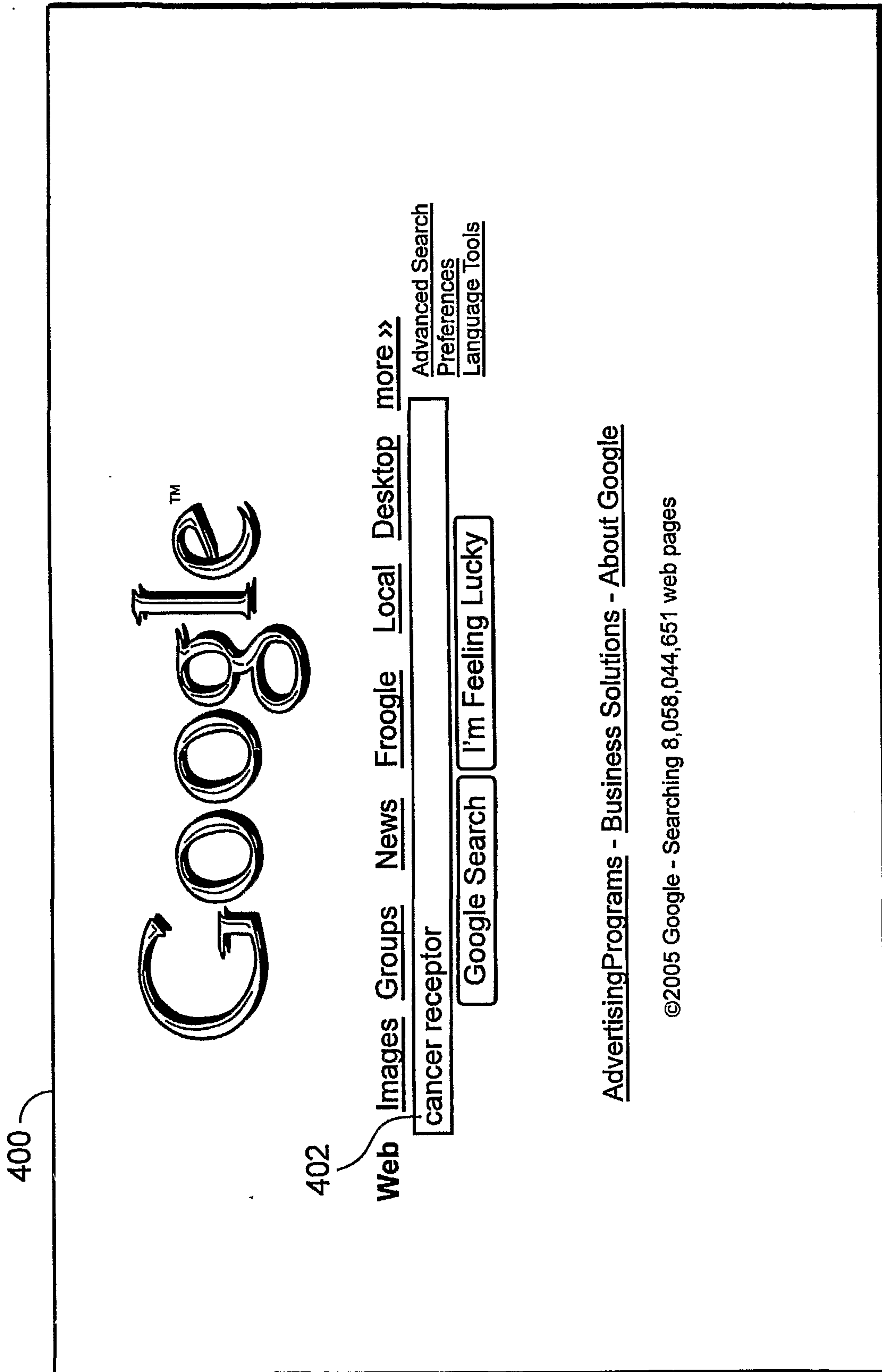


Fig. 4(a)

402

Google™

Web Images Groups News Froogle Local more »

cancer receptor

Search

Advanced Search

Preferences

Filter by: journal (2050) pubmed (5654) news source (105) authors (32)

Results 1 - 10 of about 3,900,000 for cancer receptor. (0.28 seconds)

Structure Of Key Breasts Cancer Receptor Determined

Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ... , Journal: Sience, 1995 - ncbi.nlm.nih.gov

A team of scientists from Johns Hopkins and the biotechnology company Genitope has unlocked the 3-D structure of a receptor that goes awry in 20 percent to ...

www.sciencedaily.com/releases/2003/02/030213073521.htm - 62k - Cached - Similar pages

Structure Of Key Breasts Cancer Receptor Determined

Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ... , Journal: Sience, 1995 - ncbi.nlm.nih.gov

Structure of Key Breast Cancer Receptor Determined. A team of scientists from Johns Hopkins and the biotechnology company Genitope has unlocked the 3-D ...

www.hopkinsmedical.org/press/2003/February/030212.htm - 8k - Cached - Similar pages

Artemis — March 2003

Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ... , Journal: Sience, 1995 - ncbi.nlm.nih.gov

Structure of Key Breast Cancer Receptor Determined ... The findings also explain why the HER2 receptor behaves so differently from its relatives HER1, ...

www.hopkinsmedical.org/breastcenter/ artemis/200303/feature14.htm - 12k - Cached - Similar pages

[More results from www.hopkinsmedicine.org]

Structure Of Key Breasts Cancer Receptor Determined

Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ... , Journal: Sience, 1995 - ncbi.nlm.nih.gov

A team of scientists from Johns Hopkins and the biotechnology company Genitope has unlocked the 3-D structure of a receptor that goes awry in 20 percent to ...

www.eurekalert/pub_releases/ 2003-02/fhmi-sok020703.php - 8k - Cached - Similar pages

404

406

Fig. 4(b)

SUBSTITUTE SHEET (RULE 26)



12/27

402

Google™

Web

Images

Groups

News

Froogle

Local

more »

cancer receptor filter: journal filter: year

Search

Advanced Search

Preferences

404

Filter by: pubmed (98) news source (50) authors (32) citations (20) more»

Journal title:

Year:

Go

408

search: cancer receptor > journal > year

410

Structure Of Key Breasts Cancer Receptor Determined

Citations: 200, Journal: Science, 2004 - ncbi.nlm.nih.gov, Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...

A team of scientists from Johns Hopkins and the biotechnology company Genitope

has unlocked the 3-D structure of a receptor that goes awry in 20 percent to ...

www.sciencedaily.com/releases/2003/02/030213073521.htm - 62k - [Cached](#) - [Similar pages](#)

411

Structure Of Key Breasts Cancer Receptor Determined

Citations: 198, Journal: Science, 2003 - ncbi.nlm.nih.gov, Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...

Structure of Key Breast Cancer Receptor Determined. A team of scientists from

Johns Hupkins and the biotechnology company Genitope has unlocked the 3-D ...

www.hopkinsmedical.org/press/2003/February/030212.htm - 8k - [Cached](#) - [Similar pages](#)

406

Artemis -- March 2003

Citations: 198, Journal: Science, 2003 - ncbi.nlm.nih.gov, Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...

Structure of Key Breast Cancer Receptor Determined ... The findings also explain

why the HER2 receptor behaves so differently from its relatives HER1, ...

www.hopkinsmedical.org/breastcenter/ artemis/200303/feature14.htm - 12k - [Cached](#) - [Similar pages](#)

[More results from www.hopkinsmedicine.org]

Results 1 - 10 of about 4,656 (0.28 seconds)

Fig. 4(d)

SUBSTITUTE SHEET (RULE 26)

414

402

412

413

416

Google

BASE

cancer receptor

Search within Journals

Search Google Base

Search all of Google Base

Results 1 - 10 of about 30 for cancer receptor (0.02 seconds)

Posted Items:

Journals

Refine your search:

Biotechnology (30)

Medical (15)

Photography (6)

Date (10)

Author (10)

Pubmed (10)

Citations (6)

more...

Sort by relevance

Sort by date

Structure Of Key Breasts Cancer Receptor Determined - Aug 25, 2005

Citations: 198

Journal: Science, 1995

Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...

Structure of Key Breast Cancer Receptor Determined. A team of scientists from Johns Hopkins and the biotechnology company Genitope has unlocked the 3-D ...

www.eurekalert.org/pub_releases/2003-02/jhmi-sok020703.php

Labels: AIDS Article Biotechnology

Artemis -- March 2003 - Aug 23, 2005

Citations: 198

Journal: Science, 1995

Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...

Structure of Key Breast Cancer Receptor Determined ... The findings also explain why the HER2 receptor behaves so differently from its relatives HER1, ...

www.eurekalert.org/pub_releases/2003-02/jhmi-sok020703.php

Labels: AIDS Article Biotechnology

Structure Of Key Breasts Cancer Receptor Determined - Aug 26, 2005

Citations: 200

Journal: Science, 1995

Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...

A team of scientists from Johns Hopkins and the biotechnology company Genitope has unlocked the 3-D structure of a receptor that goes away in 20 percent to ...

www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=1871007&dopt=Abstract

Labels: Cancer Article Biotechnology

Fig. 4(e)

SUBSTITUTE SHEET (RULE 26)

414

402

412

413

416



BASE

cancer receptor

Search Google Base

☒ Search within Journals ☐ Search all of Google Base

Post your own item

Posted Items: Journals

Results 1 - 10 of about 30 for cancer receptor (0.02 seconds)

Refine your search: [Biotechnology \(30\)](#) [Medical \(15\)](#) [Photography \(6\)](#) [Pubmed \(10\)](#) [Citations \(6\)](#) [Location \(3\)](#) [more...](#)

Date: between -

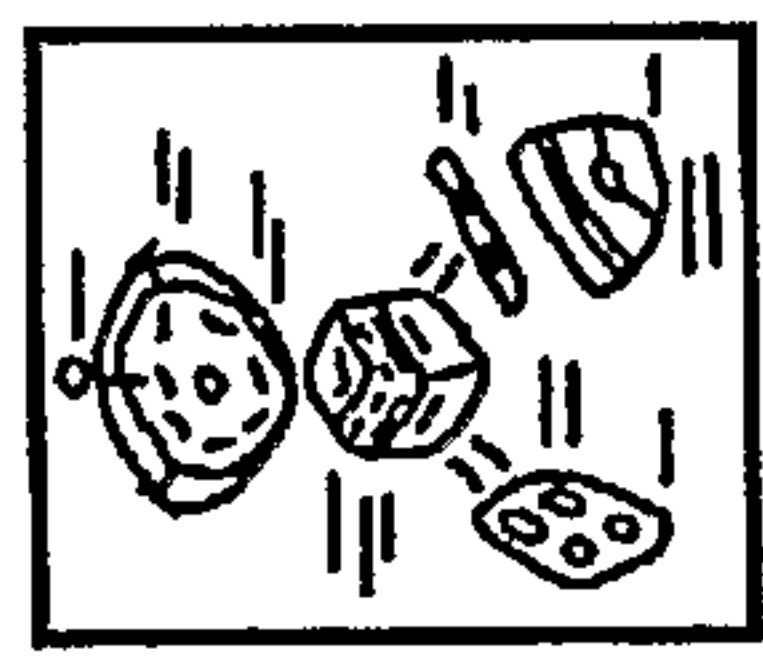
Sort by relevance Sort by date

420

406'

Structure Of Key Breasts **Cancer Receptor Determined** - Aug 25, 2005
Citations: 198 Journal: Science, 1995 Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...
Structure of Key Breast Cancer Receptor Determined. A team of scientists from Johns Hopkins and the biotechnology company Genitope has unlocked the 3-D ...
www.eurekalert.org/pub_releases/2003-02/jhmi-sok020703.php
Labels: [AIDS](#) [Article](#) [Biotechnology](#)

Artemis — March 2003 - Aug 23, 2005
Citations: 198, Journal: Science, 1995 Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...
Structure of Key Breast Cancer Receptor Determined ... The findings also explain why the HER2 receptor behaves so differently from its relatives HER1, ...
www.eurekalert.org/pub_releases/2003-02/jhmi-sok020703.php
Labels: [AIDS](#) [Article](#) [Biotechnology](#)



Structure Of Key Breasts **Cancer Receptor Determined** - Aug 26, 2005
Citations: 200 Journal: Science, 1995 Author: S Markowitz, J Wang, L Myeroff, R Parsons, L Sun, ...
A team of scientists from Johns Hopkins and the biotechnology company Genitope has unlocked the 3-D structure of a receptor that goes awry in 20 percent to ...
www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=1871007&dopt=Abstract Labels: [Cancer](#) [Article](#) [Biotechnology](#)

Fig. 4(f)

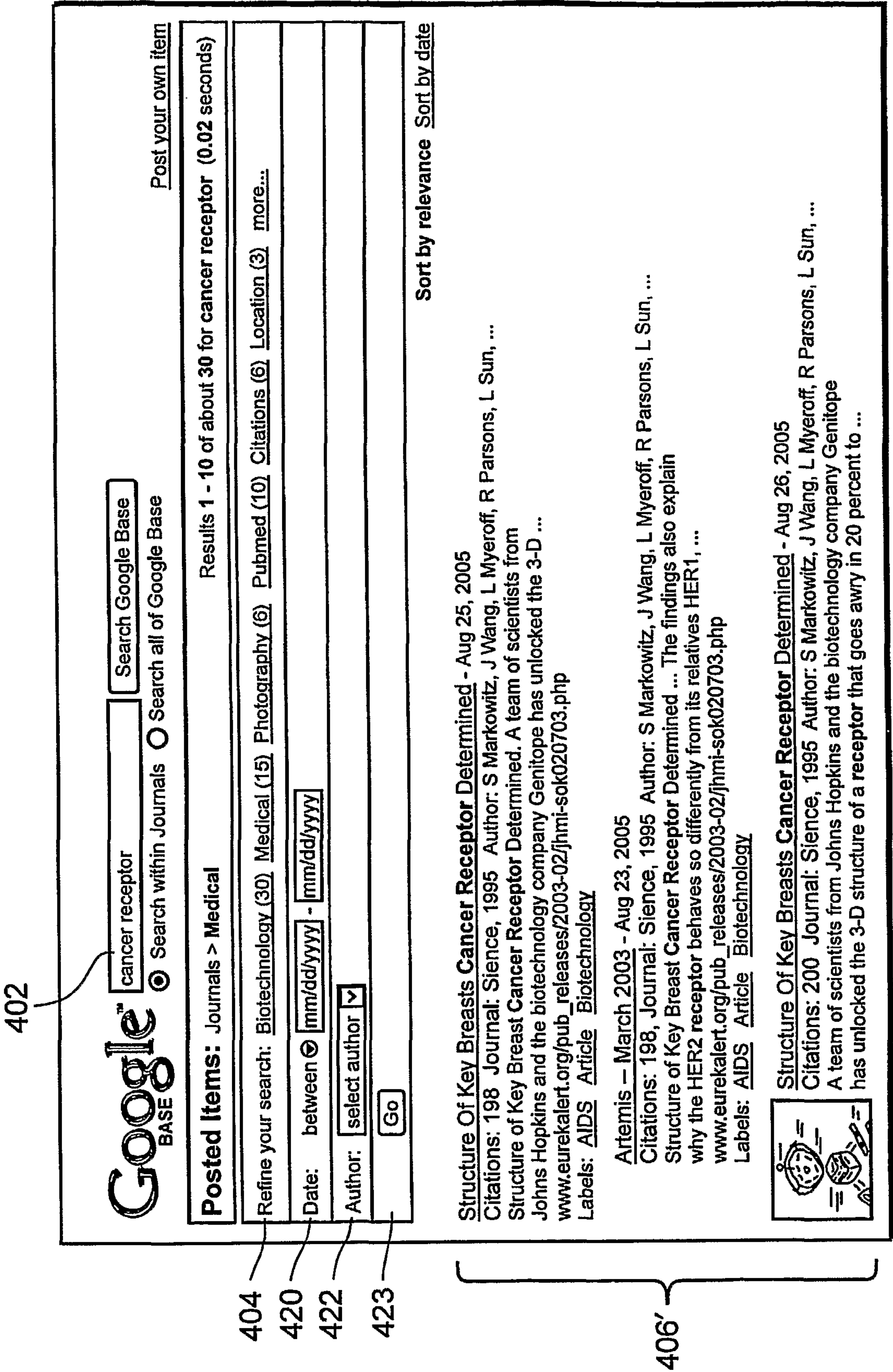


Fig. 4(g)

500

How we represent attributes in our data

store.
>
> Attributes and labels are one of the following types:
> - BOOLEAN
> - INT
> - FLOAT
> - URL
> - STRING
> - LOCATION
> - DATE
> - DATE RANGE
>
> Attributes and labels are indicated by meta-tags as follows:

>
> <start name>
> name
> </end name>
> <start value>
> value
> </end value>

Fig. 5(a)

510

<start name> journalname </end name>
<start value> Journal of Inflammation </end value>

Fig. 5(b)
Example of Attribute Data

520

<start name> Medical </end name> <start value>nullattribute </end value>

Fig. 5(c)
Example of Label Data

530

Data information type	Attributes/and attr type for this type of data item of that Type
Data information type 1	Attr/type #1-1, Attr/type #1-2,...
...	
Data information type n	Attr/type #n-1, Attr/type #n-2,...

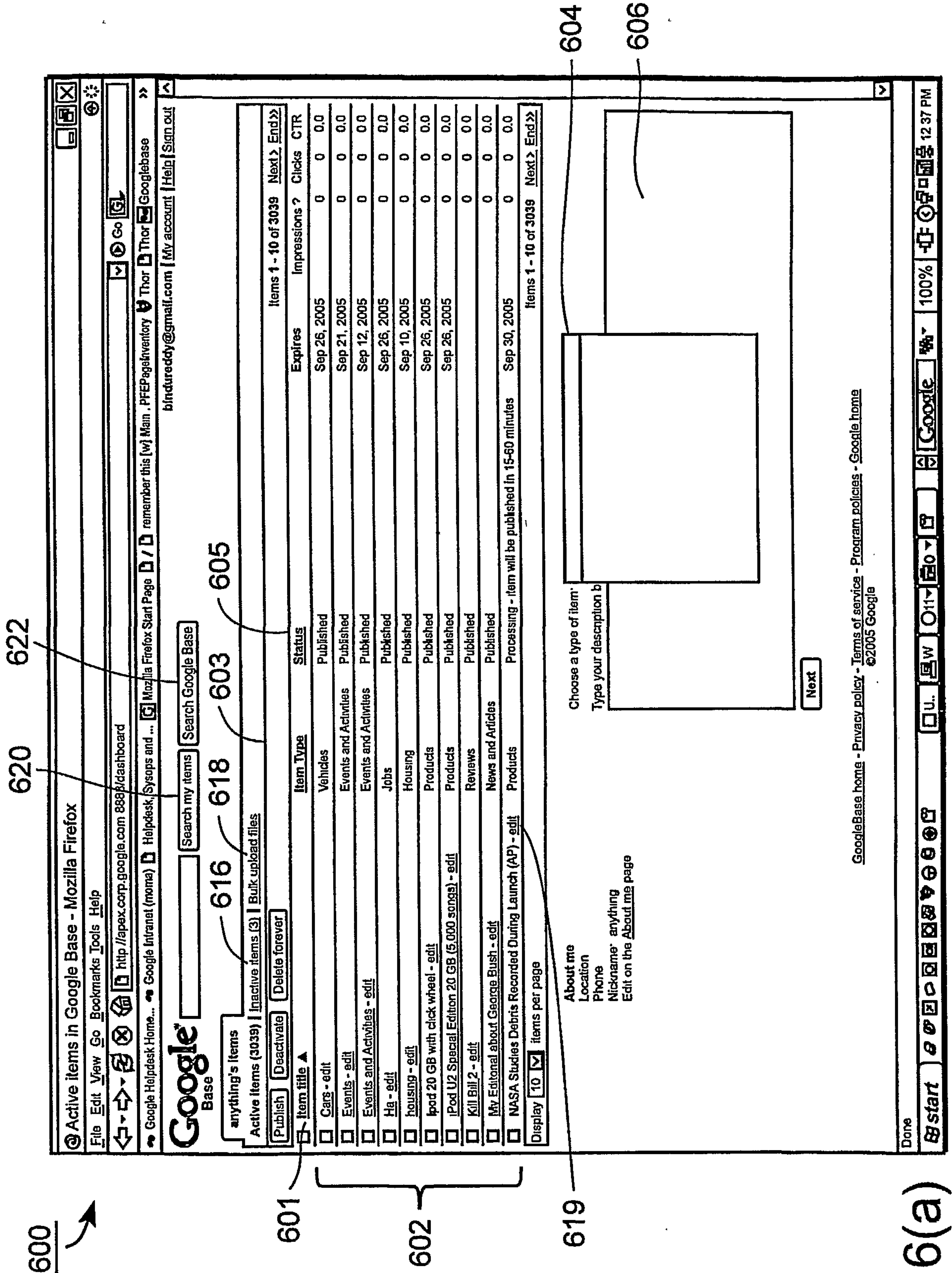
Fig. 5(d)
Type-Attribute Mapping

540

Data information type	Attributes/and attr type for this type of data item of that Type
Events and Activites	Date/date, Holiday/ holidayname
...	
Product	Numavail/integer

Fig. 5(e)
Example of data information types and their attributes

18/27



600

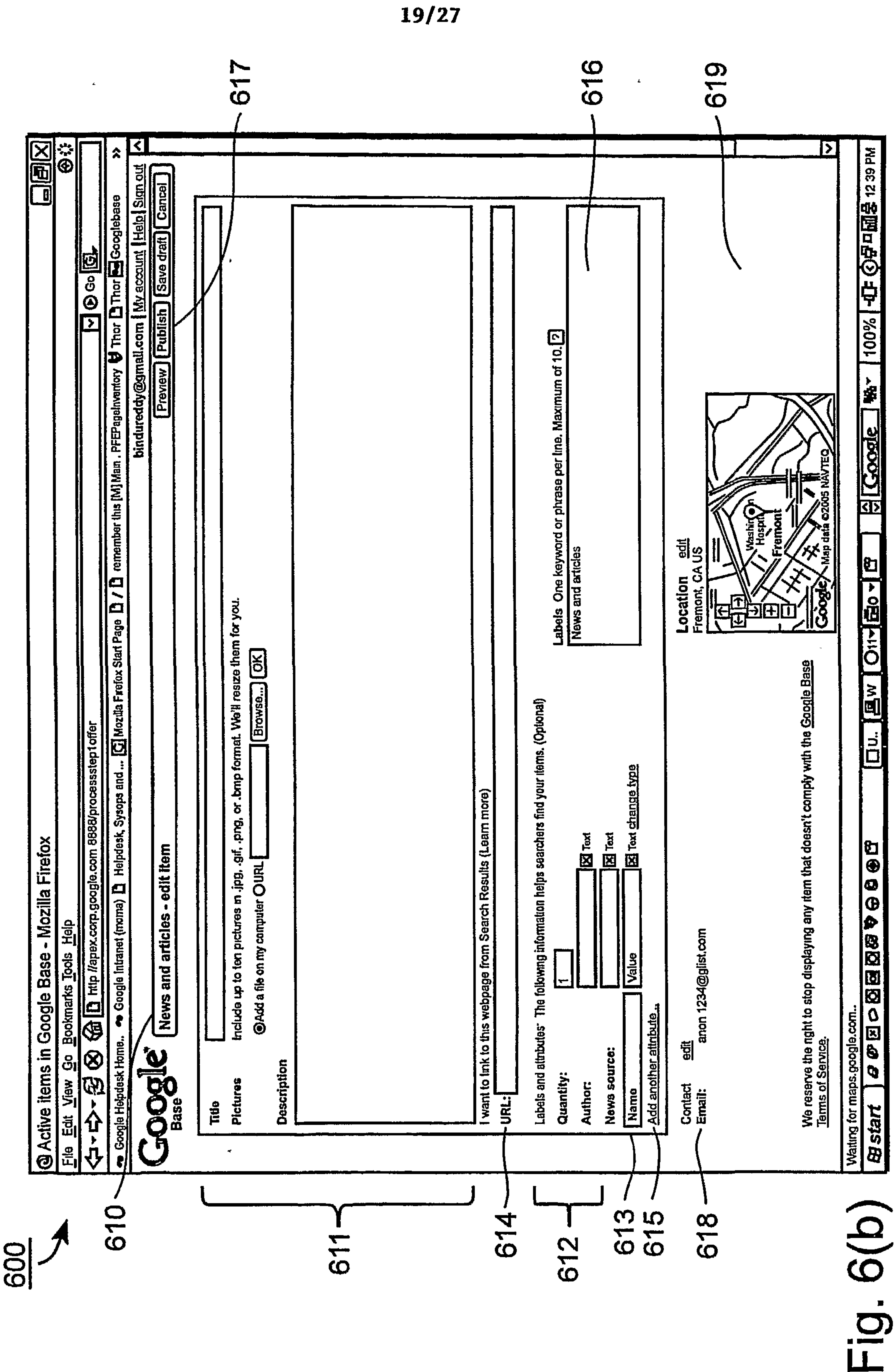


Fig. 6(b)

20/27

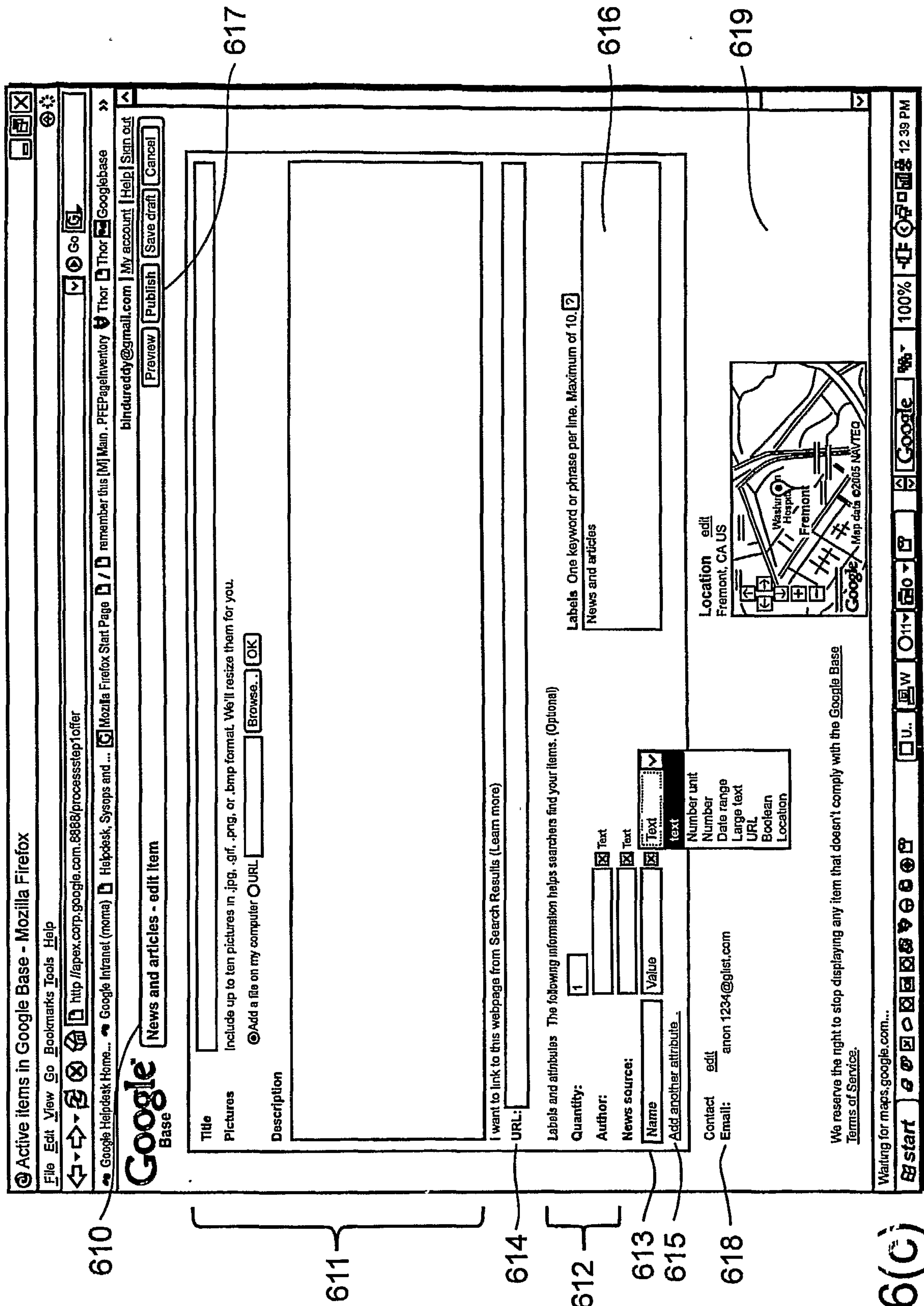


Fig. 6(c)

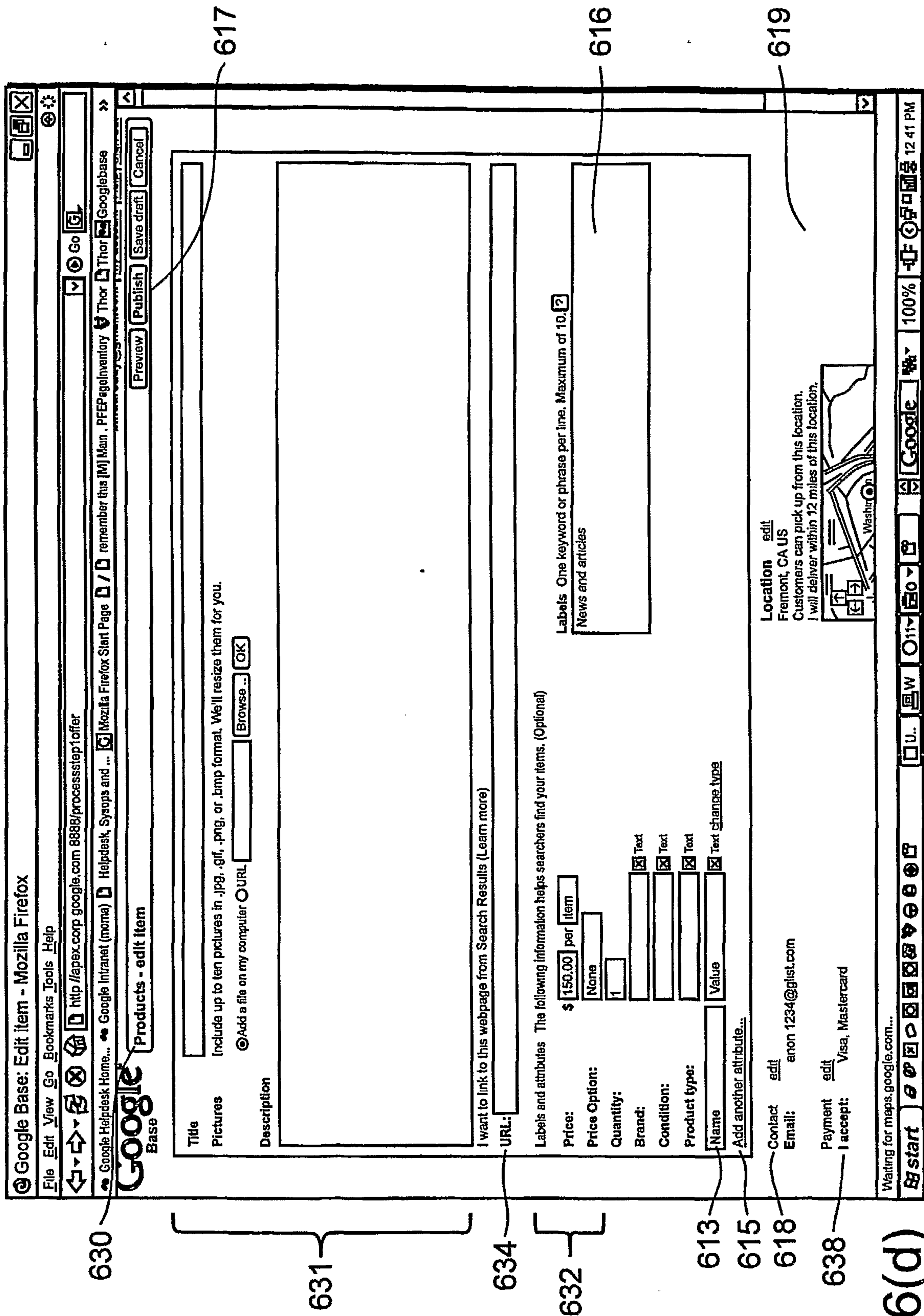


Fig. 6(d)

22/27

632 {

613

615

618

638

617

619

Google Base: Edit item - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://apex.corp.google.com/8888/processstep1offer

Google Helpdesk Home... Google Intranet (moma) Helpdesk, Sysops and ... Mozilla Firefox Start Page / remember this [M] Main . PFEPageInventory Thor Thor Googlebase

Quantity: 1

Brand: Text

Condition: Text

Product type: Text

Name: Value

Value: Text change type

Add another attribute...

Contact

This will be published on the Internet

Nick name: Text

Phone Number: Text

Email: anon1234@glist.com
bindueddy@gmail.com
Do not include email contact

[OK] [Cancel]

Payment

I accept: ☐ Cash ☐ Check ☐ CBG ☐ Visa ☐ MasterCard ☐ Discover ☐ American Express ☐ Wire Transfer ☐ PayPal

Notes: Text

[OK] [Cancel]

This item will expire in 31 days.
Maximum of 31 days. Expired items are automatically moved to your inactive items list.

We reserve the right to stop displaying any item that doesn't comply with the Google Base Terms of Service.

Preview Publish Save draft Cancel

GoogleBase home - Privacy policy - Terms of service - Program policies - Google home
©2005 Google

start http://apex.corp.google.com/8888/processstep1offer# 100% 12:41 PM

Fig. 6(e)

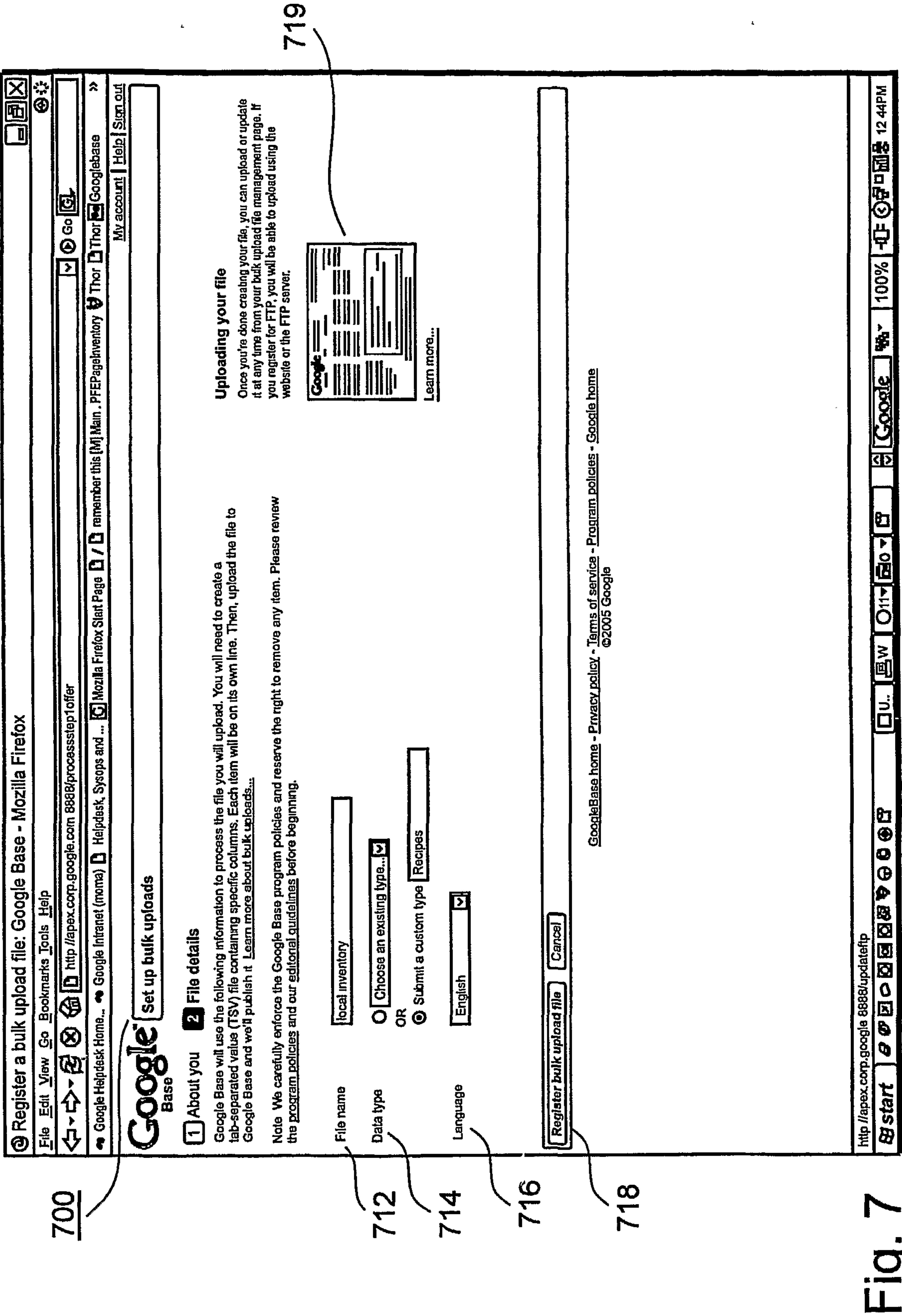


Fig. 7

24/27

<n names, separated by tabs>
 <item#1 attr#1 value> <tab> ... <tab> <item #1 attr#n value>
 ...
 <item#m attr#1 value> <tab> ... <tab> <item #m attr#n value>
 <!-- Comments - ->

Fig. 8(a)

25/27

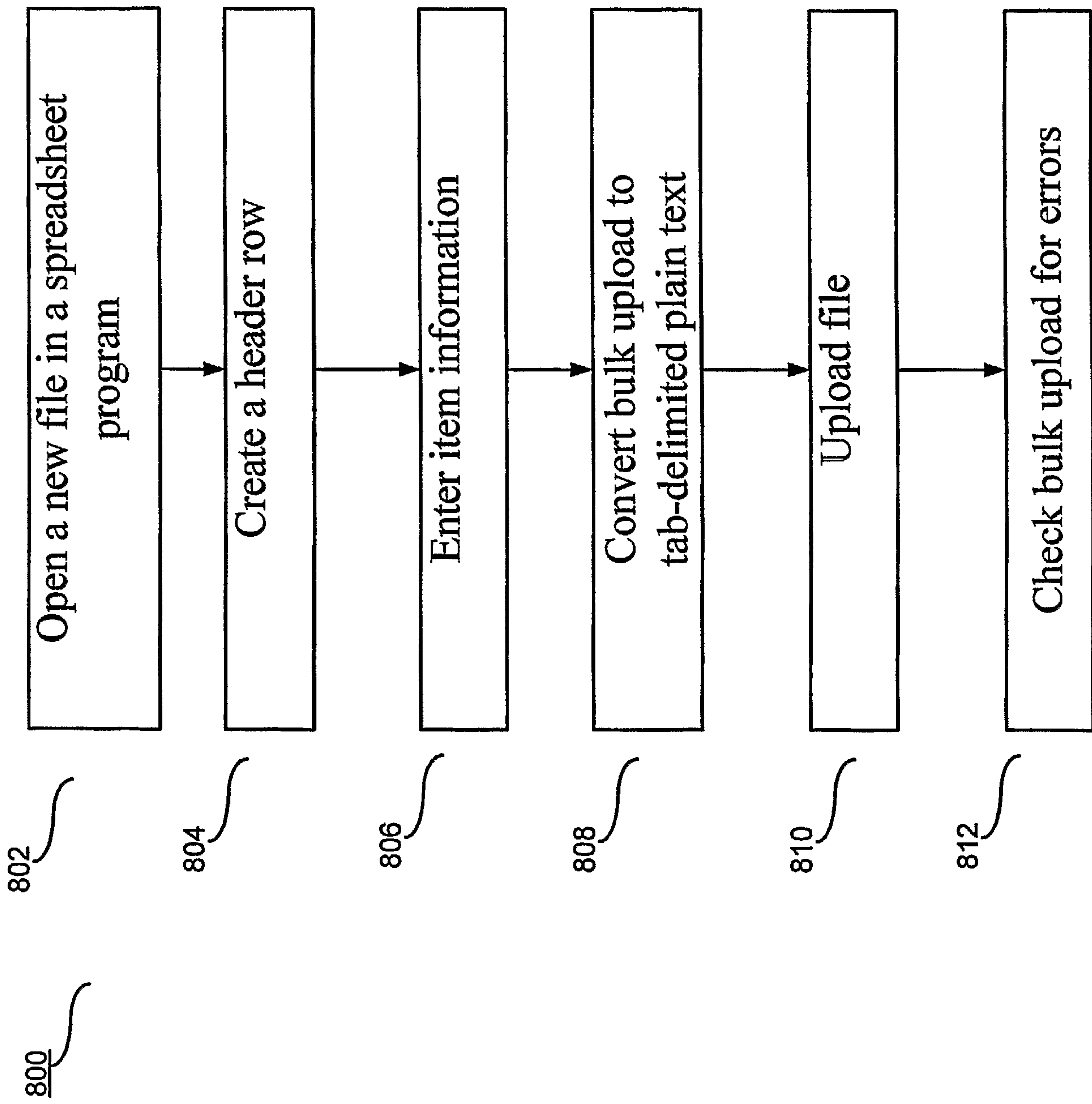


Fig 8(b)

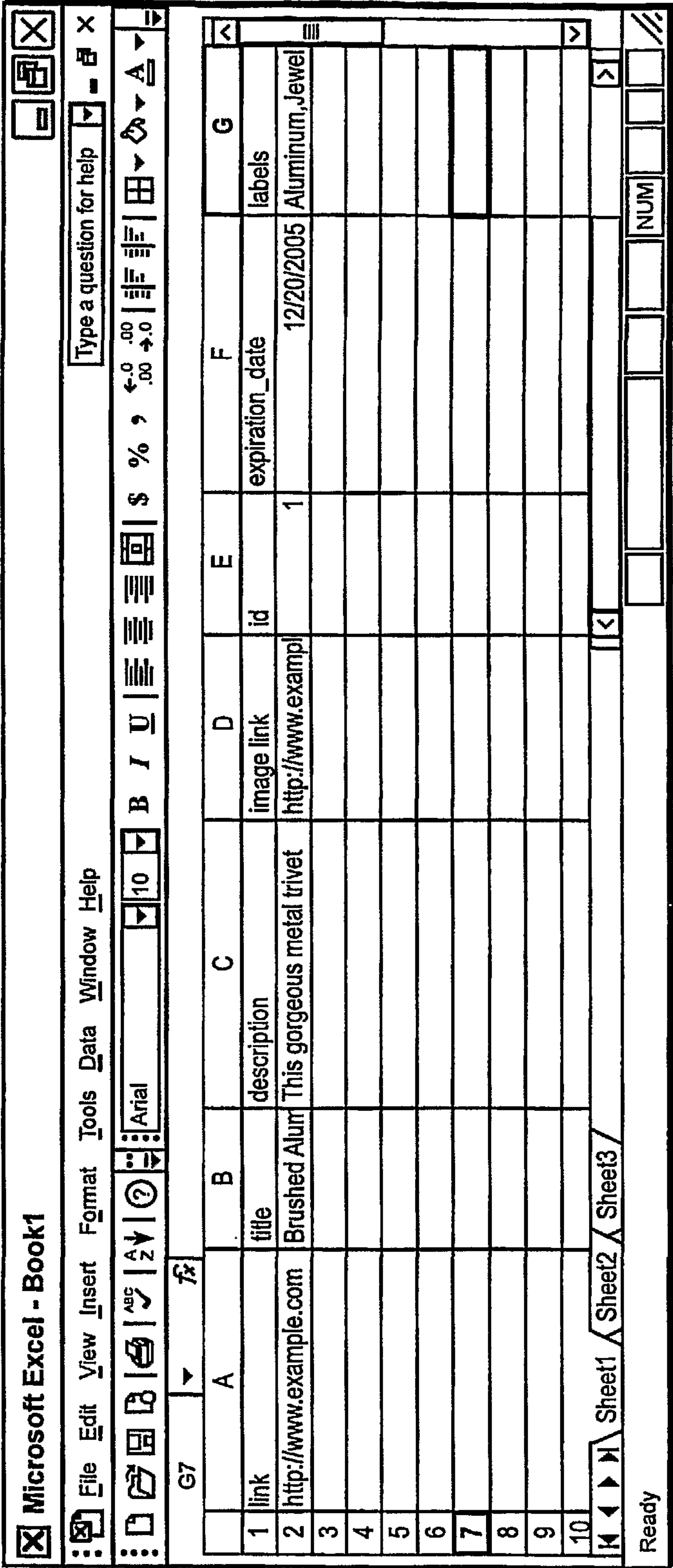


FIG. 8(c)

ॐ नमः ॐ



Fig. 8(d)