

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
22 December 2005 (22.12.2005)

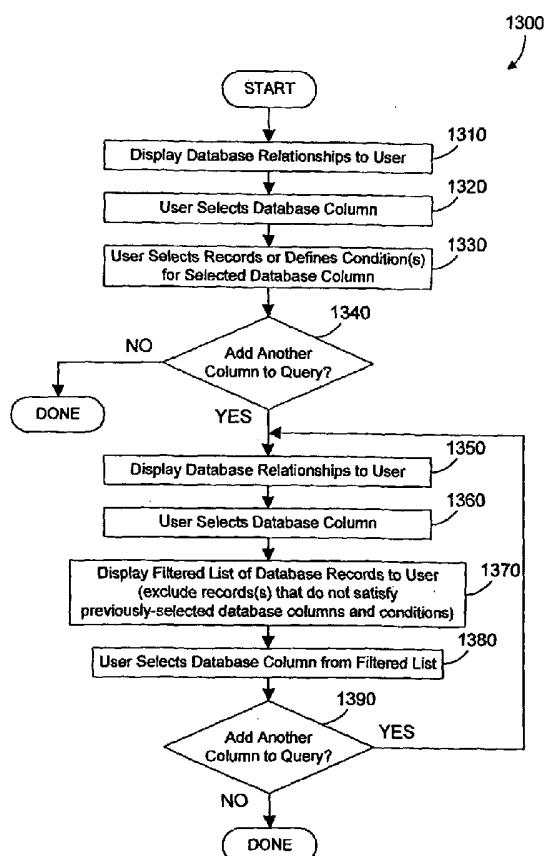
PCT

(10) International Publication Number
WO 2005/122008 A1

- (51) International Patent Classification⁷: **G06F 17/30**
- (21) International Application Number:
PCT/EP2005/052117
- (22) International Filing Date: 10 May 2005 (10.05.2005)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
10/865,261 10 June 2004 (10.06.2004) US
- (71) Applicant (for all designated States except US): **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, NY 10504 (US).
- (71) Applicant (for MG only): **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41 North Harbour, Portsmouth Hampshire PO6 3AU (GB).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **DJUGASH, Judy, Immaculate, Tina** [US/US]; 903 White Oak Court, New Brighton, MN 55112 (US). **TRAN, Hoa, Thu** [US/US]; 5022 19 1/2 Avenue NW, Rochester, MN 55901 (US).
- (74) Agent: **BURT, Roger, James**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).
- (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, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, 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.

[Continued on next page]

(54) Title: DYNAMIC GRAPHICAL DATABASE QUERY AND DATA MINING INTERFACE



(57) Abstract: A graphical query and data mining interface provides visual feedback to a user during the construction of a query that helps the user determine the quality of the query as the query is being built. The graphical query and data mining interface determines relationships in the database from a database relationship document, such as an XML document. These relationships may be between columns, including columns in different tables. The relationship of columns in the database is then graphically displayed to a user. When the user selects a column in the database, a filtered display mechanism displays only those columns or records that satisfy the portion of the query already constructed. In this manner dynamic information is provided to the user as the user builds the query that indicates to the user the quality of the query.



(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, 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).

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.

Published:

— *with international search report*

Description

DYNAMIC GRAPHICAL DATABASE QUERY AND DATA MINING INTERFACE

Technical Field

[001] This invention generally relates to computer systems, and more specifically relates to apparatus and methods for querying a database.

Background Art

[002] Since the dawn of the computer age, computers have evolved and become more and more powerful. In our present day, computers have become indispensable in many fields of human endeavour including engineering design, machine and process control, information storage and retrieval, and office computing. One of the primary uses of computers is for information storage and retrieval.

[003] Database systems have been developed that allow a computer to store a large amount of information in a way that allows a user to search for and retrieve specific information in the database. For example, an insurance company may have a database that includes all of its policy holders and their current account information, including payment history, premium amount, policy number, policy type, exclusions to coverage, etc. A database system allows the insurance company to retrieve the account information for a single policy holder among the thousands and perhaps millions of policy holders in its database.

[004] Retrieval of information from a database is typically done using queries. A query usually specifies conditions that apply to one or more columns of the database, and may specify relatively complex logical operations on multiple columns. The database is searched for records that satisfy the query, and those records are returned as the query result.

[005] One problem with using queries to retrieve information from a database is that using queries typically requires specialized knowledge of a query language, such as Structured Query Language (SQL), as well as detailed knowledge of the database and its relationships. There are many applications where a person needs to query a database, but does not have the detailed knowledge of a query language or the details of the database. Some efforts have been made to provide a graphical query interface that allows a person that does not know SQL to query a database. The main focus of these known graphical query interfaces is abstracting the database and providing an easy-to-use interface for building queries. One problem with these known graphical query interfaces is a user can construct queries that are not very meaningful because they return no data, or because they return thousands or millions of records. Because

the graphical query interface abstracts the details of the database to the user, the user has no idea whether two tables might represent disjoint sets of data. As a result, the user receives no feedback from known graphical query interfaces regarding the quality of the query until the query is completely built and then executed. If the size of the dataset is too large or too small, the user has no information regarding relationships in the database that allow the user to modify the query to return an acceptable dataset.

- [006] The result in the prior art is the generation of queries that are not terribly useful because they return a dataset that is too large or too small to be useful. Without a way to generate queries in a way that provides an indication of the quality of the query before the query is executed, the computer industry will continue to suffer from the generation and execution of queries that do not return a useful dataset.

Disclosure of Invention

- [007] The invention provides a method as claimed in claim 1 and corresponding apparatus and computer program.
- [008] A graphical query and data mining interface provides visual feedback to a user during the construction of a query that helps the user determine the quality of the query as the query is being built. The graphical query and data mining interface preferably determines relationships in the database from a database relationship document, such as an XML document. These relationships may be between columns, including columns in different tables. The relationship of columns in the database is then graphically displayed to a user. When the user selects a column in the database, a filtered display mechanism displays only those columns or records that satisfy the portion of the query already constructed. In this manner dynamic information is provided to the user as the user builds the query that indicates to the user the quality of the query.
- [009] The foregoing and other features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings.

Brief Description of the Drawings

- [010] The preferred embodiments of the present invention will hereinafter be described in conjunction with the appended drawings, where like designations denote like elements, and:
- [011] FIG. 1 is a block diagram of an apparatus in accordance with the preferred embodiments;
- [012] FIG. 2 is a block diagram of the graphical query and data mining interface shown in FIG. 1;
- [013] FIG. 3 is a flow diagram of a prior art method executed by a known graphical query

interface;

- [014] FIG. 4 is a display of one sample screen in a known graphical query interface;
- [015] FIG. 5 is a display of a sample screen in a known graphical query interface that is displayed in response to the user selecting the "Create a new query" link on the display in FIG. 4;
- [016] FIG. 6 is a display of a sample screen in a known graphical query interface that is displayed in response to the user selecting the Add Condition button in the display of FIG. 5;
- [017] FIG. 7 is a display of a sample screen in a known graphical query interface that is displayed in response to the user selecting the GeneChip Array button in the display of FIG. 6;
- [018] FIG. 8 is a display of a sample screen in a known graphical query interface that is displayed once the user has selected Human Genome U95B in the display of FIG. 7;
- [019] FIG. 9 is a display of a sample screen in a known graphical query interface that is displayed in response to the user selecting the Add Condition button in the display of FIG. 8;
- [020] FIG. 10 is a display of a sample screen in a known graphical query interface that is displayed in response to the user selecting the Probe Set ID button in the display of FIG. 9;
- [021] FIG. 11 is a display of a sample screen in a known graphical query interface that is displayed once the user has selected the 533777_at Probe Set ID in the display of FIG. 10;
- [022] FIG. 12 is a display of a sample screen in a known graphical query interface that is displayed to allow the user to select one or more columns to display in the query resultset;
- [023] FIG. 13 is a flow diagram of a method executed by the graphical query and data mining interface in accordance with the preferred embodiments;
- [024] FIG. 14 is a display of the graphical query and data mining interface in FIGS. 1 and 2 that displays database relationships in accordance with the preferred embodiments;
- [025] FIG. 15 is a display of the graphical query and data mining interface in FIGS. 1 and 2 that shows the filtered display of information that results from the user selecting the GeneChip Array Entry in FIG. 14, and that allows the user to select from the display selections for GeneChip Array Entries;
- [026] FIG. 16 is a display of the graphical query and data mining interface in FIGS. 1 and 2 when the user selects the Probe Set ID;
- [027] FIG. 17 is a display of the graphical query and data mining interface in FIGS. 1 and 2 that shows the filtered display of information that results from the user selecting the Probe Set ID in FIG. 16, and that allows the user to select from the filtered display

selections for Probe Set ID;

[028] FIG. 18 is a display of the graphical query and data mining interface in FIGS. 1 and 2 when the user selects UniGene;

[029] FIG. 19 is a display of the graphical query and data mining interface in FIGS. 1 and 2 that shows the filtered display of information that results from the user selecting UniGene in FIG. 18, and that allows the user to select from the filtered display selections for UniGene;

[030] FIG. 20 is a display of the graphical query and data mining interface in FIGS. 1 and 2 when the user selects the Display button; and

[031] FIG. 21 is a display of the graphical query and data mining interface in FIGS. 1 and 2 that shows the results of executing the query.

Best Mode for Carrying Out the Invention

[032] Overview

[033] The present invention relates to database queries. For those not familiar with databases or queries, this Overview section will provide background information that will help to understand the present invention.

[034] Known Databases and Database Queries

[035] There are many different types of databases known in the art. The most common is known as a relational database (RDB), which organizes data in tables that have rows that represent individual entries or records in the database, and columns that define what is stored in each entry or record.

[036] To be useful, the data stored in databases must be able to be efficiently retrieved. The most common way to retrieve data from a database is to generate a database query. A database query is an expression that is evaluated by a database manager. The expression may contain one or more predicate expressions that are used to retrieve data from a database. For example, let's assume there is a database for a company that includes a table of employees, with columns in the table that represent the employee's name, address, phone number, gender, and salary. With data stored in this format, a query could be formulated that would retrieve the records for all female employees that have a salary greater than \$40,000. Similarly, a query could be formulated that would retrieve the records for all employees that have a particular area code or telephone prefix.

[037] One popular way to define a query uses Structured Query Language (SQL). SQL defines a syntax for generating and processing queries that is independent of the actual structure and format of the database. SQL is very powerful for those who have detailed knowledge of SQL and who have detailed knowledge of the database being queried. However, there are a growing number of circumstances where people who do not have

a detailed knowledge of SQL or the database need to be able to query the database. As a result, graphical query interfaces have been developed that help a user to query a database even if the user does not know SQL and does not know the detailed relationships in the database. For example, IBM Corporation has developed an object oriented framework known as a Data Discovery and Query Builder. This framework abstracts out the query layer from the user and lets the user build queries using a graphical interface. For example, medical researchers that perform DNA mapping may need to access data in a very large database. A graphical query interface could be defined that uses the Data Discover and Query Builder framework that allows the researcher to access information in the database without writing SQL queries and without understanding many of the relationships in the database.

[038] The emphasis with the Data Discovery and Query Builder framework and with other known graphical query interfaces is to provide data abstraction and analysis plugins. Known graphical query interfaces do not allow a user to view database relationships, and do not filter displayed selections based on the portion of the query that has already been defined. As a result, the user, who does not have detailed knowledge of the database relationships, may define a query that could return millions of records, or could define a query that returns no records. Known graphical query interfaces provide no feedback to the user regarding the quality of the query they are building. As a result, the user may spend time building a query only to find out after executing the query that the query did not return the desired data. The user must then guess at what changes to make to the query so that it returns the desired data.

[039] An example will illustrate the deficiencies in prior art graphical query interfaces. Referring to FIG. 3, a prior art method 300 for graphically building a query begins by displaying a list of database columns to the user (step 310). The user selects a database column from the list (step 320). The user may then select individual records, or may define one or more conditions for the selected database column (step 330). If no other columns need to be added to the query (step 340=NO), the user then selects the column or columns to display in the resultset (step 350). If another column needs to be added to the query (step 340=YES), method 300 loops back to step 310 and continues.

[040] One of the problems in method 300 is that all the available columns in the database are displayed to the user in step 310. This is true even when a query has been partially built that would eliminate a vast majority of the columns from consideration. A simple example will help to illustrate, as shown graphically in FIGS. 4-11.

[041] Referring to FIG. 4, a display window 400 shows a graphical query interface for a gene database. We assume for this example that the user clicks on the "Create a new query" link. In response, the display window 500 of FIG. 5 is displayed to the user. Note that no conditions have been defined for the new query, so the user clicks on the

“Add Condition” button. In response, the display window 600 of FIG. 6 is displayed to the user. We assume for this example that the user selects the GeneChip Array by clicking on the corresponding button, as shown in FIG. 6. In response, the display window 700 is displayed to the user, showing all of the GeneChip Array entries in the database. For this example, we assume a single GeneChip Array named Human Genome U95B is the sole GeneChip Array in the database, so this is the sole button displayed to the user in FIG. 7. Once the user clicks on the Human Genome U95B button in the display window 700 of FIG. 7, the display window 800 of FIG. 8 is displayed to the user. Note that the Conditions portion of the display window 800 now shows that the user has selected the Human Genome U95B GeneChip Array. We assume the user now clicks on the Add Condition button, as shown in FIG. 8. In response, the display window 900 of FIG. 9 is displayed to the user. Note this display window presents the same selections as the display window 600 in FIG. 6. We assume the user now clicks on the Probe Set ID button, as shown in FIG. 9. In response, the list of Probe Set IDs are displayed to a user, as shown in FIG. 10. We assume for this example that there are thousands of Probe Set IDs in the database. The display window 1000 in FIG. 10 only shows a relatively small number, as indicated by the size of the scroll bar button to the right of the display window 1000. At this point, the user has no idea which Probe Set ID or IDs relate to the Human Genome U95B GeneChip Array that has already been selected, because the prior art graphical query interface displays all of the Probe Set IDs that exist in the database. As a result, the user may have to manually hunt through a paper trail to find which Probe Set IDs correspond to the Human Genome U95B GeneChip Array. We assume for this simple example that the selections 1010, 1020 and 1030 in FIG. 10 correlate to valid Probe Set IDs for the Human Genome U95B GeneChip Array. Yet the user is not provided with any of this information by the graphical query interface. Herein lies the primary deficiency in known graphical query interfaces. The user could easily select any of the Probe Set IDs shown in FIG. 10. Selecting any of the Probe Set IDs other than those corresponding to 1010, 1020 and 1030 results in selecting disjoint data, which results in the query returning no rows. Assuming the user somehow successfully identifies that Probe Set ID 5337_at is the desired Probe Set ID (from paper records or from some source of information external to the graphical query interface), the user clicks on button 1020. As a result, the display window 1100 of FIG. 11 is then displayed to the user. Note that the Conditions box now includes both the selections the user has already made. We assume the user clicks on the Change Output button, as shown in FIG. 11. As a result, the screen 1200 shown in FIG. 12 is displayed to the user, allowing the user to select one or more columns for display in the resultset. Once the query conditions are defined and the columns to display have been specified, the user could execute the query.

However, because the graphical query interface illustrated in FIGS. 3-12 does not provide any visual feedback to the user regarding the quality of the query as it is being built, the user will have to wait until the query is actually executed before knowing whether or not the query will return the desired data.

[042] Description of the Preferred Embodiments

[043] The dynamic graphical query and data mining interface in accordance with the present invention provides graphical feedback to the user regarding the quality of the query while the query is being built. Database relationships are displayed to the user. When the user selects something in the database, the display of available selections is filtered according to the portion of the query that has already been constructed. In this manner the user is provided dynamic visual feedback regarding the quality of the query as the query is being built. Using this interface, a user may also mine data from the database because the relationships between database columns are shown, and because the displayed results are filtered according to the user's selections.

[044] Referring to FIG. 1, a computer system 100 is one suitable implementation of an apparatus in accordance with the preferred embodiments of the invention. Computer system 100 is an IBM eServer iSeries computer system. However, those skilled in the art will appreciate that the mechanisms and apparatus of the present invention apply equally to any computer system, regardless of whether the computer system is a complicated multi-user computing apparatus, a single user workstation, or an embedded control system. As shown in FIG. 1, computer system 100 comprises a processor 110, a main memory 120, a mass storage interface 130, a display interface 140, and a network interface 150. These system components are interconnected through the use of a system bus 160. Mass storage interface 130 is used to connect mass storage devices, such as a direct access storage device 155, to computer system 100. One specific type of direct access storage device 155 is a readable and writable CD RW drive, which may store data to and read data from a CD RW 195.

[045] Main memory 120 in accordance with the preferred embodiments contains data 121, an operating system 122, a database 123, and a graphical query and data mining interface 124. Data 121 represents any data that serves as input to or output from any program in computer system 100. Operating system 122 is a multitasking operating system known in the industry as OS/400; however, those skilled in the art will appreciate that the spirit and scope of the present invention is not limited to any one operating system. Database 123 is any suitable database, whether currently known or developed in the future. Database 123 preferably includes one or more tables. Graphical query and data mining interface 124 provides a graphical query interface that provides dynamic feedback to the user that helps the user understand relationships in the database without destroying the abstractions provided by the interface, and in a

way that helps the user build a useful query. The graphical query and data mining interface 124 includes a database relationship document 125 that specifies relationships in the database 123. A significant advantage of the preferred embodiments is the ability to specify relationships in the database relationship document 125 that span across different tables. The database relationship document 125 is preferably an eXtensible Markup Language (XML) document. By reading the database relationship document 125, the graphical query and data mining interface 124 may determine one or more relationships in the database 123, including relationships across multiple tables, which may then be displayed to a user.

[046] The graphical query and data mining interface 124 also includes a filtered display mechanism 126. This mechanism helps the user to know the quality of the query as the query is being constructed. When a user makes a selection in the database, the information displayed to the user is then filtered by the filtered display mechanism 126 to only display information that satisfies all of the previous user selections. In this manner the amount of information presented to the user is reduced so the user can make more intelligent decisions regarding how to build a query that will return a desired number of rows. In addition, the user will know if the query being built will return no records during the construction of the query, thereby allowing the user to back up and specify one or more different selections that will return desired data. This is a huge improvement over the prior art, which allows the user to graphically build a query, but provides no indication of the quality of the query until the query is executed. At the point of executing the query, if the query returns no rows, or thousands of rows, the query is probably not terribly useful to the user. The filtered display mechanism 126 presents only information that meets the criteria of information that the user has previously selected. In this manner the user receives a visual indication of the quality of the query as the query is being built.

[047] Due to the dynamic visual feedback provided by the user, the graphical query and data mining interface 124 may be easily used to mine data from a database. As the user adds selections to the query, the results are repeatedly narrowed and filtered to display only those selections that meet all of the previously-selected criteria. As a result, the interface 124 is a very effective tool for a user to mine data from the database 123.

[048] Computer system 100 utilizes well known virtual addressing mechanisms that allow the programs of computer system 100 to behave as if they only have access to a large, single storage entity instead of access to multiple, smaller storage entities such as main memory 120 and DASD device 155. Therefore, while data 121, operating system 122, database 123, and graphical query and data mining interface 124 are shown to reside in main memory 120, those skilled in the art will recognize that these items are not necessarily all completely contained in main memory 120 at the same time. It should

also be noted that the term “memory” is used herein to generically refer to the entire virtual memory of computer system 100, and may include the virtual memory of other computer systems coupled to computer system 100.

[049] Processor 110 may be constructed from one or more microprocessors and/or integrated circuits. Processor 110 executes program instructions stored in main memory 120. Main memory 120 stores programs and data that processor 110 may access. When computer system 100 starts up, processor 110 initially executes the program instructions that make up operating system 122. Operating system 122 is a sophisticated program that manages the resources of computer system 100. Some of these resources are processor 110, main memory 120, mass storage interface 130, display interface 140, network interface 150, and system bus 160.

[050] Although computer system 100 is shown to contain only a single processor and a single system bus, those skilled in the art will appreciate that the present invention may be practiced using a computer system that has multiple processors and/or multiple buses. In addition, the interfaces that are used in the preferred embodiment each include separate, fully programmed microprocessors that are used to off-load compute-intensive processing from processor 110. However, those skilled in the art will appreciate that the present invention applies equally to computer systems that simply use I/O adapters to perform similar functions.

[051] Display interface 140 is used to directly connect one or more displays 165 to computer system 100. These displays 165, which may be non-intelligent (*i.e.*, dumb) terminals or fully programmable workstations, are used to allow system administrators and users to communicate with computer system 100. Note, however, that while display interface 140 is provided to support communication with one or more displays 165, computer system 100 does not necessarily require a display 165, because all needed interaction with users and other processes may occur via network interface 150.

[052] Network interface 150 is used to connect other computer systems and/or workstations (*e.g.*, 175 in FIG. 1) to computer system 100 across a network 170. The present invention applies equally no matter how computer system 100 may be connected to other computer systems and/or workstations, regardless of whether the network connection 170 is made using present-day analog and/or digital techniques or via some networking mechanism of the future. In addition, many different network protocols can be used to implement a network. These protocols are specialized computer programs that allow computers to communicate across network 170. TCP/IP (Transmission Control Protocol/Internet Protocol) is an example of a suitable network protocol.

[053] At this point, it is important to note that while the present invention has been and will continue to be described in the context of a fully functional computer system,

those skilled in the art will appreciate that the present invention is capable of being distributed as a program product in a variety of forms, and that the present invention applies equally regardless of the particular type of computer-readable signal bearing media used to actually carry out the distribution. Examples of suitable computer-readable signal bearing media include: recordable type media such as floppy disks and CD RW (*e.g.*, 195 of FIG. 1), and transmission type media such as digital and analog communications links.

- [054] Referring now to FIG. 2, additional details of the graphical query and data mining interface 124 shown in FIG. 1 include a record/column selection mechanism 210, a condition specification mechanism 220, and a query result display mechanism 230. The record/column selection mechanism 210 allows a user to select appropriate columns or rows in the database for inclusion in the query being built. The condition specification mechanism 220 allows specifying conditions for the query, such as ranges, arithmetic operators, logical operators, etc. The query result display mechanism 230 displays to the user the results of executing the query.
- [055] Referring to FIG. 13, a method 1300 in accordance with the preferred embodiments begins by displaying database relationships to the user (step 1310). The database relationships are preferably determined by the graphical query and data mining interface reading the corresponding database relationship document 125 (such as an XML document) that defines the relationships between data in the database. For example, in the field of Microarray research, a standard known as MicroArray Gene Expression (MAGE) is evolving. A standard for MAGE is maintained using the XML format called MAGE-ML. MAGE consists of various different packages, and each package has various data elements associated with it. These relationships could be represented in a tree structure. The MAGE-ML XML file is one specific example of a suitable database relationship document 125 in accordance with the preferred embodiments. Any type of document that provides any information regarding one or more relationships in the database is within the scope of the database relationship document 125.
- [056] The user selects a database column from the displayed relationships (step 1320). The user then selects records or defines one or more conditions for the selected database column (step 1330). If no more columns need to be added to the query (step 1340=NO), method 1300 is done. If more columns need to be added to the query (step 1340=YES), the database relationships are again displayed to the user (step 1350). The user then selects a database column from the list (step 1360). The list of possible values is then filtered according to the user's previous selections and displayed to the user (step 1370). The user then selects a database column from the filtered list (step 1380). If another column needs to be added to the query (step 1390=YES), method 1300 loops back to step 1350 and continues. If no other columns need to be added to

the query (step 1390=NO), method 1300 is done.

[057] A simple example is shown in FIGS. 14-21 that illustrates the advantages of the graphical query and data mining interface 124 of the preferred embodiments. We assume the interface 124 includes multiple tabs 1402, 1404 and 1406. The 1402 tab is selected in all of FIGS. 14-21 for the purpose of illustration, resulting in the display of a corresponding window 1410 that displays database relationship information for GeneChip Arrays (GCA) to the user. We assume that the database relationship information stored in the database relationship document 125 in FIG. 1 includes the relationships displayed graphically in window 1410, namely: a GeneChip Array Entry 1430 includes one or more Probe Set IDs 1440, which includes one or more Organism 1442, Misc 1444, and UniGene 1446. Note that Misc 1444 is representative of any suitable number of additional selections below the Probe Set ID 1440. With this database relationship tree displayed in window 1410, a user may now select the GeneChip Array Entry 1430 by clicking on the box next to it, as shown in FIG. 14. In response, the display window 1420 to the right displays the user's selection, as shown as 1460 in FIG. 15. Note that the GeneChip Array Entry 1460 includes a drop-down list that displays all of the GeneChip Array Entries in the database. For this example, we assume that there is a single GeneChip Array Entry named Human Genome U95B that appears in the drop-down list, so we assume the user selects the Human Genome U95B, as shown by the X in the box to the right of the selection in the drop-down list in FIG. 15.

[058] Now that the user has selected the desired GeneChip Array Entry in the database, the user may now go back to the relationship diagram in window 1410 and click on the Probe Set ID box, as shown in FIG. 16. In response to the user selecting the Probe Set ID box, the Probe Set ID 1470 is displayed in the window 1420 to the right in FIG. 17. This Probe Set ID 1470 includes a drop-down list. Note, however, that the drop-down list does not include thousands of Probe Set IDs, as depicted in the prior art in FIG. 10. Instead, the list of Probe Set IDs is filtered to display only those Probe Set IDs that correspond to the Human Genome U95B that the user previously selected. This is one of the extremely powerful aspects of the graphical query and data mining interface 124 of the preferred embodiments. With each selection the user makes in window 1420, the future possibilities are filtered according to those selections already made. As a result, the user gets a visual indication of the quality of the query before the query is executed. This also allows for data mining since the graphical indication helps direct the user to certain data.

[059] We assume the user selects the Probe Set ID 53377_at from the drop-down list, as shown in FIG. 17. Now the user may go back to the relationship diagram in window 1410 and click on the UniGene item 1446, as shown in FIG. 18. When the UniGene

item is clicked in FIG. 18, a UniGene 1480 is displayed in the window 1420 of FIG. 19. Note that the UniGene 1480 includes a drop-down list that is filtered to show only the UniGenes that satisfy the two earlier user selections above (namely, GeneChip Array Entry = Human Genome U95B, and Probe Set ID = 53377_at). There may be thousands or millions of UniGenes, but filtering the list according to the previous selections narrows the list to a single UniGene, namely Hs_366575, as shown in FIG. 19. We assume the user selects Hs_366575 as shown in window 1420 of FIG. 19. At this point the user decides the query is complete, and clicks on the Display button 1450, as shown in FIG. 20. The query results are then displayed in a separate window 1490, as depicted in FIG. 21.

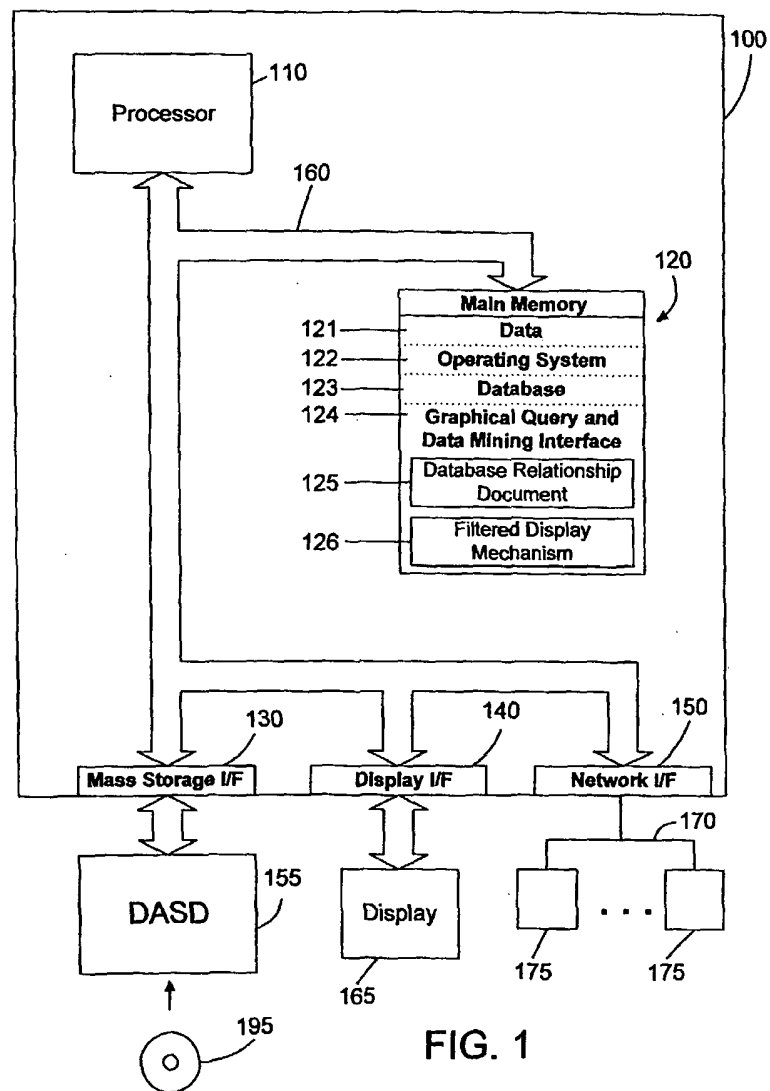
[060] The great power provided by the graphical query and data mining interface in accordance with the preferred embodiments comes from displaying database relationship information to the user in a way that does not affect the data abstraction provided by the graphical query interface, and from filtering displayed results according to previous user selections. In this manner the user is provided with a graphical indication of the quality of the query as the query is constructed. Thus, if a selection would cause no data to be returned, the user will have a visual indication of this while the query is being built, and can then make appropriate changes to the query to retrieve valid data. In addition, if a query would cause too many records to be returned, the user will generally have a visual indication of this before the query is even executed, thereby allowing the user to modify the query to narrow it down to retrieve a suitable number of records.

[061] One skilled in the art will appreciate that many variations are possible within the scope of the present invention. For example, it is within the scope of the preferred embodiments to actually run a partial query in a background process as the user continues to build the query, and to display a number of rows that the query returns. In this manner the user will have information retrieved from the database regarding the number of rows the query will return in making future selections when continuing the construction of the query.

Claims

- [001] A method comprising the steps of: providing a graphical user interface for a user to build a database query; and the user building the query using the graphical user interface, during which the graphical user interface provides to the user information regarding quality of the query before the query is executed.
- [002] The method of claim 1 wherein the graphical user interface provides a display to a user of at least one relationship in the database; and, in response to a user selection when building the query, the graphical user interface filters the display of information to the user according to at least one previous user selection.
- [003] The method of claim 2 wherein the display of information to the user comprises at least one item in the database that the user may select.
- [004] The method of claim 2 further comprising the step of determining the at least one relationship in the database by reading a database relationship document corresponding to the database.
- [005] The method of claim 4 wherein the database relationship document comprises an XML document.
- [006] The method of claim 2 further comprising the step of constructing the query in a series of steps, wherein the information displayed to the user is filtered at each step according to all user selections in all previous steps.
- [007] The method of claim 2 further comprising the step of displaying a number of rows in the database that satisfy the query.
- [008] The method of claim 1 wherein the graphical user interface displays in a first window at least one relationship in the database specified in a database relationship document; and displays in a second window information that is filtered according to at least one previous user selection, wherein the information comprises at least one item in the database that the user may select.
- [009] The method of claim 8 comprising the step of displaying in a third window results of executing the query.
- [010] The method of claim 9 wherein the results of executing the query include a number of rows that satisfy the query.
- [011] The method of claim 8 wherein the database relationship document comprises an XML document.
- [012] The method of claim 8 further comprising the step of constructing the query in a series of steps, wherein the information displayed to the user is filtered at each step according to all user selections in all previous steps.
- [013] An apparatus comprising means adapted for carrying out all the steps of the method according to any preceding method claim.

- [014] A computer program comprising instructions for carrying out all the steps of the method according to any preceding method claim, when said computer program is executed on a computer system.



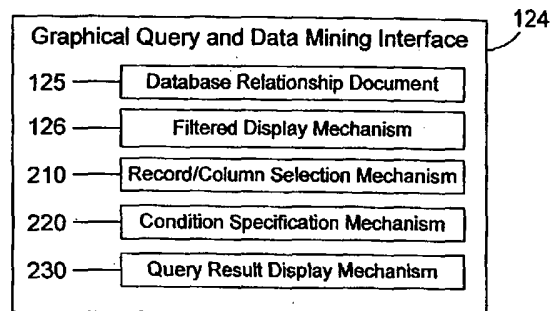
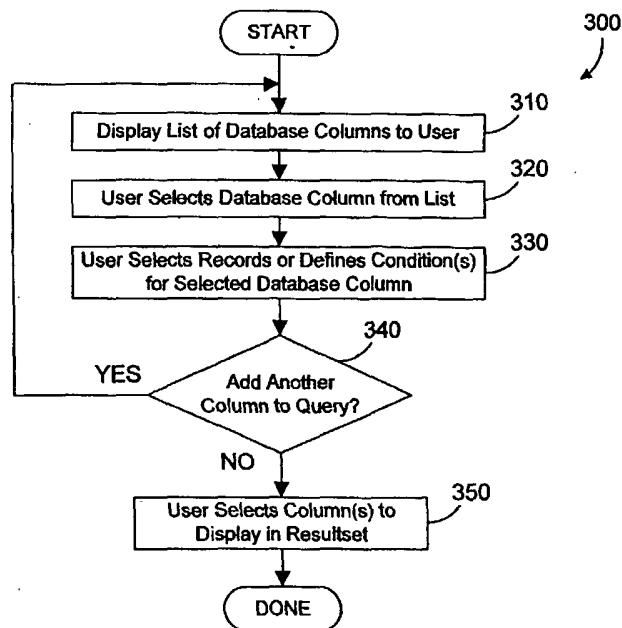


FIG. 2

FIG. 3 Prior Art

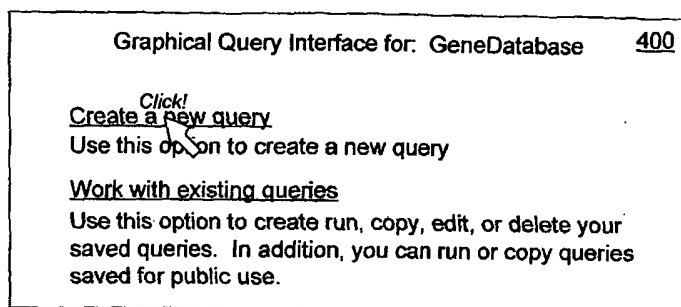


FIG. 4 Prior Art

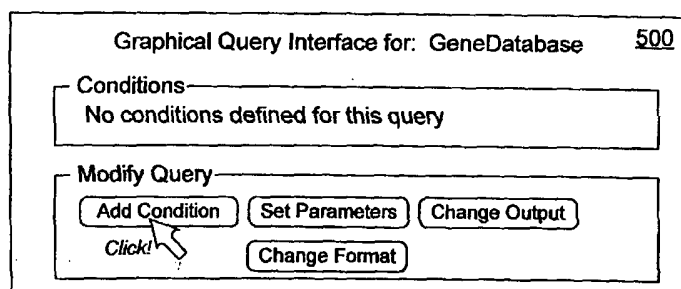


FIG. 5 Prior Art

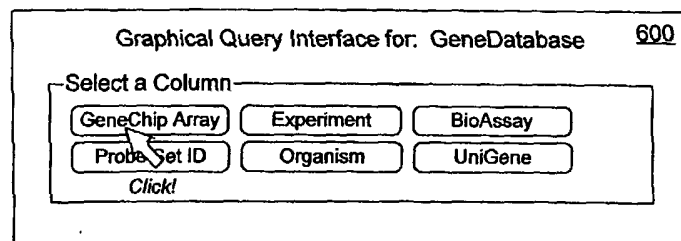


FIG. 6 Prior Art

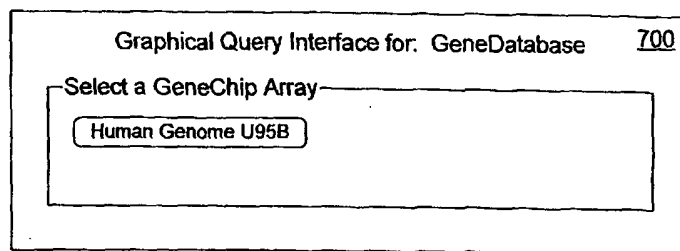


FIG. 7 Prior Art

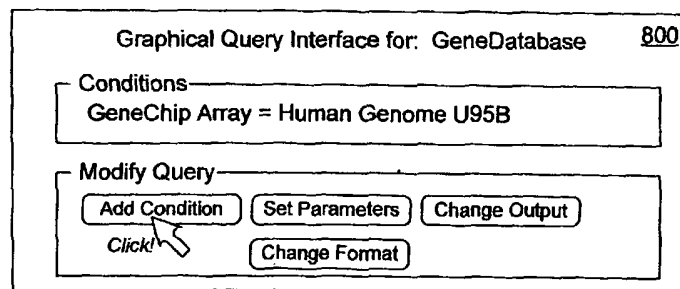


FIG. 8 Prior Art

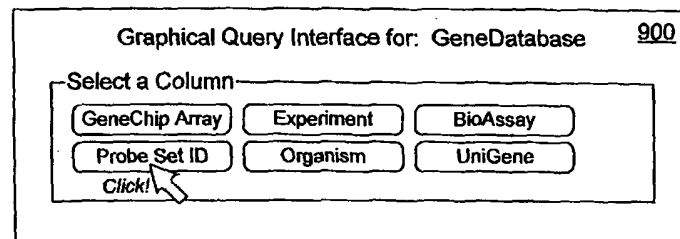


FIG. 9 Prior Art

Graphical Query Interface for: GeneDatabase 1000

Select a ProbeSet ID—

1010

45995_at	45998_at	45999_at	46002_rf
46002_rdz	46005_bc	46010_kj	46345_fw
46458_qa	46455_tg	46459_pm	46488_ng
46502_rt_a	46593_btv	46690_cx	46784_lh
46855_qk	46872_ub	46875_fs	46933_hw
46991_cy	47038_ph	47125_xfd	47286_ur
47301_hs	47456_ij	47513_mx	47788_pl
47826_jw	47943_op	48034_wa	48191_nv
48330_wm	48415_kz	48529_ud	48600_qy
48720_jx	48947_lw	49022_dc	49111_her
49233_ps	49312_kk	49441_hy	49522_fc
49710_sx	49889_lj	49954_hy	50031_ws
50157_lk	50246_aq	50399_dc	50408_ss
50526_luw	50643_qp	50810_xm	50927_tt
51015_hg	51112_md	51234_ed	51366_ut
51560_yi	51691_jd	51743_pa	51847_by
51929_so	52028_nc	52169_mo	52202_ls
52444_ur	52522_rv	52680_lm	52771_pm
52805_oc	53071_wh	53112_tg	53229_vv
53377_at	53384_qat	53497_oo	53565_zr
53720_poj	53879_wl	53980_mo	54120_rk

1030

1020

FIG. 10

Prior Art

Graphical Query Interface for: GeneDatabase

Conditions

GeneChip Array = Human Genome U95B
Probe Set ID = 533777_at

Modify Query

Add Condition Set Parameters Change Output
Change Format Click!

FIG. 11 Prior Art

Graphical Query Interface for: GeneDatabase 1200

Select Column(s) to Display in Resultset

GeneChip Array Experiment BioAssay
Probe Set ID Organism UniGene
Click!

FIG. 12 Prior Art

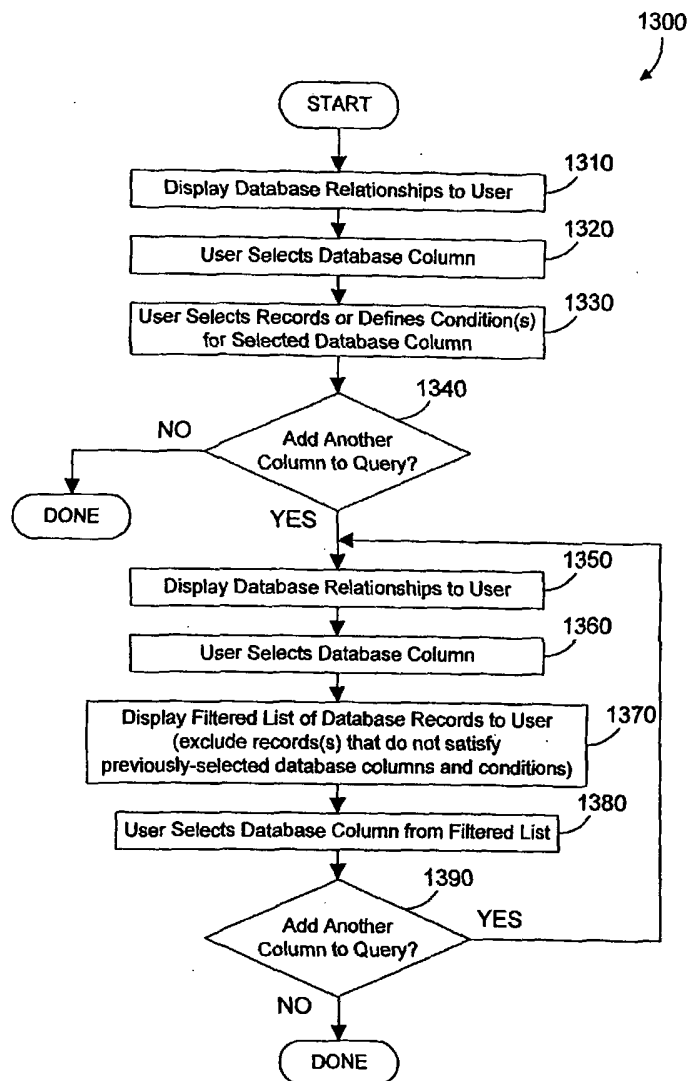


FIG. 13

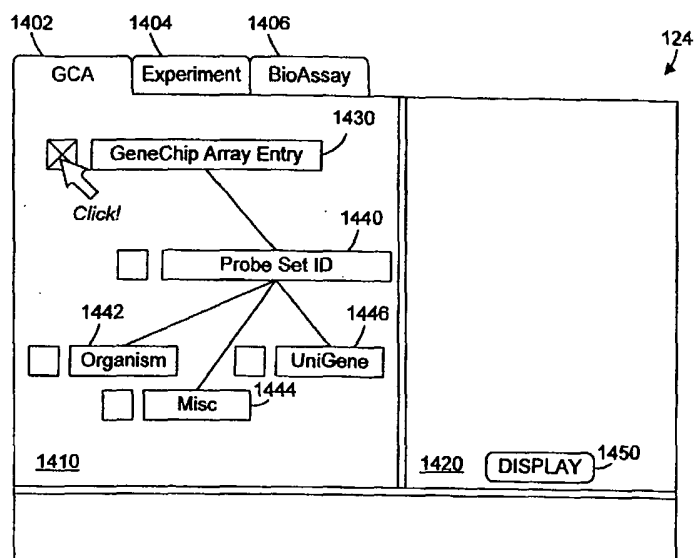


FIG. 14

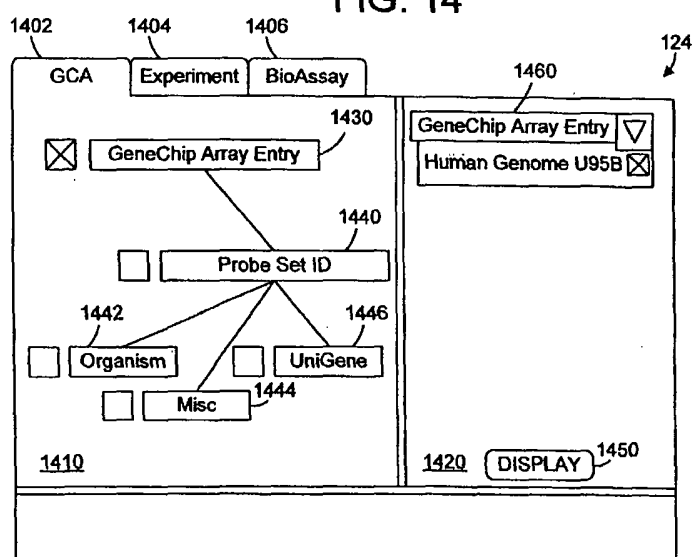
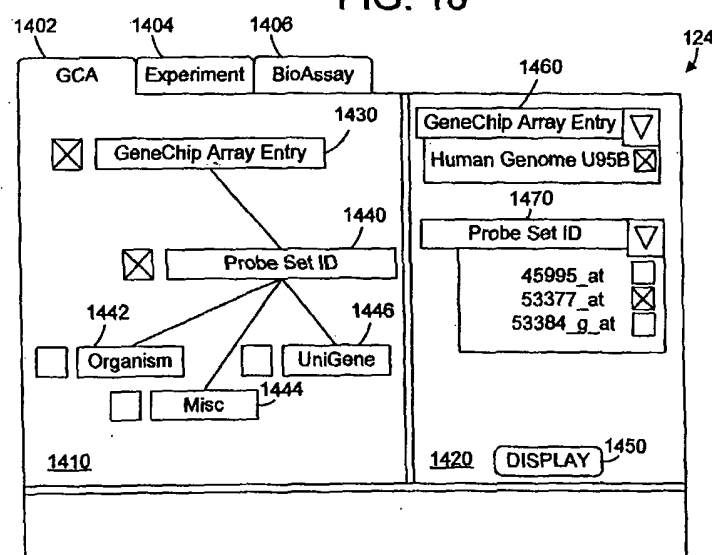
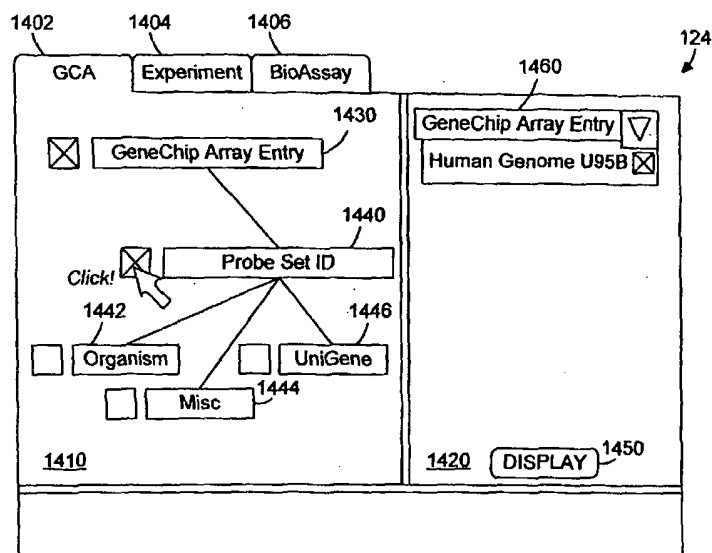


FIG. 15

9/11



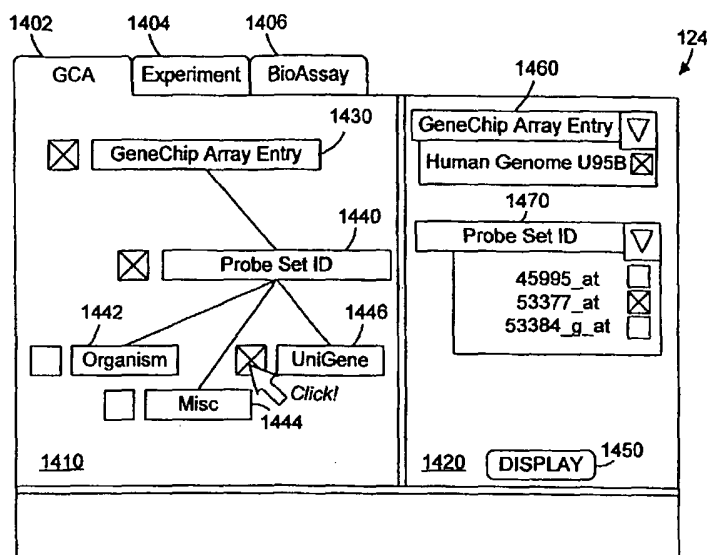


FIG. 18

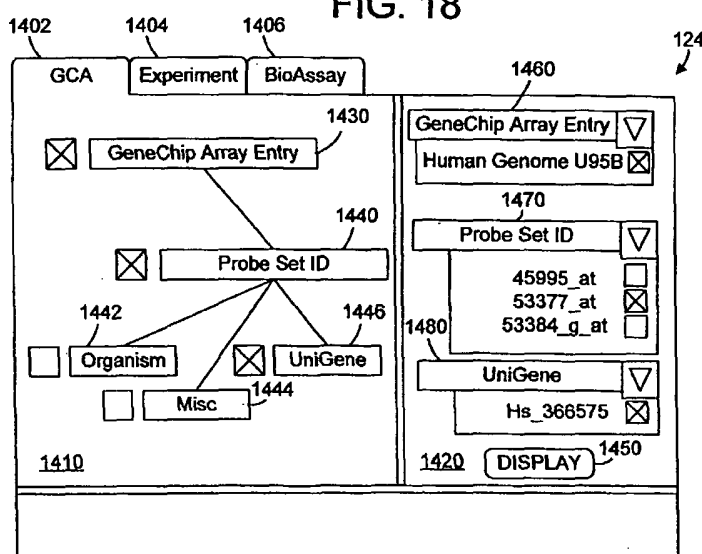


FIG. 19

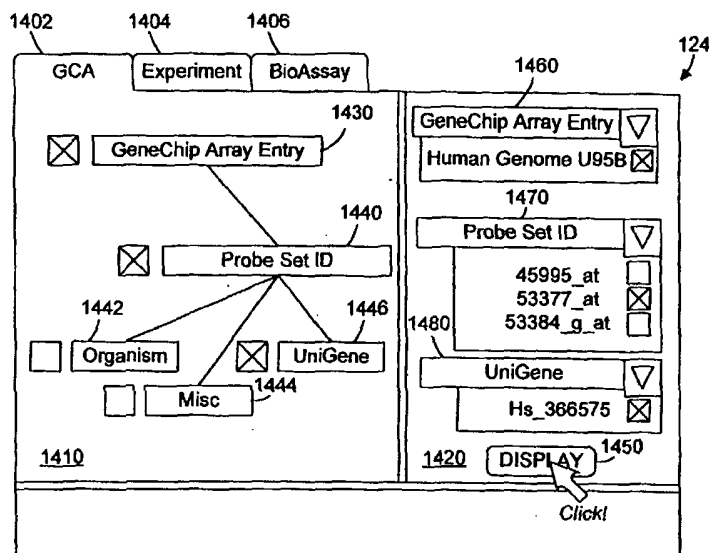


FIG. 20

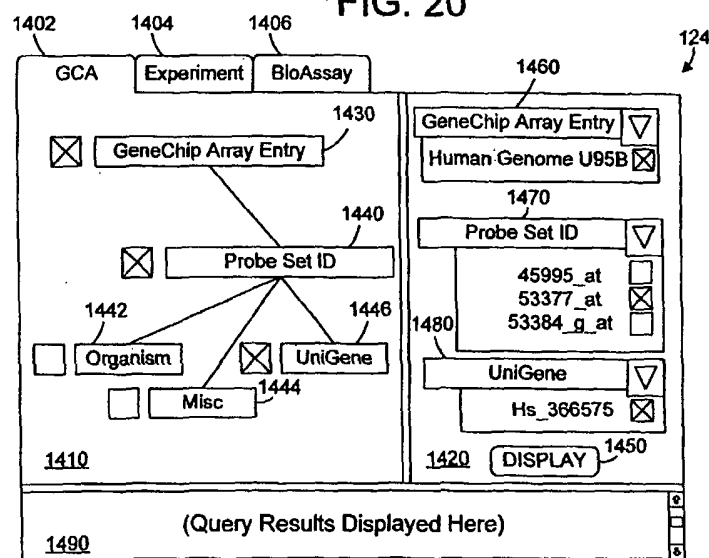


FIG. 21

INTERNATIONAL SEARCH REPORT

International Application No.

PCT/EP2005/052117

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F17/30

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)

IPC 7 G06F

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

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

EPO-Internal, WPI Data, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 331 579 A (ROLLS-ROYCE PLC) 30 July 2003 (2003-07-30) column 9, lines 25-34, 51-54, paragraphs 14, 25, 33, 50, 58 -----	1-14
X	US 2003/069882 A1 (NIESWAND BENNO ET AL) 10 April 2003 (2003-04-10) paragraphs '0020! - '0027! -----	1-14
A	US 2003/212669 A1 (DEDHIA AATISH ET AL) 13 November 2003 (2003-11-13) abstract -----	1-14



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents :

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

E earlier document but published on or after the international filing date

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

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

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

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

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

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

Z document member of the same patent family

Date of the actual completion of the international search

10 August 2005

Date of mailing of the international search report

26/08/2005

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
 Fax: (+31-70) 340-3016

Authorized officer

Haffner, R

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/EP2005/052117

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 1331579	A	30-07-2003	EP 1331579 A2	30-07-2003
			JP 2003337832 A	28-11-2003
			US 2003154205 A1	14-08-2003
US 2003069882	A1	10-04-2003	EP 1300776 A1	09-04-2003
US 2003212669	A1	13-11-2003	NONE	