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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 November 2001 (29.11.2001)

PCT

(10) International Publication Number
WO 01/90948 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US01/16258

(22) International Filing Date: 21 May 2001 (21.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/576,336 22 May 2000 (22.05.2000) US

(71) Applicant *(for all designated States except US)*: **DIGITAL GENE TECHNOLOGIES, INC.** [US/US]; Suite 110, 11149 North Torrey Pines Road, La Jolla, CA 90237 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **BODZIN, Leon, J.** [US/US]; 11832 Meriden Lane, San Diego, CA 92128-4336 (US).

(74) Agent: **LESAVICH, Stephen**; McDonnell Boehnen Hulbert & Berghoff, Suite 3200, 300 South Wacker Drive, Chicago, IL 60606 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published:**
— *without international search report and to be republished upon receipt of that report*

*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.*

WO 01/90948 A2

(54) Title: METHOD AND SYSTEM FOR MULTIPLE DEPLOYMENTS OF ELECTRONIC INFORMATION

(57) Abstract: A method and system for multiple deployment of electronic information in an electronic data warehouse. New deployments of electronic information are built into a file system for an electronic data warehouse on a network device on a computer network. A deployment includes electronic information as well as other components (e.g., tools, action scripts, etc.) that are used to process, manipulate and/or provide access to the electronic information. Access to a deployment is provided with an access view. The access view includes a list of components from the deployment that can be accessed via the access view. The access view is used to switch from a first access point in a hierarchical structure for a first deployment to a second access point in a hierarchical structure for a second deployment. Switching between deployments allows electronic information to be compared between deployments. The methods and system described herein may be used as new bioinformatic techniques to manipulate and store multiple deployments of biotechnology data used for simultaneous sequence-specific identification of expressed genes.

# METHOD AND SYSTEM FOR MULTIPLE DEPLOYMENTS OF ELECTRONIC INFORMATION

## FIELD OF THE INVENTION

This invention relates to storing electronic information. More specifically, it relates to a method and system for creating and manipulating multiple deployments of electronic information in an electronic data warehouse.

## BACKGROUND OF THE INVENTION

A "data warehouse" is a information storage system that is used to store and access large amounts of electronic information. A data warehouse is typically used to store large amounts of experimental data collected from experiments (e.g., biotechnology experiments), and other types of data including sales data, business data, financial data. A data warehouse may also be used to store large amounts of other types of electronic information including electronic content (e.g., audio, video, graphical images, etc.).

A data warehouse is often used to provide electronic information that is displayed using one or more bioinformatic techniques. As is known in the art, "bioinformatics" includes the systematic development and application of information technologies and data mining techniques for processing, analyzing and displaying data obtained by experiments, modeling, database searching, and instrumentation to make observations about biological processes.

A data warehouse can be distributed over several computers or computer networks. A data warehouse typically includes one or more servers, databases, file systems, or other storage mediums used to store data from numerous sources. Data in a data warehouse may be stored in a variety of formats. A data warehouse is typically accessed through one or more access points such as file system entry points, via one

or more servers.

Access to a data warehouse is typically transparent to a user, who uses simple commands to retrieve and analyze data stored in a warehouse. The data warehouse also includes information about how the warehouse is organized, where the data can be found in the warehouse, and any connections or associations between data. The data warehouse also allows an organization to organize its data, coordinate updates, and determine relationships between data gathered from different parts of an organization.

There are several problems associated with using data warehouses. Resource mappings are typically used to locate data in a data warehouse. The resource mappings may include Uniform Resource Locators ("URL"). As is known in the art, a Uniform Resource Locator specifies the protocol to be used in accessing a resource, a name of a server on which the resource resides and, optionally, a path to a resource (such as a directory, an electronic document or a file on the specified server).

However, installing and configuring such resource mappings typically requires manual procedures for initial configurations and updates. Resource mappings in data warehouses may change frequently. Keeping resource mappings up to date is costly and typically requires a large amount of support from administrative personal on a regular basis.

Another problem is that servers, such as Hyper Text Transfer Protocol ("HTTP") servers, are often used to manage resource mappings in a data warehouse. Servers that manage resource mappings interpret a request for data and direct the request to an appropriate location and device in a data warehouse. Servers also to have to be updated frequently, and this updating process also requires administrative

support.

Additionally, servers are typically incapable of performing "partial redirects." As is known in the art, a "partial redirect" uses a resource path relative to a desired mapping. Typically servers, such as HTTP servers only provide resolution of paths relative to a server's primary directory. Partial redirects are desirable because they enable distribution of resources at a file system level in coordination with one or more servers that actually serve the resource.

Another problem is that at a file system level, softlinks and mount points are typically used to distribute a data access load. As is known in the art, a "softlink" is a symbolic reference for data stored in a first location (e.g., a table) but provides an actual reference for the data stored in a second location (e.g., a file system). As is known in the art, a "mount point" is a physical connection point at which a device, such as a server, database, etc. is made available in a data warehouse. However, servers that do not provide partial redirects have trouble using multiple mount points. For example, it is often desirable to partition development activities and production activities into physically separate file systems in a data warehouse. An organization's production activities can proceed in a stable environment while additional development on data stored in a data warehouse can proceed without interfering with production activities.

Another problem is that it is often difficult to resolve a resource mapping in a data warehouse. Resolvers are often used in a data warehouse. One such resolver known in the art is the Persistent Uniform Resource Locator ("PURL"). This resolver was proposed by the Online Computer Library Center. Information about the Online Computer Library Center can be found at the URL "www.oclc.org." More information on the PURL resolver can be found at the URL "purl.oclc.org."

Functionally, the PURL resolver is simply a URL. However, instead of pointing directly to the location of an Internet resource, a PURL points to an intermediate resolution service. The PURL resolution service associates the PURL with the actual URL and returns that URL to requesting client. The client can then complete the URL transaction in the normal fashion. On the Internet, this is a standard "redirect".

This resolver was developed for providing a means of managing persistent names for items that move around on the Internet. These persistent names can be used in cataloging general records and general electronic documents. However, this resolver can only be used to locate resources on the Internet and was not intended to be used for resolving filepaths in a data warehouse. Another deficiency in this resolver includes the inability to resolve user-oriented resolution services and is used only from an end-user client and not from other servers.

Another problem with a data warehouse is that physical filepaths typically cannot be partitioned or resolved across file systems. In addition URLs typically cannot be dynamically partitioned or resolved across multiple networks. As a result, data typically cannot be distributed across multiple file systems or multiple networks to balance computing resources required for managing resource mappings against computing resources required for input/output processing to read/write data to/from a data warehouse.

Another problem is that a data warehouse typically is used only to store data. It may also be desirable to store other components such as software utilities that create input or output used by other components in data warehouse. It may also be desirable to store components that process raw data into processed data in a data warehouse along with the raw and processed data. However, data warehouses known in the art

typically do not allow such other components to be stored in a data warehouse.

Another problem is that a data warehouse may be used by multiple users, multiple developers and include multiple components. A user should be provided with a unique view of all or only selected components in a data warehouse. A developer should be provided with a unique view to develop, test and debug components of a data warehouse. A component may be allocated to a unique computing resource (e.g., a specified server). As is known in the art, there have been attempts to provide a unique view of data on a computer network. See for example, "Integrating SiteMinder & NetDynamics," Netegrity White Paper, December 22, 1997, "How to Secure Access for E-Commerce Portals," Netegrity White Paper, February 1999, and "SiteMinder Delivers Industry-Leading Performance, Scalability and Reliability," Netegrity White Paper, December, 1999, available at the URL "www.netegrity.com." However, the view access techniques described still do not solve all of the problems associated with providing access to selected components in a data warehouse.

Thus, it is desirable to provide a data warehouse with multiple components that can be used by multiple users and multiple developers. The data warehouse should include a flexible electronic framework to store electronic information including experimental data.

## SUMMARY OF THE INVENTION

In accordance with preferred embodiments of the present invention, some of the problems associated with data warehouses are overcome. A method and system for multiple deployment of electronic information in an electronic data warehouse is provided.

One aspect of the present invention includes a method for creating an electronic framework for deploying electronic information in a data warehouse. Another aspect of the present invention includes a method for viewing electronic information stored in a deployment in an electronic framework. Another aspect of the invention includes a method for switching between deployments stored in an electronic framework.

The methods and system described herein allow new deployments of electronic information to be built into a file system for an electronic data warehouse on a network device on a computer network. A deployment is a distinct body of electronic information that exists in a hierarchy in a file system in parallel with other deployments, if any. A deployment includes electronic information as well as other components (e.g., tools, action scripts, etc.) that are used to manipulate and/or provide access to the electronic information.

Access to a deployment is provided with an access view. The access view includes a list of components from the deployment that can be accessed via the access view. The access view may be used to switch between deployments. The access view may be used to switch from a first access point in a hierarchical structure for a first deployment to a second access point in a hierarchical structure for a second deployment. Switching between deployments using an access view allows electronic information to be compared between deployments.

The methods and system described herein may be used as a new bioinformatic technique to create, manipulate and provide access to multiple deployments of biotechnology data used for simultaneous sequence-specific identification of expressed genes. However, the methods and system are not limited for use as bioinformatic techniques can be used to create, manipulate and provide access to an electronic data warehouse for virtually any type of electronic information.

The foregoing and other features and advantages of preferred embodiments of the present invention will be more readily apparent from the following detailed description. The detailed description proceeds with references to the accompanying drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the present invention are described with reference to the following drawings, wherein:

FIG. 1 is a block diagram illustrating an exemplary data storage system;

FIG. 2 is a flow diagram illustrating a method for creating an electronic framework for deploying electronic information;

FIG. 3 is a block diagram illustrating an exemplary deployment of electronic information;

FIG. 4 is a block diagram illustrating an exemplary data flow for management activities performed on a deployment;

FIG. 5 is a flow diagram that illustrates a method for viewing electronic information deployed into an electronic framework;

FIG. 6 is a block diagram that illustrates an exemplary Access Control List Resolver;

FIG. 7 is a block diagram that illustrates an exemplary user access model based on an Agent-to-Domain relationship for the Access Control List Resolver;

FIG. 8 is a block diagram visually illustrating an exemplary access view for the method of FIG. 5; and

FIG. 9 is a flow diagram that illustrates a method for switching between deployments stored in an electronic framework.

## DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

### Exemplary data storehouse system

FIG. 1 is a block diagram illustrating an exemplary data storehouse system 10 for an embodiment of the present invention. The exemplary data storehouse system 10 includes electronic information 12 that will be stored in a data storehouse 14. The electronic information includes, but is not limited to, experimental data collected from experiments (e.g., biological experiments), organizational data, business data, financial data, etc. Data storehouse 14 includes a data warehouse 16 and an access control system 18. The data warehouse 16 is used to store raw and processed electronic information 12. It is also used to store multiple types of components (e.g., tools, action scripts, etc.) that are used to manipulate and/or provide access to the electronic information 12 as is explained below. The access control system 18 is used to provide access to the data warehouse 16 and to resolve requests for electronic information from deployments in the data warehouse 16 as is explained below. The data storehouse 14 exists on, and is accessed via a computer network 20 (e.g., the Internet, an intranet, other Local Area Network ("LAN"), Network File System, ("NFS"), etc.) The data storehouse 14 is connected to a data viewer 22 that is used to request access and to view electronic information 12 from the data storehouse 14.

In one embodiment of the present invention, the data viewer 22 includes a network browser including Internet Explorer by Microsoft Corporation of Redmond, Washington, Netscape Navigator by Netscape Corporation of Mountain View, California, and others. In another embodiment of the present invention, the data viewer 22 can also be a hardware or software application or component other than a network browser. However, the present invention is not limited to the components described for the data storage system 10, and more, fewer or equivalent components

can also be used.

An operating environment for components of the data storage system 10 include a processing system with one or more high speed Central Processing Unit(s) ("CPU") and a memory. The CPU may be electrical or biological. In accordance with the practices of persons skilled in the art of computer programming, the present invention is described below with reference to acts and symbolic representations of operations or instructions that are performed by the processing system, unless indicated otherwise. Such acts and operations or instructions are referred to as being "computer-executed" or "CPU executed."

It will be appreciated that acts and symbolically represented operations or instructions include the manipulation of electrical signals or biological signals by the CPU. An electrical system or biological system represents data bits which cause a resulting transformation or reduction of the electrical signals or biological signals, and the maintenance of data bits at memory locations in a memory system to thereby reconfigure or otherwise alter the CPU's operation, as well as other processing of signals. The memory locations where data bits are maintained are physical locations that have particular electrical, magnetic, optical, or organic properties corresponding to the data bits.

The data bits may also be maintained on a computer readable medium including magnetic disks, optical disks, organic memory, and any other volatile (e.g., Random Access Memory ("RAM")) or non-volatile (e.g., Read-Only Memory ("ROM")) mass storage system readable by the CPU. The computer readable medium includes cooperating or interconnected computer readable medium, which exist exclusively on the processing system or be distributed among multiple interconnected processing systems that may be local or remote to the processing

system.

**An electronic framework for deploying electronic information**

FIG. 2 is a flow diagram illustrating a Method 24 for creating an electronic framework for deploying electronic information. At Step 26, a new deployment of electronic information is built into a file system on a network device on a computer network. A "deployment" is a distinct body of electronic information that exists in a hierarchy in a file system in parallel with other deployments, if any. At Step 28, pre-determined portions of electronic information from a previous deployment of electronic information, if any, are migrated into the new deployment of electronic information. At Step 30, management activities are performed on the new deployment of electronic information to allow the new deployment of electronic information to be accessed from the computer network. At Step 32, executable computer software from the new deployment of electronic information is instantiated. The executable computer software is used to process and access electronic information in an electronic data warehouse including a plurality of deployments of electronic information.

In one specific embodiment of the present invention, at Step 26, a new deployment of electronic information is built into a file system on a network device (e.g., a network computer) on a computer network. Each deployment is referred to by a moniker and is made available in a consistent manner from a primary data repository in the data warehouse 16. In such an embodiment, Step 26 includes building the deployment with a hierarchical arrangement with a primary tier and a secondary tier. The primary tier includes entry points into a deployment and the secondary tier includes electronic information for the deployment. However, the present invention is not limited to such a hierarchical arrangement and other or

equivalent hierarchies, can also be used.

FIG. 3 is a block diagram illustrating an exemplary deployment of electronic information 34. The deployment 34 includes a hierarchical arrangement with a primary tier 36 and a secondary tier 38 illustrated visually in FIG. 3. The primary tier 36 is made up of primary data warehouse 16 entry points 40, 42 (two of which are illustrated) that are used by multiple components that make up the data storage system 10. The entry points include multiple entry points to multiple data repositories 44, 46. The multiple data repositories are accessed via one primary repository entry point 48. However, the present invention is not limited to the hierarchy or the entry points described and other or equivalent hierarchies or data entry points can also be used.

A first deployment includes a distinct body of electronic information that exists in a file system in parallel with other deployments under the secondary tier 38 in the hierarchical arrangement, if any. One purpose of this secondary tier 38, or deployment 50, is to provide a uniform layout for electronic information in a deployment. Another purpose of this secondary tier 36 is to enable separate developers to work on server-based software and system utilities, sometimes in cooperation with other developers; and to keep a version of production software, system utilities and data stable for users. FIG. 3 illustrates a first deployment 50 of electronic information with solid lines including data repositories 44 and 46 in a first file system.

At Step 28, pre-determined portions of electronic information from a previous deployment of electronic information, if any, are migrated into the new deployment of electronic information. In one embodiment of the present invention, Step 28 includes creating new data structures and migrating electronic information from old data structures used for a previous deployment into the new data structures for a new

deployment.

FIG. 3 illustrates a new or second deployment 52 of electronic information created at Step 28. New electronic information in the second deployment 52 is illustrated with dashed lines, while electronic information migrated from the first deployment 50 is illustrated with solid lines.

As can be seen from FIG. 3, an exemplary new data repository 54 and a new data repository 56 in the secondary tier 38' include some, but not all of the original information from data repositories 44 and 46. New electronic information in the new or second deployment is illustrated with dashed lines while old or previous electronic information from the old or previous is illustrated with solid lines.

The primary tier 36' for the second deployment 52 is made up of primary data warehouse 16 entry points 58, 60 (only two of which are illustrated). The entry points include multiple entry points to multiple data repositories 54, 56. The multiple data repositories 44, 46, 54, 56 for the first deployment 50 and the second deployment 52 are accessed via the one primary repository entry point 48.

At Step 30, management activities are performed on the new deployment of electronic information to allow the new deployment of electronic information to be accessed from the computer network. The management activities include, but are not limited to: (1) processing raw electronic information into processed electronic information; (2) creating one or more separate data repositories with electronic information; (3) logically or physically partitioning the electronic information into one or more sub-components; (4) relocating portions of the electronic information to other network devices (e.g., servers, etc.) on a computer network; (6) creating access points to the electronic information; or (6) creating access privileges to the electronic information.

FIG. 4 is a block diagram illustrating an exemplary data flow 62 for management activities performed on a deployment at Step 30. However, the present invention is not limited to this exemplary data flow 62 and other data flow flows with more, fewer or equivalent activities can be performed at Step 30. In one example, at Box 64, raw experimental data is collected from experiments. If a biological experiment is being run, raw experimental data may be collected directly from an instrument that is collected biological data (e.g., a gel-electrophoresis instrument or a micro-array chip). At Box 66, the raw experimental data is saved in one or more raw data files. At Box 68, the raw experimental data in the raw data files is normalized and calibrated for visual viewing. For example at Step 68, raw experimental data is normalized and calibrated for visual viewing using the methods and systems described in co-pending applications No. 09/318,679 entitled "Methods for Normalization of Experimental Data," and 09/318,699, entitled "Methods and System for Amplitude Normalization and Selection of Data Peaks," assigned to the same Assignee as the present application. At Box 70, viewable images are created from the normalized and calibrated data for visual viewing. At Box, 72, a viewable image repository is created in the data warehouse 16 to store the digital images.

As another example, at Box 74 scholarly research related to the biological experiments conducted is collected from public databases (e.g., on the Internet) and private databases (e.g., on one or more intranets). At Box 76, database queries are run on the data collected. At Box 78, results from the database queries are formatted into electronic information used for visual display. At Box 80, a query repository is created in the data warehouse 16 to store the formatted electronic information.

FIG. 3 visually illustrates the management activities performed at Step 30 (FIG. 2) of logically (e.g., primary tiers 36, 36'and secondary tiers 38, 38') and

physically (e.g., the first deployment 50 and the second deployment 52) partitioning the electronic information into one or more sub-components. FIG. 3 also visually illustrates access points created for deployments (e.g., 40, 42, 48, 58, 60) performed at Step 30 (FIG. 2).

All or selected portions of the electronic information in a deployment can also be physically re-located to other network devices on the computer network 20 at Step 30 (not illustrated in FIG. 3 or 4). After a deployment or selected portions of a deployment are physically relocated, one or more "soft-links" or mount points are created to access electronic information as is explained below.

At Step 32, executable computer software from the new deployment of electronic information is instantiated. The executable computer software is used to process and access electronic information in the data warehouse 16 including multiple deployments of electronic information (e.g. 50, 52 of FIG. 3).

For example, the executable computer software includes computer source code translated into machine code in a format that can be loaded into memory and run by a device's central processor (e.g., complied C, C++, etc.). For computer software that is interpreted, the executable computer software includes products created source code in the proper format (e.g., products created from scripts such as PERL, JavaScript, etc.).

In one exemplary embodiment of the present invention, the executable computer software instantiated includes instantiating Hyper Text Markup Language ("HTML") or eXtensible Markup Language ("XML") files using Common Gateway Interface ("CGI") scripts written in the PERL scripting language.

As is known in the art, HTML and XML are hardware independent markup languages used to create and display electronic data. HTML and XML uses mark-up

tags to mark elements, such as text and graphics, in an electronic document to indicate how the elements should be displayed to a user and how the elements should respond to user actions such as activation of a link by a key press or mouse click.

As is known in the art, a CGI script is an application that is executed by a server in response to a request by a client, such as an application program. CGI scripts can be written in many programming languages, including C, C++, Visual Basic, PERL, JavaScript, etc. CGI scripts don't necessarily need to be scripts; they can also be batch programs or compiled programs.

In such an embodiment, the CGI scripts are used to generate, for example, a home page, and a user project resource page used on the data viewer 22 (FIG. 1) (e.g., HTML or XML pages). The CGI scripts may also be used to generate applications used with a viewing application (e.g., a browser) in connection with data viewer 22. For example, the applications may, include, functions such as "SAVE," "DELETE," "OPEN," etc. that are available from the viewing application. The HTML pages and viewing applications allow a user to view electronic information 12 stored in the data warehouse 16 via the access control system 18, using the data viewer 22.

The CGI scripts may also be used to create a number of different viewing application logs and other executable utility applications, applications programs or software tools used to manipulate or view the electronic information in a deployment.

**Deployments of biotechnology data**

In one exemplary preferred embodiment of the present invention, Method 24 is used to create, manipulate and store multiple deployments of biotechnology data used for simultaneous sequence-specific identification of expressed genes. In such an embodiment, Method 24 is used as a bioinformatic technique. However, the present invention is not limited to creating, manipulating and storing deployments of

biotechnology data, and Method 24 can be used for virtually any type of electronic information or electronic data (e.g., telecommunications data, electrical data, optical data, physical data, other data, etc.).

As is known in the biotechnology arts, Deoxyribonucleic acid ("DNA") is a double-stranded heteropolymer that can be thought of symbolically as paired complementary continuous strings of four nucleotide base elements, deoxyAdenosine, deoxyCytidine, deoxyGuanosine, and deoxyThymidine. The four bases are usually abbreviated as "A," "C," "G" and "T" respectively, and base elements on one strand of DNA interact via hydrogen bonds with a complimentary counterpart on the other strand. For example, an "A" can only interact with a "T," and a "G" can only interact with a "C." This relationship is called "base pairing." "Genes" are regions of DNA, and "proteins" are the ultimate products encoded by genes. Proteins are built from a fundamental set of amino acids, and DNA carries amino-acid coding information. When DNA is replicated or copied, a new DNA strand is synthesized using each of the original strands as templates.

DNA itself does not act as a template for protein decoding or synthesizing. A complementary copy of one of the two strands of DNA is synthesized out of corresponding ribose nucleotides to generate a ribonucleic acid ("RNA") copy of a gene with a method called "transcription."

The form of mRNA is very precise and always starts at one precise nucleotide and ends exactly at another. Complementary DNA ("cDNA") is an exact, double-stranded DNA copy of mRNA. One of the cDNA strands is complementary to the mRNA, and the other is identical.

The RNA copy of a gene is then decoded by protein synthesis with a method called "translation." The form of RNA that provides the template for protein synthesis

is called messenger RNA ("mRNA").

RNA isolated from a target cell or tissue of an organism (e.g., a cell to which a new drug has been applied) is analyzed using a method of simultaneous sequence-specific identification of mRNAs. In one preferred embodiment of the present invention, isolated RNA is analyzed using a method of simultaneous sequence-specific identification of mRNAs known as TOGA (TOtal Gene expression Analysis) described in Sutcliffe, J.G., et al *Proc Natl Acad Sci U S A* 2000 Feb 29; 97(5):1976-1981, International published application PCT/US99/23655, U.S. Patent No. 5,459,037, U.S. Patent No. 5,807,680, and U.S. Patent No. 6,030,784, all of which are hereby incorporated herein by reference.

However, other methods can also be used to provide sequence-specific identification of mRNAs, and the present invention is not limited to TOGA sequence-specific identification of mRNAs.

In one preferred embodiment of the present invention, preferably, prior to the application of the TOGA method or other methods, the isolated RNA is enriched to form a starting polyA-containing mRNA population by methods known in the art. In such a preferred embodiment, the TOGA method further comprises an additional Polymerase Chain Reaction ("PCR") step performed using one of four 5' PCR primers and cDNA templates prepared from a population of antisense complementary RNA ("cRNA", complimentary to mRNA). A final PCR step using one of a possible 256 5' PCR primers and a universal 3' PCR primer produces as PCR products, cDNA fragments that corresponded to a 3'-region of the starting mRNA population.

A label (e.g., a dye) is incorporated in the PCR products to permit detection of the PCR products by laser-induced fluorescence. Gel-electrophoresis, Capillary electrophoresis, mass spectrometry or equivalent techniques (e.g., micro-array on a

chip) are used to resolve molecules from the PCR products into distinct bands of measurable lengths. The produced PCR products can be identified by a) an initial 5' sequence comprising a nucleotide base sequence of a remainder of a recognition site or a restriction endonuclease that was used to cut and isolate a 3' region of cDNA reverse transcripts made from a mRNA population, plus the nucleotide base sequence of preferably four parsing bases immediately 3' to the remainder of the restriction enconuclease recognition site, or more preferably the sequence of the entire fragment; and b) the length of the fragment.

Whether the TOGA method is used or not, PCR product data is digitized by creating data files with digital information. (See, e.g., FIG. 4, boxes 64, 66, 68, 70, 72). The data files may include digital values, for example, of optical signal of electrophoresis patterns or other data used to identify the mRNA (e.g., data from a micro-array on a chip used to isolate the mRNA).

In one preferred embodiment of the present invention, the experimental data collected includes polynucleotide data for DNA, cDNA, RNA, cRNA, mRNA, or other polynucleotides. The polynucleotide data can include, but is not limited to, a length of a nucleotide fragment, a base composition of a nucleotide fragment, a base sequence of a nucleotide fragment, an intensity of a dye label signal used to tag a nucleotide fragment, or other nucleotide data. Method 24 can be used to store such polynucleotide data in multiple deployments in a data warehouse 16. However, the present invention is not limited to stormy polynucleotide data and can be used to store virtually any type of electronic information or experimental data.

**Viewing deployments of electronic information**

Method 24 (FIG. 2) is used to create an electronic framework for deploying electronic information. FIG. 5 illustrates a Method 82 for viewing electronic

information deployed into an electronic framework in a data warehouse 16 created with Method 24 (FIG. 2). At Step 84, a request is received to view electronic information from a deployment stored in an electronic framework in an electronic data warehouse including plural deployments of electronic information. The plural deployments include electronic information as well as software components to manipulate the electronic information. At Step 86, an access view used to provide access to the requested electronic information in the deployment is determined. The access view includes a list of components from the deployment that can be accessed via the access view. At Step 88, the determined access view is provided in response to the request to view electronic information from the deployment.

The access view provides access to view electronic information including data, tools, computer source code, or other types of electronic information. Specific electronic information in a deployment is called a "resource".

As was discussed above, normally, servers such as HTTP servers, are used to manage resource mappings. As is known in the art, HTTP is the protocol used to carry requests from a browser to a server and to transport electronic documents from server back to the requesting browser. However, HTTP servers are not capable of providing partial redirects as was discussed above. A partial redirect uses a resource path relative to a mapping, whereas HTTP servers can only provide resolution of paths relative to a server's primary directory.

Method 82 is illustrated with one specific embodiment of the present invention. However, the present invention is not limited to such an embodiment, and the present invention can be used with other embodiments.

In such a specific embodiment of the present invention, at Step 84, a request is received on the access control system 18 in the data storehouse 14 from the data

viewer 22. For example, a user "Leon", desires to view data from experiment number one from a deployment 52 stored in an electronic framework in the data warehouse 16 with data viewer 22. The user Leon is authorized to view data from an experiment number one, but not for any other experiments stored in the deployment 52.

In one embodiment of the present invention, the request is received on an Access Control List ("ACL") Resolver system 89 is included in the access control system 18. The ACL Resolver system 89 included in the access control system 18 manages literal file paths and resource mappings.

FIG. 6 is a block diagram that illustrates an exemplary ACL Resolver system 89. The ACL Resolver system 89 includes an ACL Resolver 90, multiple Agents 92, 94, and multiple resource servers 96, 98 on a computer host that includes data storage system 10.

The ACL resolver 90 is included in the access control system 18 (FIG.1). One ACL Resolver system 89 is illustrated. Each deployment includes one ACL Resolver 90.

Given a resource name, the ACL Resolver 90 provides a fully qualified path for Resources 100, 102, into the data warehouse 16. At the same time, it also provides a Uniform Resource Locator ("URL") to the same Resources 100, 102 in the data warehouse 16. The ACL Resolver 90 defines the relationships between Agents 92, 94 and Resources 100, 102, in terms of access rights stored in ACL lists 106. The ACL Resolver 90 can be used for full and partial redirects to locate resources 100, 102 or computer network 20. Thus, the ACL resolver 90 is able to resolve physical filepaths partitioned across file systems.

Agents and Resources are defined in terms of one another. That is, Agents 92, 94 are objects that access Resources 100, 102. Resources 100, 102 are objects that provide access to Agents 92, 94. An Agent may start out as a Resource when it exists

as a file located in the data warehouse 16. However, when a Resource is an executable file, upon execution it may become an Agent.

An Agent (e.g., 92) accesses other Resources (e.g., 100, 102) that are stored in the data warehouse 16. Thus, by making use of the ACL Resolver 90, an Agent is able to find Resource locations by way of a file path or URL using the Resource servers 96, 98. In one embodiment of the present invention, the properties of an Agent are included in its "Domain." A Domain is a manager of an Agent; and a manager of any other Domain included within it. However, an Agent may have more, fewer, or equivalent properties. An Agent's Domain manages the Agent.

The ability to distribute the Resources across a computer network 20 is accomplished with the help of the ACL Resolver 90 in the access control system 18. This is due to the ability to assign a name to a given Resource and have that Resource resolve to a distinct URL. Since a URL is made up of components that specify a resource server using the tuple "Host:Server," assigning a Resource to a server effectively provides a distributed system, even if all the Resources are on the same server.

The ACL Resolver 90 also defines the relationships between Resources and the data warehouse 16. In this capacity, the ACL Resolver 90 enforces an electronic framework that defines a data warehouse 16 layout. Thus, in one specific embodiment of the present invention, the ACL Resolver 90 is used to build the data warehouse 16 into an empty file system as was described above for Method 24.

The ACL Resolver 90 also allows application components such as the data viewer 22, and other tools to be developed with a uniform way. That is, individual components in a system of components can be developed while keeping a production version of the system of components intact and separate from the development

version in a multi-developer environment.

The ACL Resolver 90 handles a device separately from that of a server to allow distribution of components onto unique network computing resources. The ACL Resolver 90 can further identify a host in the context of a sub-network to enable self-containment. The ACL Resolver system 89 allows a sub-network of host devices that run one or more servers to serve applications and data in a unified server-centric system. This overall ACL Resolver 90 architecture enables full testing and profiling of one deployment against another prior to the rollout of a deployment version. The ACL Resolver 90 relies on a server (e.g., a HTTP server) for configuration requirements. However, beyond maintenance of simple primary configuration parameters the ACL Resolver 90 provides instantaneous switchover from one deployment version to another without manual intervention by a system administrator. This allows a prior deployment to be immediately used when failure occurs in a new deployment.

Returning to FIG. 5 at Step 86, an access view used to provide access to the requested electronic information for the desired experiment in the deployment 52 is determined. The access view includes a list of components from the deployment that can be accessed via the access view. In exemplary embodiment of the present invention, the ACL Resolver 90 is used to determine the access view. However, the present invention is not limited to this embodiment and the access control system 18 can use other or equivalent components to determine an access view.

FIG. 7 is a block diagram that illustrates an exemplary user-access model 110 based on an relationship used by the Access Control List Resolver 90 (FIG. 6). As was described above, a Domain is a manager of an Agent; and a manager of any other Domain included within it.

In FIG. 7, an ACCESS Domain 112 is a manager of a Root Agent 118 and a VIEW Domain 116 included within it. The VIEW Domain 116 is a manager of a Virtual Agent 122 and a WAREHOUSE Domain 120 included within it. The WAREHOUSE Domain 120 is a manager of a Repository Agent 126 and an ELEMENT Domain 124 included within it. The ELEMENT Domain 124 is a manager of a Project Agent 130 an EXPERIMENT-1 Domain 128 and an EXPERIMENT-2 Domain 132 included within it. The ELEMENT Domain 124 is illustrated with only one PROJECT Agent 130 and only two EXPERIMENT Domains 128 and 132 for two experiments one and two. However, the data warehouse 16 will typically include multiple projects and experiments (e.g., tens to hundreds or thousands of projects and experiments). In addition, the WAREHOUSE Domain 120 typically includes multiple repositories and thus, includes multiple REPOSITORY Agents 126 and multiple ELEMENT Domains 124. Additionally, an ELEMENT Domain may act as a manager for a Resource and a Tool (Not illustrated in FIG. 7).

The Access Control Lists 134, 136, 138, 140, 142 and 144 include a list of ACL-TOKENS that have access to an associated Domain. The ACL-TOKENS include, but are not limited to, authorized Agent and user names. For example, the Access Control List 138 for the WAREHOUSE Domain 120 includes ACL-TOKENS for the VIEW Domain 116, the VIRTUAL Agent 122 and a list of authorized users that can view the WAREHOUSE Domain 120 (e.g., Leon).

FIG. 7 illustrates one exemplary user-access model 110 that is used to access a deployment 52 (FIG. 3) of electronic information built into a file system of the data warehouse 16 on a network device on a computer network 20 at Step 26 of Method 24 (FIG. 2). However, other or equivalent user-access models can also be used. In this specific embodiment of the present invention, the deployment 52 includes one or

more data repositories (58,60) of electronic information with the hierarchical arrangement illustrated in FIG. 3. The hierarchy is accessed with the user-access model of FIG. 7 using Method 82 of FIG. 5. The access points (48, 58, 60) in the hierarchy of FIG. 3 for the deployment 52 are controlled with the ACL lists (134, 136, 138, 140 and 142) of FIG. 7 with the ACL Resolver 90 of FIG. 6. The Agents of FIG. 7 correspond to the Agents (92, 94) of FIG. 6, the ACL lists of FIG. 7 correspond to the ACL list 106 of FIG. 6, and the Resources (100, 102) correspond to the Domains of FIG. 7. However, the present invention is not limited to this embodiment, and other or equivalent embodiments can also be used.

Thus, at Step 88 an access view used to provide access to the requested electronic information for the desired experiment one in the deployment is determined. The access view includes a list of components from the deployment that can be accessed via the access view. In this example, one or more Access Control lists are determined by the ACL Resolver 90. For example, Access Control List 134 is read. The Access Control List will include ACL-TOKENS for the ACCESS Domain 112 and the ROOT Agent 118 and a list of all authorized users (e.g., Leon, Chuck, Jan and Bob). A USER Agent 114 for the user Leon is thus authorized to access a subset of what the ROOT Agent 118 can view. Other Access Control lists 136, 138, 140 and 142 are read to determine if Leon is authorized to view experimental data stored in the Domain EXPERIMENT-1 128 in the deployment 52.

In this example, all of the Access Control lists include Leon in the list of all authorized users and also includes the associated Agents and Domains illustrated in FIG. 7. However, in any given Access Control list, the list of authorized user may not include Leon, or the Access Control list may not include tokens for the associated Agents or Domains listed in FIG 7. As a result, access by those Agents or to these

Domains for the user Leon would be denied.

In this example, the Access Control List 142 for the EXPERIMENT-1 Domain 128 is eventually located. The Access Control List 142 includes ACL-TOKENS for the ELEMENT Domain 124 and EXPERIMENT-1 Domain 128, PROJECT Agent 130, and a list of authorized users including Leon. Thus, the USER Agent 114 for Leon has access to the experimental data for experiment-1 included in the repository 52.

In one embodiment of the present invention, Step 88 provides the determined access view multiple projects with plural experiments including experimental data. In another embodiment of the present invention, the plural projects with plural experiments include plural biotechnology projects multiple biotechnology experiments including DNA, cDNA, RNA, cRNA or mRNA experimental data. Such an embodiment of the present invention may also include at Step 88 providing the determined access view to plural resources and plural tools used to manipulate the plural resources.

Returning again to FIG. 5 at Step 88, the determined access view is provided to the data viewer 22 in response to the request to view electronic information for the desired experiment number one from the deployment 52. The access view is visually displayed by the data viewer 22. In one embodiment of the present invention, the access view is electronic content written in a mark-up language such as HTML, XML, etc. In another embodiment of the present invention, the access view is electronic content other than that written in a mark-up language that is useable by the data viewer 22.

FIG. 8 is a block diagram visually illustrating an exemplary access view 146 provided at Step 88. The access view 146 is displayed on data viewer 22. The access

view 146 includes a list of projects 148, 150 and experiments 152 that user Leon has access to. The access view 146 also includes a list of resources 154, 156, and tools 158, 160 and 162 that user Leon has access to from the data warehouse 16. For example, Leon has access to Project-1 148 including Experiment-1(A) 152 and Project-2 150. However, Leon does not have access to any experiments (1(B) – 1(N)) illustrated by the dotted lines for Project-2 150. The access view 146 typically does not display any components a user does not have access to. The dotted lines in FIG. 8 are used for illustrative purposes only. The access view 146 is used in a similar manner for access to resources and tools.

**Switching between multiple deployments of electronic information**

FIG. 9 illustrates a Method 164 for switching between deployments stored in an electronic framework. At Step 166, an electronic framework in an electronic data warehouse is provided including plural deployments of electronic information. The deployment includes electronic information and components used to manipulate the electronic information. At Step 168, a request is received for electronic information in a first deployment in the electronic data warehouse (e.g., using Step 84 of Method 82 of FIG. 5). At Step 170, access to the electronic information in the first deployment is provided using a first access view (e.g., using Step 88 of Method 82). The first access view includes a first list of components from the first deployment that can be accessed via the first access view. At Step 172, a request is received for electronic information in a second deployment in the electronic data warehouse (e.g., Step 84 of Method 82). At Step 174, the first access view is switched to a second access view. Switching from the first access view to the second access view includes switching from a first access point in a hierarchical structure for the first deployment to a second access point in a hierarchical structure for the second deployment. At Step 176,

access is provided to the electronic information in the second deployment using the second access view. The second access view includes a second list of components from the second deployment that can be accessed via the second access view.

In one embodiment of the present invention, Method 164 allows switching between plural deployments of experimental information in the data warehouse 16. The ability to virtually instantaneously switch between plural deployments of experimental information in the data warehouse 16 includes switching from a "new" deployment back to a "previous" deployment if a problem should be encountered with the new deployment. This helps ensure that a user will continue to have access to experimental information in the data warehouse 16, even if there is a problem with a new deployment in the data warehouse 16.

In one embodiment of the present invention, Step 174 includes switching from the first access view to a second access view using the ACL resolver 90. In such an embodiment, the ACL resolver 90 provides a fully qualified path and URL into the data warehouse for electronic information in a deployment. The ACL resolver 90 performs partial redirects using information from the fully qualified URL to enable access to electronic information by an Agent using the fully qualified path at the file system level in coordination with one or more servers that actually serve the electronic information.

Additionally, the ACL Resolver 90 also separates the functionality of a host and a server to allow distribution of components onto unique distributed network computing resources. The ACL Resolver can also be used to identify a host in the context of a sub-network to enable self-containment.

In one embodiment of the present invention, once a deployment is created, a copy is generated, typically including a subset of the original deployment. This

enables software developers to complete development of components of the deployment. Software developers can separate into distinguishable groups of one or more developers, and work with a deployment as a unit without disturbing a production deployment or another developer's deployment. This deployment architecture enables testing and profiling of one deployment against another prior to the rollout of a deployment version.

Method 164 allows a sub-network of host machines that run one or more servers to serve applications and electronic information from a data warehouse 16 in a "virtual" unified server-centric system. The virtual unified server-centric system may actually be distributed over plural sub-networks or plural computer networks. Method 164 also allows virtually instantaneous switchover from one deployment version to another. Method 164 provides the ability to switch back to a previous deployment when failure occurs in a new deployment.

The methods and system described herein may be used as a new bioinformatic method to manipulate and store multiple deployments of biotechnology data used for simultaneous sequence-specific identification of expressed genes. However, the present invention is not limited to such an embodiment and can be used to create, manipulate and provide access to an electronic data warehouse for virtually any type of electronic information.

It should be understood that the programs, processes, methods and system described herein are not related or limited to any particular type of computer or network system (hardware or software), unless indicated otherwise. Various types of general purpose or specialized computer systems may be used with or perform operations in accordance with the teachings described herein.

In view of the wide variety of embodiments to which the principles of the

present invention can be applied, it should be understood that the illustrated embodiments are exemplary only, and should not be taken as limiting the scope of the present invention. For example, the steps of the flow diagrams may be taken in sequences other than those described, and more or fewer elements may be used in the block diagrams. While various elements of the preferred embodiments have been described as being implemented in software, in other embodiments hardware implementations may alternatively be used and visa-versa.

The claims should not be read as limited to the described order or elements unless stated to that effect. Therefore, all embodiments that come within the scope and spirit of the following claims and equivalents thereto are claimed as the invention.

**I CLAIM:**

1. A method for creating an electronic framework for deploying electronic information, comprising the steps of:

building a new deployment of electronic information into a file system on a network device on a computer network;

migrating pre-determined portions of electronic information from a previous deployment of electronic information, if any, into the new deployment of electronic information;

performing management activities on the new deployment of electronic information to allow the new deployment of electronic information to be accessed from the computer network; and

instantiating executable computer software from the new deployment of electronic information, wherein executable computer software is used to process and access electronic information in an electronic data warehouse including a plurality of deployments of electronic information.

2. A computer readable medium having stored therein instructions for causing a central processing unit to execute the method of Claim 1.

3. The method of Claim 1 wherein the electronic information includes raw electronic data or processed electronic data, computer software source code, executable programs created from computer software source code, or data templates for storing electronic data.

4. The method of Claim 1 wherein the electronic information includes

polynucleotide experimental data including DNA, cDNA, RNA, cRNA or mRNA data.

5. The method of Claim 3 wherein the data templates for storing experimental data include a hierarchical arrangement of data templates in combination with data from a plurality of sources, wherein the data templates may include actual data or a symbolic link to actual data.

6. The method of Claim 1 wherein the step of building a new deployment of electronic information into a file system on a network device on a computer network includes building the deployment with a hierarchical arrangement with a primary tier and a secondary tier, wherein the primary tier includes entry points into a deployment and the secondary tier includes electronic information for the deployment.

7. The method of Claim 1 wherein the step of migrating pre-determined portions of electronic information from a previous deployment of electronic information, if any, into the new deployment of electronic information includes creating new data structures and migrating electronic information from old data structures used for a previous deployment into the new data structures used for a new deployment.

8. The method of Claim 1 wherein step of performing management activities includes: processing raw electronic information into processed electronic information; creating access points to the electronic information; creating access privileges to the electronic information, logically or physically partitioning the electronic information

into one or more sub-components; or relocating portions of the electronic information to other network devices on the computer network.

9. The method of Claim 1 wherein the step of instantiating executable computer software from the new deployment of electronic information includes instantiating utility applications, applications programs or software tools used to manipulate or view electronic information in a deployment.

10. The method of Claim 1 wherein the step of instantiating executable computer software from the new deployment of electronic information includes instantiating executable computer software to create pages of electronic content for electronic information included in an individual deployment, wherein the pages of electronic content are provided in a hardware independent mark-up language.

11. The method of Claim 10 wherein the hardware independent mark-up language includes Hyper Text Markup Language or Extensible Markup Language.

12. A method for viewing electronic information from a deployment stored in an electronic framework, the method comprising the steps of:

receiving a request to view electronic information from a deployment stored in an electronic framework in an electronic data warehouse including a plurality of deployments of electronic information, wherein the deployment includes electronic information and components used to manipulate the electronic information;

determining an access view used to provide access to the requested electronic information in the deployment, wherein the access view includes a list of components

from the deployment that can be accessed via the access view; and

providing the determined access view in response to the request to view electronic information from the deployment.

13. A computer readable medium having stored therein instructions for causing a central processing unit to execute the Method of Claim 12.

14. The method of Claim 12 wherein the step of receiving a request to view electronic information from a deployment stored in an electronic framework includes receiving the request in an Access Control List Resolver.

15. The method of Claim 12 wherein the step of determining an access view includes determining whether access is permitted to a plurality of electronic resources in a deployment and a plurality of agents that act upon the plurality of electronic resources.

16. The method of Claim 15 wherein the plurality of agents include a domain, wherein a domain is a manager of an agent, and the domain is a manger of any other domains included within it.

17. The method of Claim 12 wherein the step of determining an access view used to provide access to the requested electronic information in the deployment includes determining what electronic data and what tools used to manipulate the electronic data can be accessed in a deployment.

18. The method of Claim 12 wherein the step of providing the determined access view in response to the request to view electronic information from the deployment includes providing the determined access view to a plurality of projects with a plurality of experiments.

19. The method of Claim 18 wherein the plurality of projects with a plurality of experiments include a plurality of biotechnology projects with a plurality of biotechnology experiments including DNA, cDNA, RNA, cRNA or mRNA data.

20. The method of Claim 12 wherein the step of providing the determined access view in response to the request to view electronic information from the deployment includes providing the determined access view to a plurality of resources and a plurality of tools used to manipulate the plurality of resources.

21. A method for switching between deployments stored in an electronic framework, the method comprising the steps of:

providing an electronic framework in an electronic data warehouse including a plurality of deployments of electronic information, wherein the deployment includes electronic information and components used to manipulate the electronic information;

receiving a request for electronic information in a first deployment in the electronic data warehouse;

providing access to the electronic information in the first deployment using a first access view, wherein the first access view includes a first list of components from the first deployment that can be accessed via the first access view;

receiving a request for a electronic information in a second deployment in the

electronic data warehouse;

switching from the first access view to a second access view, wherein switching from the first access view to the second access view includes switching from a first access point in a hierarchical structure for the first deployment to a second access point in a hierarchical structure for the second deployment; and

providing access to the electronic information in the second deployment using the second access view, wherein the second access view includes a second list of components from the second deployment that can be accessed via the second access view.

22. A computer readable medium having stored therein instructions for causing a central processing unit to execute the Method of Claim 21.

23. The method of Claim 21 wherein the step of switching from the first access view to a second access view includes switching from the first access view to the second access view using an access control list resolver.

24. The method of Claim 23 wherein the access control list resolver provides a fully qualified path into the data warehouse for electronic information in a deployment.

25. The method of Claim 24 wherein the access control list resolver performs partial redirects using the fully qualified path to enable access to electronic information on a file system level in coordination with one or more servers that actually store the electronic information.

26. The method of Claim 21 wherein the step of switching from the first access view to a second access view includes determining what access is permitted to a plurality of electronic resources in the second deployment and what access is permitted a plurality of agents that act upon the plurality of electronic resources.

27. The method of Claim 26 wherein the plurality of agents include a domain, wherein a domain is a manager of an agent, and the domain is a manger of any other domains included within it.

28. The method of Claim 21 wherein the plurality of deployments include electronic information for a plurality of biotechnology projects with a plurality of biotechnology experiments including DNA, cDNA, RNA, cRNA or mRNA data.

29. An electronic data warehouse system, comprising in combination:

an electronic data warehouse including an electronic framework for storing raw and processed electronic information and for storing multiple types of components that are used to manipulate and provide access to the electronic information, wherein the data warehouse includes a plurality of deployments, wherein a deployment is a distinct body of electronic information that exists in a hierarchy in a file system on a network device on a computer network in parallel with other deployments, if any;

an access control system for providing access to the data warehouse and to resolve requests for electronic information in deployments in the data warehouse; and

a data viewer for requesting access to and for viewing desired electronic

information from a deployment in the data warehouse.

30. The electronic data warehouse system of Claim 29 wherein a deployment includes a hierarchical arrangement with a primary tier and a secondary tier, wherein the primary tier includes entry points into a deployment and the secondary tier includes electronic information for the deployment.

31. The electronic data warehouse system of Claim 29 wherein the plurality of deployments include electronic information for a plurality of biotechnology projects with a plurality of biotechnology experiments including DNA, cDNA, RNA, cRNA or mRNA.

# FIG. 1

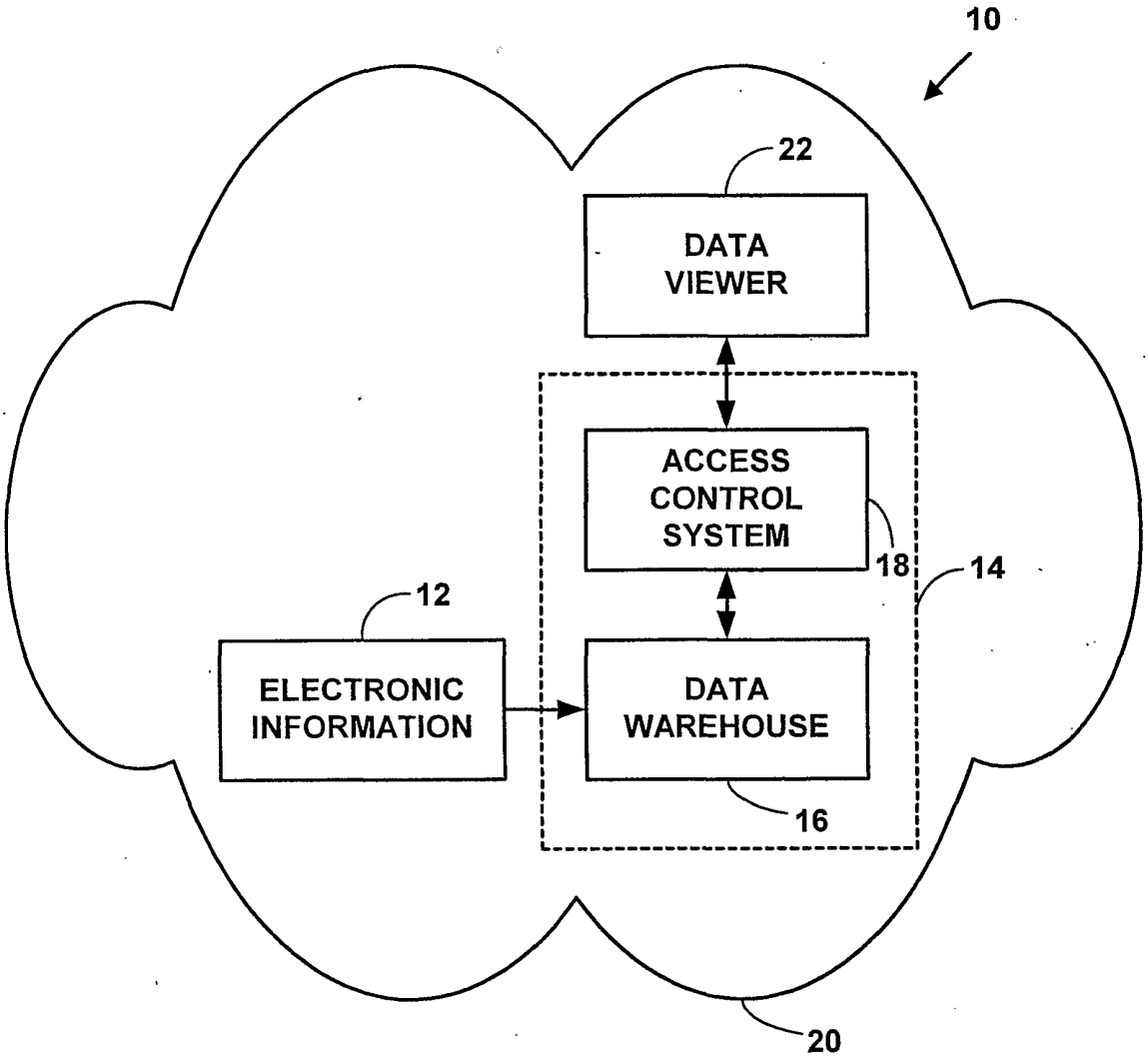
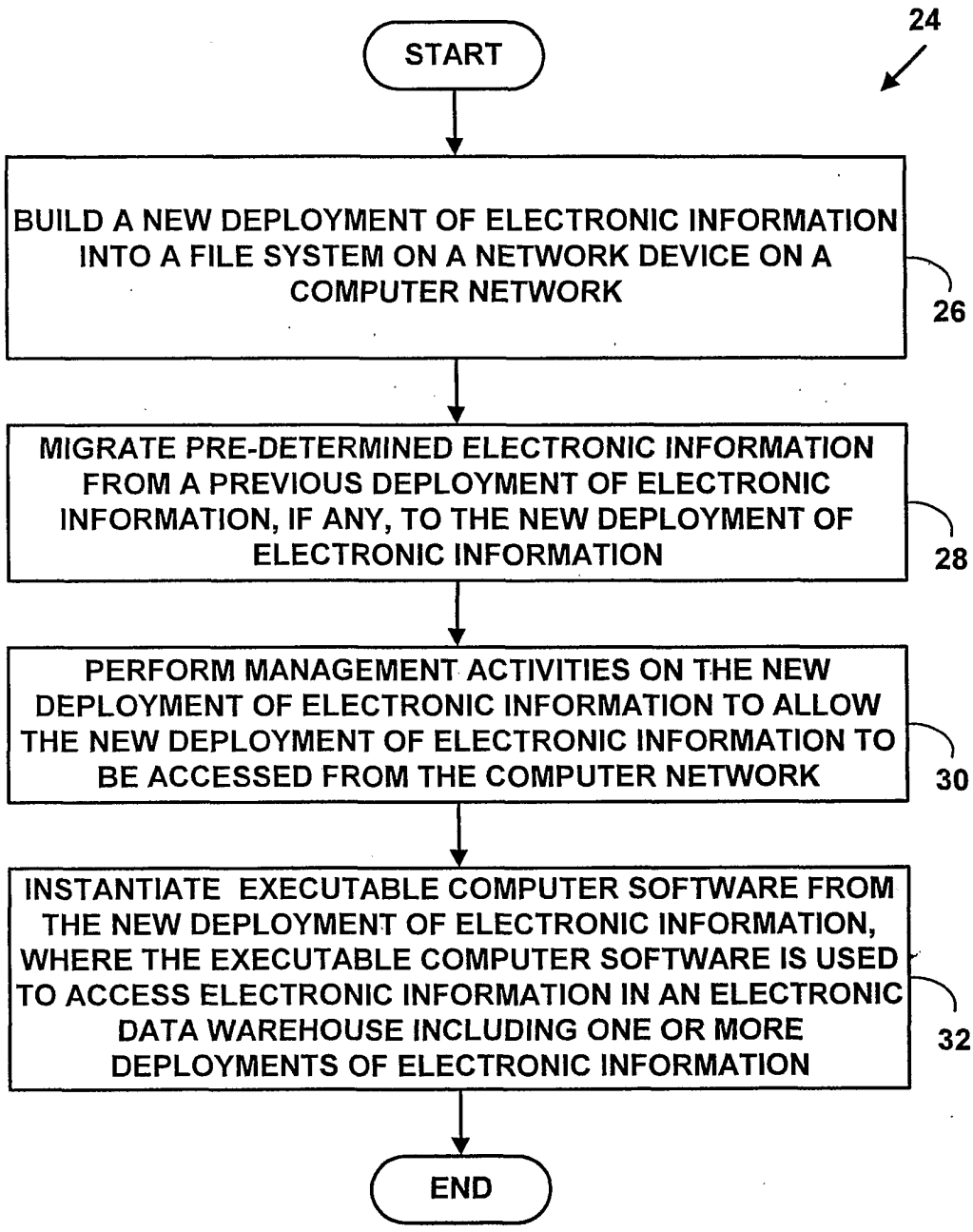
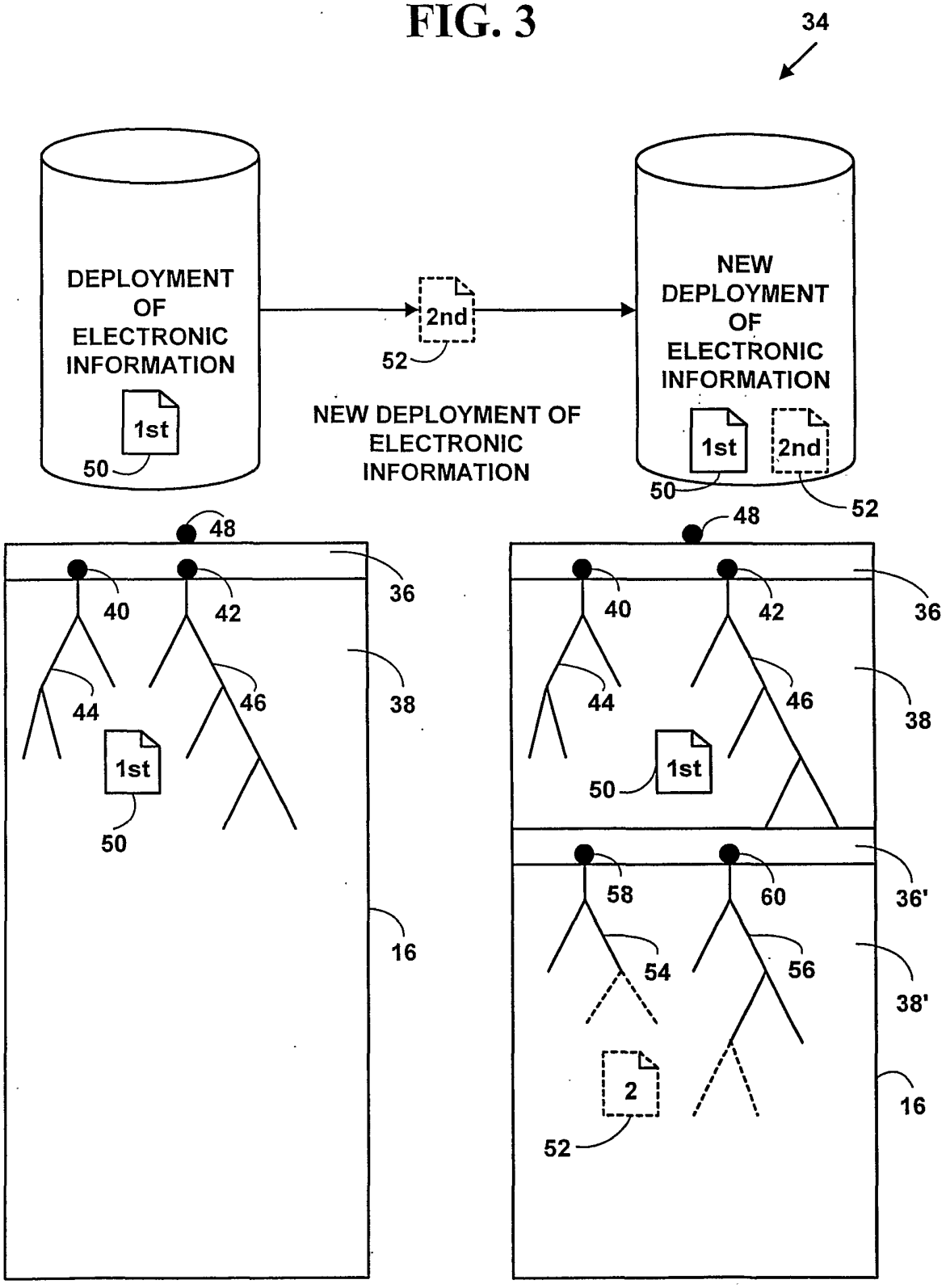


FIG. 2



# FIG. 3



# FIG. 4

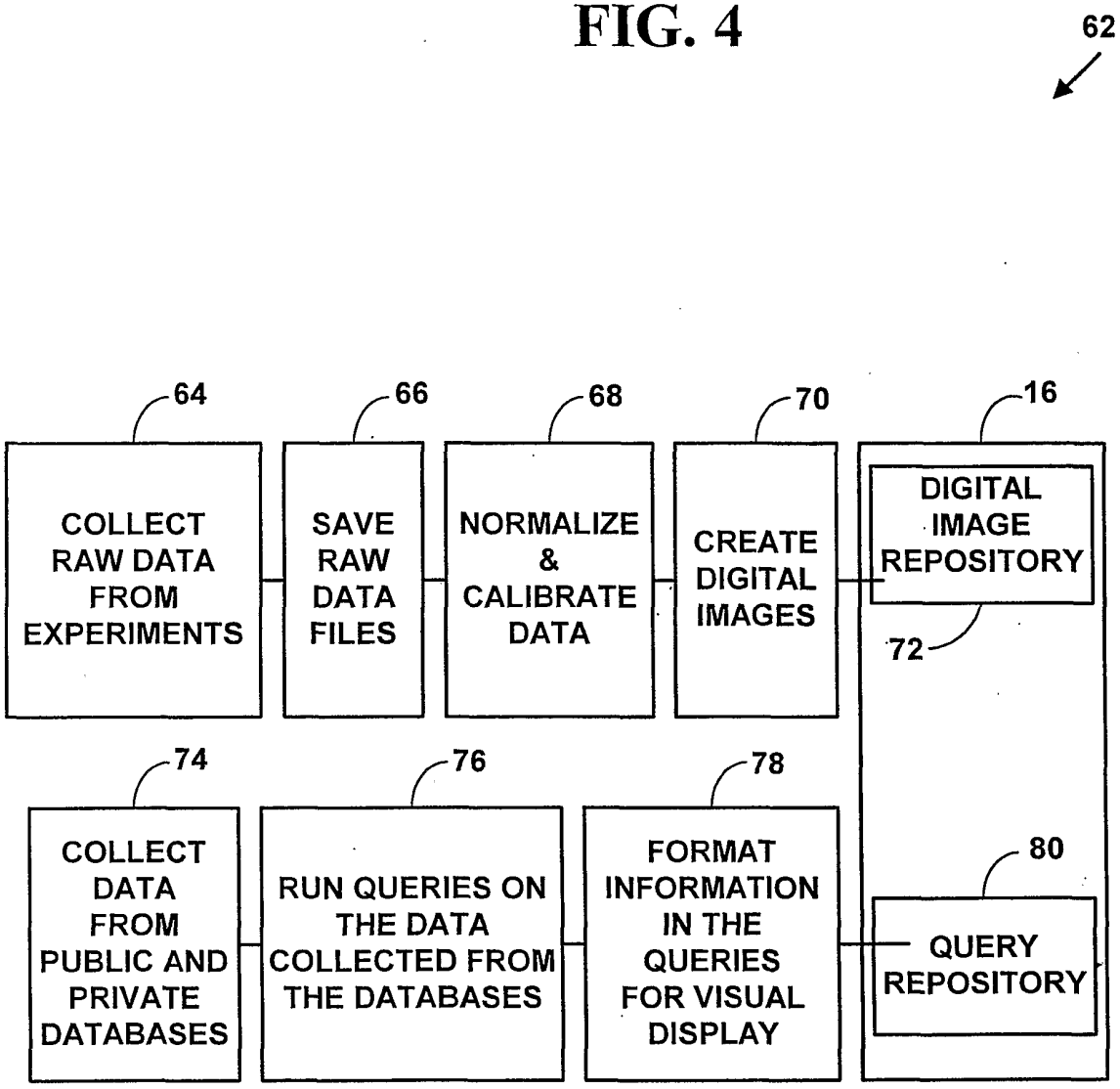
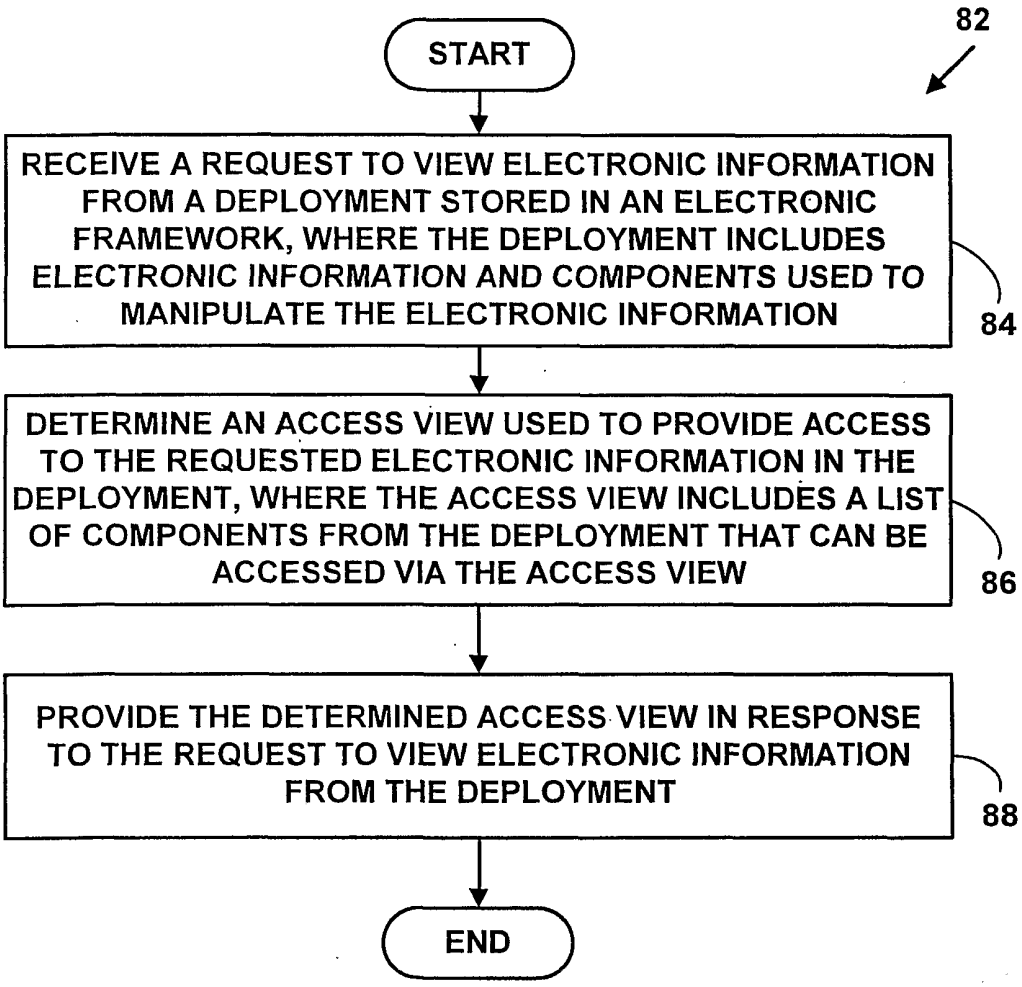
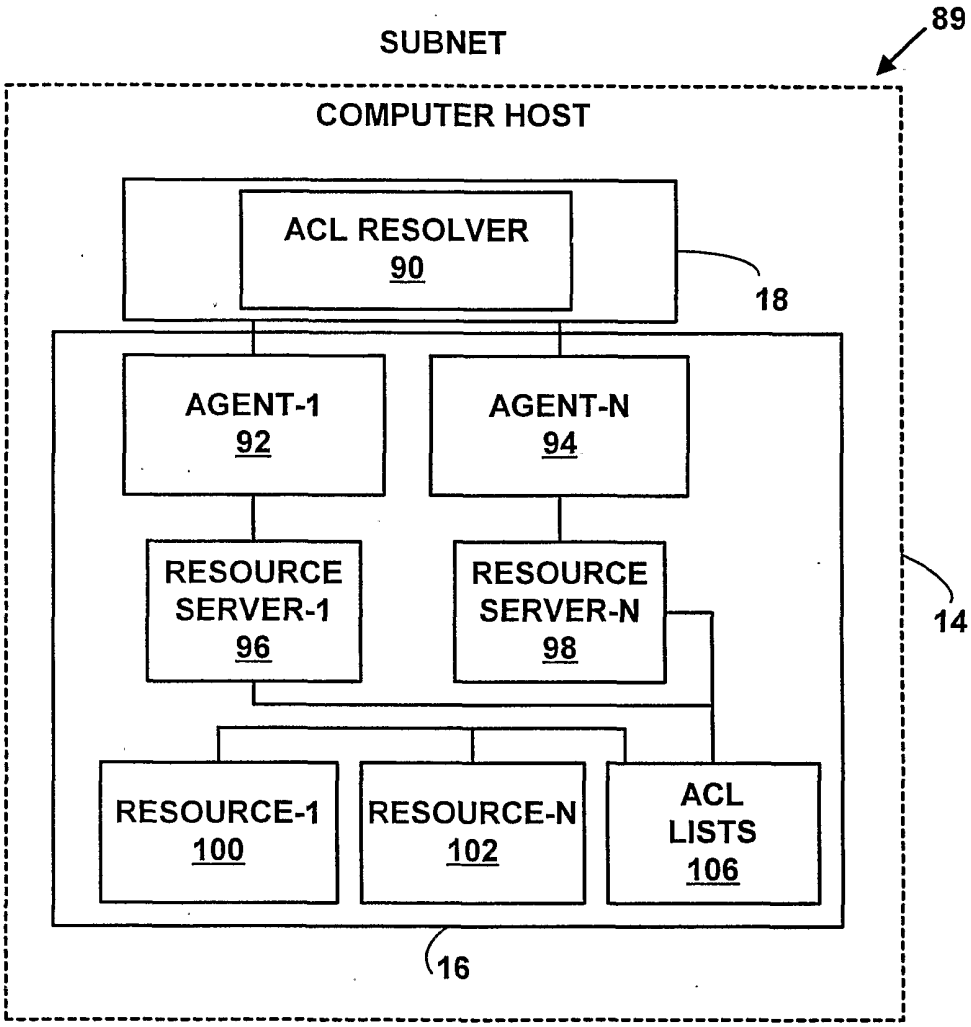


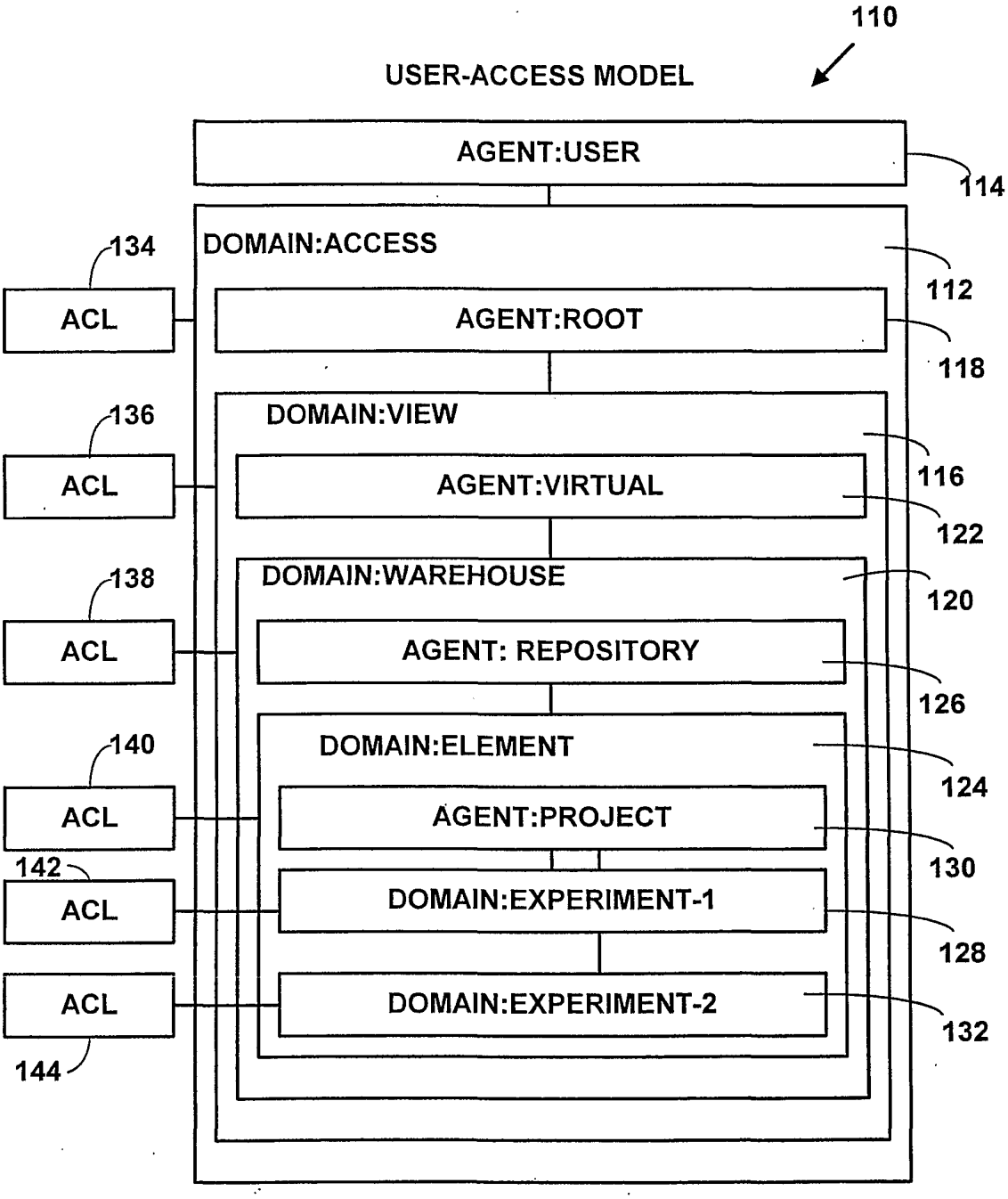
FIG. 5



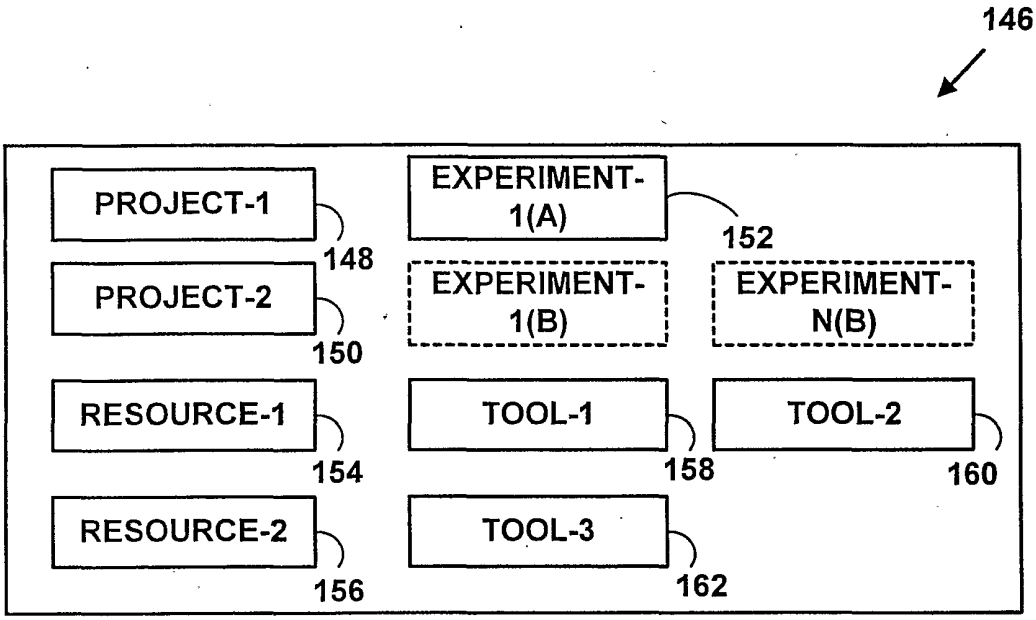
# FIG. 6



# FIG. 7



# FIG. 8




ACCESS VIEW

# FIG. 9