906 PH Ex. 1

08/324443

TOWNSEND and TOWNSEND KHOURIE and CREW
Steuart Street Tower
One Market Plaza
San Francisco, CA 94105
(415) 543-9600

Atty. Docket No. 02307-553

"Express Mail" Label No. TB380892941US

Date of Deposit October 17, 1994

PATENT APPLICATION
COMMISSIONER OF PATENT AND TRADEMARKS
Washington, D. C. 20231

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20231

By Sunil Dutt
Sunil Dutt

Sir:

Transmitted herewith for filing is the
[X] patent application of
[ ] design patent application of
[ ] continuation-in-part patent application of

100 6TAL

Inventors: Michael Doyle, David Martin and Cheong Ang

For: EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

Enclosed are:

[X] 10 sheets of [ ] formal [X] informal drawings.
[ ] An assignment of the invention to ______.
[ ] A [ ] signed [ ] unsigned Declaration & Power of Attorney.
[X] A [ ] signed [X] unsigned Declaration.
[ ] A Power of Attorney.
[ ] A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27.
[ ] A certified copy of a ______ application.
[ ] Information Disclosure Statement under 37 CFR 1.97.
[X] Appendix A; Appendix B

In view of the Unsigned Declaration as filed with this application and pursuant to 37 CFR §1.53(d), Applicant requests deferral of the filing fee until submission of the Missing Parts of Application.

DO NOT CHARGE THE FILING FEE AT THIS TIME.

Charles J. Kulas
Charles J. Kulas
Reg. No.:35,809
Attorneys for Applicant

Telephone:
(415) 543-9600
APPNOFEE.TRN 12/92

08/324443

Attorney Docket No. 2307U-553

PATENT APPLICATION

EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

Inventors:
Michael Doyle,
David Martin,
Cheong Ang

Assignee:
University of California
1320 Harbor Bay Parkway, Suite 150
Alameda, CA 94502

TOWNSEND and TOWNSEND KHOURIE and CREW
Steuart Street Tower, 20th Floor
One Market Plaza
San Francisco, California 94105
(415) 543-9600

10/

A/NO fee

08/324443

PATENT

Attorney Docket No. 2307U-553

EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

Notice Regarding Copyrighted Material

BACKGROUND OF THE INVENTION

This invention relates generally to manipulating data in a computer network, and specifically to retrieving, presenting and manipulating embedded program objects in distributed hypermedia systems.

Computer networks are becoming increasingly popular as a medium for locating and accessing a wide range of data from locations all over the world. The most popular global network is the Internet with millions of computer systems connected to it. The Internet has become popular due to widely adopted standard protocols that allow a vast interconnection of computers and localized computer networks to communicate with each other. Computer systems connected to a network such as the Internet may be of varying types, e.g., mainframes, workstations, personal computers, etc. The computers are manufactured by different companies using proprietary hardware and operating systems and thus have incompatibilities in their instruction sets, busses, software, file formats and other aspects of their architecture and operating systems. Localized computer networks connected to the Internet may be incompatible with other computer systems and localized networks in terms of the physical layer of communication including the specific hardware used to implement the network. Also, different networks use differing, incompatible protocols for transferring information and are not able to communicate with each other without a translation mechanism such as a "gateway".

The Internet provides a uniform and open standard for allowing various computers and networks to communicate with each other. For example, the Internet uses Transfer Control Protocol/Internet Protocol ("TCP/IP") that defines a uniform packet-switched communication standard which is ultimately used in every transfer of information that takes place over the Internet.

Other Internet standards are the HyperText Transmission Protocol ("HTTP") that allows hypertext documents to be exchanged freely among any computers connected to the Internet and HyperText Markup Language ("HTML") that defines the way in which hypertext documents designate links to information. See, e.g., Berners-Lee, T. J., "The world-wide web," Computer Networks and ISDN Systems 25 (1992).

A hypertext document is a document that allows a user to view a text document displayed on a display device connected to the user's computer and to access, retrieve and view other data objects that are linked to hypertext words or phrases in the hypertext document. In a hypertext document, the user may "click on," or select, certain words or phrases in the text that specify a link to other documents, or data objects. In this way, the user is able to navigate easily among data objects. The data objects may be local to the user's computer system or remotely located over a network. An early hypertext system is Hypercard, by Apple Computer, Inc. Hypercard is a standalone system where the data objects are local to the user's system.

When a user selects a phrase in a hypertext document that has an associated link to another document, the linked document is retrieved and displayed on the user's display screen. This allows the user to obtain more information in an efficient and easy manner. This provides the user with a simple, intuitive and powerful way to "branch off" from a main document to learn more about topics of interest.

Objects may be text, images, sound files, video data, documents or other types of information that is presentable to a user of a computer system. When a document is primarily text and includes links to other data objects according to the hypertext format, the document is said to be a hypertext document. When graphics, sound, video or other media capable of being manipulated and presented in a computer system is used as the object linked to, the document is said to be a hypermedia document. A hypermedia document is similar to a hypertext document, except that the user is able to click on images, sound icons, video icons, etc., that link to other objects of various media types, such as additional graphics, sound, video, text, or hypermedia or hypertext documents.

Fig. 1 shows examples of hypertext and hypermedia documents and links associating data objects in the documents to other data objects. Hypermedia document 10 includes hypertext 20, an image icon at 22, a sound icon at 24 and more hypertext 26. Fig. 1 shows hypermedia document 10 substantially as it would appear on a user's display screen. The user is able to select, or "click" on icons and text on a display screen by using an input device, such as a mouse, in a manner well-known in the art.

When the user clicks on the phrase "hypermedia," software running on the user's computer obtains the link associated with the phrase, symbolically shown by arrow 30, to access hypermedia document 14. Hypermedia document 14 is retrieved and displayed on the user's display screen. Thus, the user is presented with more information on the phrase "hypermedia." The mechanism for specifying and locating a linked object such as hypermedia document 14 is an HTML "element" that includes an object address in the format of a Uniform Resource Locator (URL).

Similarly, additional hypertext 26 can be selected by the user to access hypertext document 12 via link 32 as shown in Fig. 1. If the user selects additional hypertext 26, then the text for hypertext document 12 is displayed on the user screen. Note that hypertext document 12, itself, has hypertext at 28. Thus, the user can click on the phrase "hypermedia" while viewing document 12 to access hypermedia document 14 in a manner similar to that discussed above.

Documents, and other data objects, can be referenced by many links from many different source documents. Fig. 1 shows document 14 serving as a target link for both documents 10 and 12. A distributed hypertext or hypermedia document typically has many links within it that specify many different data objects located in computers at different geographical locations connected by a network. Hypermedia document 10 includes image icon 22 with a link to image 16. One method of viewing images is to include an icon, or other indicator, within the text.

Typically, the indicator is a very small image and may be a scaled down version of the full image. The indicator may be shown embedded within the text when the text is displayed on the display screen. The user may select the indicator to obtain the full image. When the user clicks on image icon 22 browser software executing on the user's computer system retrieves the corresponding full image, e.g., a bit map, and displays it by using external software called a "viewer." This results in the full image, represented by image 16, being displayed on the screen.

An example of a browser program is the National Center for Supercomputing Application's (NCSA) Mosaic software developed by the University of Illinois at Urbana/Champaign, Illinois. Another example is "Cello" available on the Internet at http://www.law.cornell.edu/. Many viewers exist that handle various file formats such as ".TIF," ".GIF," formats. When a browser program invokes a viewer program, the viewer is launched as a separate process. The view displays the full image in a separate "window" (in a windowing environment) or on a separate screen. This means that the browser program is no longer active while the viewer is active. By using indicators to act as place holders for full images that are retrieved and displayed only when a user selects the indicator, data traffic over the network is reduced. Also, since the retrieval and display of large images may require several seconds or more of transfer time the user does not have to wait to have images transferred that are of no interest to the user.

Returning to Fig. 1, another type of data object is a sound object shown as sound icon 24 within the hypermedia document. When the user selects sound icon 24, the user's computer accesses sound data shown symbolically by data file 40. The accessed sound data plays through a speaker or other audio device.

As discussed above, hypermedia documents allow a user to access different data objects. The objects may be text, images, sound files, video, additional documents, etc. As used in this specification, a data object is information capable of being retrieved and presented to a user of a computer system. Some data objects include executable code combined with data. An example of such a combination is a "self-extracting" data object that includes code to "unpack" or decompress data that has been compressed to make it smaller before transferring. When a browser retrieves an object such as a self-extracting data object the browser may allow the user to "launch" the self-extracting data object to automatically execute the unpacking instructions to expand the data object to its original size. Such a combination of executable code and data is limited in that the user can do no more than invoke the code to perform a singular function such as performing the self-extraction after which time the object is a standard data object.

Other existing approaches to embedding interactive program objects in documents include the Object Linking and Embedding (OLE) facility in Microsoft Windows, by Microsoft Corp., and OpenDoc, by Apple Computer, Inc. At least one shortcoming of these approaches is that neither is capable of allowing a user to access embedded interactive program objects in distributed hypermedia documents over networks.

Fig. 2 is an example of a computer network. In Fig. 2, computer systems are connected to Internet 100, although in practice Internet 100 may be replaced by any suitable computer network. In Fig. 2, a user 102 operates a small computer 104, such as a personal computer or a work station. The user's computer is equipped with various components, such as user input devices (mouse, trackball, keyboard, etc.), a display device (monitor, liquid crystal display (LCD), etc.), local storage (hard disk drive, etc.), and other components. Typically, small computer 104 is connected to a larger computer, such as server A at 106. The larger computer may have additional users and computer systems connected to it, such as computer 108 operated by user 110. Any group of computers may form a localized network. A localized network does not necessarily adopt the uniform protocols of the larger interconnecting network (i.e., Internet 100) and is more geographically constrained than the larger network. The localized network may connect to the larger network through a "gateway" or "node" implemented on, for example, a server.

Internet 100 connects other localized networks, such as server B at 120, which interconnects users 122, 124 and 126 and their respective computer systems to Internet 100. Internet 100 is made up of many interconnected computer systems and communication links. Communication links may be by hardwire, fiber optic cable, satellite or other radio wave propagation, etc. Data may move from server A to server B through any number of intermediate servers and communication links or other computers and data processing equipment not shown in Fig. 2 but symbolically represented by Internet 100.

A user at a workstation or personal computer need not connect to the Internet via a larger computer, such as server A or server B. This is shown, for example, by small computer 130 connected directly to Internet 100 as by a telephone modem or other link. Also, a server need not have users connected to it locally, as is shown by server C at 132 of Fig. 2. Many configurations of large and small computers are possible.

Typically, a computer on the Internet is characterized as either a "client" or "server" depending on the role that the computer is playing with respect to requesting information or providing information. Client computers are computers that typically request information from a server computer which provides the information. For this reason, servers are usually larger and faster machines that have access to many data files, programs, etc., in a large storage associated with the server. However, the role of a server may also be adopted by a smaller machine depending on the transaction. That is, user 110 may request information via their computer 108 from server A. At a later time, server A may make a request for information from computer 108. In the first case, where computer 108 issues a request for information from server A, computer 108 is a "client" making a request of information from server A. Server A may have the information in a storage device that is local to Server A or server A may have to make requests of other computer systems to obtain the information. User 110 may also request information via their computer 108 from a server, such as server B located at a remote geographical location on the Internet. However, user 110 may also request information from a computer, such as small computer 124, thus placing small computer 124 in the role of a "server." For purposes of this specification, client and server computers are categorized in terms of their predominant role as either an information requestor or provider. Clients are generally information requestors, while servers are generally information providers.

Referring again to Fig. 1, data objects such as distributed hypermedia documents 10, 12 and 14, image 16 and sound data file 40, may be located at any of the computers shown in Fig. 2. Since these data objects may be linked to a document located on another computer the Internet allows for remote object linking.

For example, hypertext document 10 of Fig. 1 may be located at user 110's client computer 108. When user 110 makes a request by, for example, clicking on hypertext 20 (i.e., the phrase "hypermedia"), user 110's small client computer 108 processes links within hypertext document 10 to retrieve document 14. In this example, we assume that document 14 is stored at a remote location on server B. Thus, in this example, computer 108 issues a command that includes the address of document 14. This command is routed through server A and Internet 100 and eventually is received by server

B. Server B processes the command and locates document 14 on its local storage. Server 14 then transfers a copy of the document back to client 108 via Internet 100 and server A. After client computer 108 receives document 14, it is displayed so that user 110 may view it.

Similarly, image object 16 and sound data file 40 may reside at any of the computers shown in Fig. 2. Assuming image object 16 resides on server C when user 110 clicks on image icon 22, client computer 108 generates a command to retrieve image object 16 to server C. Server C receives the command and transfers a copy of image object 16 to client computer 108. Alternatively, an object, such as sound data file 40, may reside on server A so that it is not necessary to traverse long distances via the Internet in order to retrieve the data object.

The Internet is said to provide an "open distributed hypermedia system." It is an "open" system since Internet 100 implements a standard protocol that each of the connecting computer systems, 106, 130, 120, 132 and 134 must implement (TCP/IP). It is a "hypermedia" system because it is able to handle hypermedia documents as described above via standards such as the HTTP and HTML hypertext transmission and mark up standards, respectively. Further, it is a "distributed" system because data objects that are imbedded within a document may be located on many of the computer systems connected to the Internet. An example of an open distributed hypermedia system is the so-called "world-wide web" implemented on the Internet and discussed in papers such as the Berners-Lee reference given above.

The open distributed hypermedia system provided by the Internet allows users to easily access and retrieve different data objects located in remote geographic locations on the Internet. However, this open distributed hypermedia system as it currently exists has shortcomings in that today's large data objects are limited largely by bandwidth constraints in the various communication links in the Internet and localized networks, and by the limited processing power, or computing constraints, of small computer systems normally provided to most users. Large data objects are difficult to update at frame rates fast enough (e.g., 30 frames per second) to achieve smooth animation. Moreover, the processing power needed to perform the calculations to animate such images in real time does not exist on most workstations, not to mention personal computers. Today's browsers and viewers are not capable of performing the computation necessary to generate and render new views of these large data objects in real time.

For example, the Internet's open distributed hypermedia system allows users to view still images. These images are simple non-interactive two-dimensional images, similar to photographs. Much digital data available today exists in the form of high-resolution multi-dimensional image data (e.g., three dimensional images) which is viewed on a computer while allowing the user to perform real time viewing transformations on the data in order for the user to better understand the data.

An example of such type of data is in medical imaging where advanced scanning devices, such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT), are widely used in the fields of medicine, quality assurance and meteorology to present physicians, technicians and meteorologists with large amounts of data in an efficient way. Because visualization of the data is the best way for a user to grasp the data's meaning, a variety of visualization techniques and real time computer graphics methods have been developed. However, these systems are bandwidth-intensive and compute-intensive and often require multiprocessor arrays and other specialized graphics hardware to carry them out in real time. Also, large amounts of secondary storage for data are required. The expense of these requirements has limited the ability of researchers to readily exchange findings since these larger computers required to store, present and manipulate images are not available to many of the researchers that need to have access to the data.

On the other hand, small client computers in the form of personal computers or workstations such as client computer 108 of Fig. 2 are generally available to a much larger number of researchers. Further, it is common for these smaller computers to be connected to the Internet. Thus, it is desirable to have a system that allows the accessing, display and manipulation of large amounts of data, especially image data, over the Internet to a small, and relatively cheap, client computer.

Due to the relatively low bandwidth of the Internet (as compared to today's large data objects) and the relatively small amount of processing power available at client computers, many valuable tasks performed by computers cannot be performed by users at client computers on the Internet. Also, while the present open distributed hypermedia system on the Internet allows users to locate and retrieve data objects it allows users very little, if any, interaction with these data objects. Users are limited to traditional hypertext and hypermedia forms of selecting linked data objects for retrieval and launching viewers or other forms of external software to have the data objects presented in a comprehensible way.

Thus, it is desirable to have a system that allows a user at a small client computer connected to the Internet to locate, retrieve and manipulate data objects when the data objects are bandwidth-intensive and compute-intensive. Further, it is desirable to allow a user to manipulate data objects in an interactive way to provide the user with a better understanding of information presented and to allow the user to accomplish a wider variety of tasks.

SUMMARY OF THE INVENTION

The present invention provides a method for running embedded program objects in a computer network environment. The method includes the steps of providing at least one client workstation and one network server coupled to the network environment where the network environment is a distributed hypermedia environment; displaying, on the client workstation, a portion of a hypermedia document received over the network from the server, where the hypermedia document includes an embedded controllable application; and interactively controlling the embedded controllable application from the client workstation via communication sent over the distributed hypermedia environment.

The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way. The invention not only allows the user to use a hypermedia format to locate and retrieve program objects, but also allows the user to interact with an application program located at a remote computer. Interprocess communication between the hypermedia browser and the embedded application program is ongoing after the program object has been launched. The user is able to use a vast amount of computing power beyond that which is contained in the user's client computer.

In one application, high resolution three dimensional images are processed in a distributed manner by several computers located remotely from the user's client computer. This amounts to providing parallel distributed processing for tasks such as volume rendering or three dimensional image transformation and display. Also, the user is able to rotate, scale and otherwise reposition the viewpoint with respect to these images without exiting the hypermedia browser software. The control and interaction of viewing the image may be provided within the same window that the browser is using assuming the environment is a "windowing" environment. The viewing transformation and volume rendering calculations may be performed by remote distributed computer systems.

Once an image representing a new viewpoint is computed the frame image is transmitted over the network to the user's client computer where it is displayed at a designated position within a hypermedia document. By transmitting only enough information to update the image, the need for a high bandwidth data connection is reduced. Compression can be used to further reduce the bandwidth requirements for data transmission.

Other applications of the invention are possible. For example, the user can operate a spreadsheet program that is being executed by one or more other computer systems connected via the network to the user's client computer. Once the spreadsheet program has calculated results, the results may be sent over the network to the user's client computer for display to the user. In this way, computer systems located remotely on the network can be used to provide the computing power that may be required for certain tasks and to reduce the data bandwidth by only transmitting results of the computations.

Still other applications of the present invention are possible, as disclosed in the specification, below.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates examples of hypertext and hypermedia documents and links;

Fig. 2 is an example of a computer network;

Fig. 3 is an illustration of a computer system suitable for use with the present invention;

Fig. 4 is an illustration of basic subsystems in the computer system of Fig. 3;

Fig. 5 is an illustration of an embodiment of the invention using a client computer, server computer and a network;

Fig. 6 shows another embodiment of the present invention using additional computers on the network;

Fig. 7A is a flowchart of some of the functionality within the HTMLparse.c file;

Fig. 7B is a flowchart of some of the functionality within the HTMLformat.c file;

Fig. 8A is a flowchart of some of the functionality within the HTMLwidget.c file;

Fig. 8B is a flowchart of some of the functionality within the HTML.c file;

Fig. 9 is a screen display generated in accordance with the present invention; and

Fig. 10 is a diagram of the various processes and data paths in the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

375 pages of Source code on 4 microfiche Appendices A and B are provided to this specification. The source code should be consulted to provide details of a specific embodiment of the invention in conjunction with the discussion of the routines in this specification. The source code in Appendix A includes NCSA Mosaic version 2.4 source code along with modifications to the source code to implement the present invention. Appendix B includes source code implementing an application program interface. The source code is written in the "C" computer language to run on an X-Window platform.

SS
10/7/98

Fig. 3 is an illustration of a computer system suitable for use with the present invention. Fig. 3 depicts but one example of many possible computer types or configurations capable of being used with the present invention. Fig. 3 shows computer system 150 including display device 153, display screen 155, cabinet 157, keyboard 159 and mouse 161. Mouse 161 and keyboard 159 are "user input devices." Other examples of user input devices are a touch screen, light pen, track ball, data glove, etc.

Mouse 161 may have one or more buttons such as buttons 163 shown in Fig. 3. Cabinet 157 houses familiar computer components such as disk drives, a processor, storage means, etc. As used in this specification "storage means" includes any storage device used in connection with a computer system such as disk drives, magnetic tape, solid state memory, bubble memory, etc. Cabinet 157 may include additional hardware such as input/output (I/O) interface cards for connecting computer system 150 to external devices such as an optical character reader, external storage devices, other computers or additional devices.

Fig. 4 is an illustration of basic subsystems in computer system 150 of Fig. 3. In Fig. 4, subsystems are represented by blocks such as central processor 180, system memory 181 consisting of random access memory (RAM) and/or read-only memory (ROM), display adapter 182, monitor 183 (equivalent to display device 153 of Fig. 3), etc. The subsystems are interconnected via a system bus 184.

Additional subsystems such as a printer, keyboard, fixed disk and others are shown. Peripherals and input/output (I/O) devices can be connected to the computer system by, for example serial port 185. For example, serial port 185 can be used to connect the computer system to a modem for connection to a network or serial port 185 can be used to interface with a mouse input device. The interconnection via system bus 184 allows central processor 180 to communicate with each subsystem and to control the execution of instructions from system memory 181 or fixed disk 186, and the exchange of information between subsystems. Other arrangements of subsystems and interconnections are possible.

Fig. 5 is an illustration of an embodiment of the invention using a client computer, server computer and a network.

In Fig. 5, client computer 200 communicates with server computer 204 via network 206. Both client computer 200 and server computer 204 use a network protocol layer to communicate with network 206. In a preferred embodiment, network 206 is the Internet and the network protocol layers are TCP/IP. Other networks and network protocols may be used. For ease of illustration, additional hardware and software layers are not shown in Fig. 5.

Client computer 200 includes processes, such as browser client 208 and application client 210. In a preferred embodiment, application client 210 is resident within client computer 200 prior to browser client 208's parsing of a hypermedia document as discussed below. In a preferred embodiment application client 210 resides on the hard disk or RAM of client computer 200 and is loaded (if necessary) and executed when browser client 208 detects a link to application client 210. The preferred embodiment uses the XEvent interprocess communication protocol to exchange information between browser client 208 and application client 210 as described in more detail, below. Another possibility is to install application client 210 as a "terminate and stay resident" (TSR) program in an operating system environment, such as X-Window. Thereby making access to application client 210 much faster.

Browser client 208 is a process that a user of client computer 200 invokes in order to access various data objects, such as hypermedia documents, on network 206. Hypermedia document 212 shown within client computer 200 is an example of a hypermedia document, or object, that a user has requested access to. In this example, hypermedia document 212 has been retrieved from a server connected to network 206 and has been loaded into, e.g., client computer 200's RAM or other storage device.

Once hypermedia document 212 has been loaded into client computer 200, browser client 208 parses hypermedia document 212. In parsing hypermedia document 212, browser client 208 detects links to data objects as discussed above in the Background of the Invention section. In Fig. 5, hypermedia document 212 includes an embedded program link at 214. Embedded program link 214 identifies application client 212 as an application to invoke. In this present example, the application, namely, application client 210, resides on the same computer as the browser client 208 that the user is executing to view the hypermedia document. Embedded program link 214 may include additional information, such as parameters, that tell application client 210 how to proceed. For example, embedded program link 214 may include a specification as to a data object that application client 210 is to retrieve and process.

When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information) and application client 210 executes instructions to perform processing in accordance with the present invention.

An example of the type of processing that application client 210 may perform is multidimensional image visualization. Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200. This means that application client 210 can make requests over network 206 for data objects, such as multidimensional image objects. For example, application client 210 may request an object, such as object 1 at 216, located in server computer 204. Application client 210 may make the request by any suitable means. Assuming network 206 is the Internet, such a request would typically be made by using HTTP in response to a HTML-style link definition for embedded program link 214.

Assuming application client 210 has made a request for the data object at 216, server process 218 ultimately receives the request. Server process 218 then retrieves data object 216 and transfers it over network 206 back to application client 210. To continue with the example of a multidimensional visualization application, data object 216 may be a three dimensional view of medical data for, e.g., an embryo.

After application client 210 receives the multidimensional data object 216, application client 210 executes instructions to display the multidimensional embryo data on the display screen to a user of the client computer 200. The user is then able to interactively operate controls to recompute different views for the image data. In a preferred embodiment, a control window is displayed within, or adjacent to, a window generated by browser client 208 that contains a display of hypermedia document 212. An example of such display is discussed below in connection with Fig. 9. Thus, the user is able to interactively manipulate a multidimensional image object by means of the present invention. In order to make application client 210 integral with displays created by browser client 208, both the browser client and the application client must be in communication with each other, as shown by the arrow connecting the two within client computer 200. The manner of communication is through an application program interface (API), discussed below.

Browser client 208 is a process, such as NCSA Mosaic, Cello, etc. Application client 210 is embodied in software presently under development called "VIS" and "Panel" created by the Center for Knowledge Management at the

University of California, San Francisco, as part of the Doyle Group's distributed hypermedia object embedding approach described in "Integrated Control of Distributed Volume Visualization Through the World-Wide-Web," by C. Ang, D. Martin, M. Doyle; to be published in the Proceedings of Visualization 1994, IEEE Press, Washington, D.C., October 1994.

Versions and descriptions of software embodying the present invention are generally available as hyperlinked data objects from the Visible Embryo Project's World Wide Web document at the URL address "HTTP://visembryo.ucsf.edu/".

Another embodiment of the present invention uses an application server process executing on server computer 204 to assist in processing that may need to be performed by an external program. For example, in Fig. 5, application server 220 resides on server computer 204. Application server 220 works in communication with application client 210 residing on client computer 200. In a preferred embodiment, application server 220 is called VRServer, also a part of Doyle Group's approach. Since server computer 204 is typically a larger computer having more data processing capabilities and larger storage capacity, application server 220 can operate more efficiently, and much faster, than application client 210 in executing complicated and numerous instructions.

In the present example where a multidimensional image object representing medical data for an embryo is being viewed, application server 220 could perform much of the viewing transformation and volume rendering calculations to allow a user to interactively view the embryo data at their client computer display screen. In a preferred embodiment, application client 210 receives signals from a user input device at the user's client computer 200. An example of such input would be to rotate the embryo image from a current position to a new position from the user's point of view. This information is received by application client 210 and processed to generate a command sent over network 206 to application server 220. Once application server 220 receives the information in the form of, e.g., a coordinate transformation for a new viewing position, application server 220 performs the mathematical calculations to compute a new view for the embryo image. Once the new view has been computed, the image data for the new view is sent over network 206 to application client 210 so that application client 210 can update the viewing window currently displaying the embryo image. In a preferred embodiment, application server 220 computes a frame buffer of raster display data, e.g., pixel values, and transfers this frame buffer to application client 210. Techniques, such as data compression and delta encoding, can be used to compress the data before transmitting over network 206 to reduce the bandwidth requirement.

It will be readily seen that application server 220 can advantageously use server computer 204's computing resources to perform the viewing transformation much more quickly than could application client 210 executing on client computer 200. Further, by only transmitting the updated frame buffer containing a new view for the embryo image, the amount of data sent over network 206 is reduced. By using appropriate compression techniques, such as, e.g., MPEG (Motion Picture Experts Group) or JPEG (Joint Photographic Experts Group), efficient use of network 206 is preserved.

Fig. 6 shows yet another embodiment of the present invention. Fig. 6 is similar to Fig. 5, except that additional computers 222 and 224 are illustrated. Each additional computer includes a process labeled "Application (Distributed)." The distributed application performs a portion of the task that an application, such as application server 220 or application client 210, perform. In the present example, tasks such as volume rendering may be broken up and easily performed among two or more computers. These computers can be remote from each other on network 206. Thus, several computers, such as server computer 204 and additional computers 222 and 224 can all work together to perform the task of computing a new viewpoint and frame buffer for the embryo for the new orientation of the embryo image in the present example. The coordination of the distributed processing can be performed at client computer 200 by application client 210, at server computer 204 by application server 220, or by any of the distributed applications executing on additional computers, such as 222 and 224. In a preferred embodiment, distributed processing is coordinated by a program called "VIS" represented by application client 210 in Fig. 6.

Other applications of the invention are possible. For example, the user can operate a spreadsheet program that is being executed by one or more other computer systems connected via the network to the user's client computer. Once the spreadsheet program has calculated results, those results may be sent over the network to the user's client computer for display within the hypermedia document on the user's client computer. In this way, computer systems located remotely on the network can be used to provide the computing power that may be required for certain tasks and to reduce the data bandwidth required by only transmitting results of the computations.

Another type of possible application of this invention would involve embedding a program which runs only on the client machine, but which provides the user with more functionality than exists in the hypermedia browser alone. An example of this is an embedded client application which is capable of viewing and interacting with images which have been processed with Dr. Doyle's MetaMAP invention (US Patent 4,847,604). This MetaMAP process uses object-oriented color map processing to allow individual color index ranges within paletted images to have object identities, and is useful for the creation of, for example, interactive picture atlases. It is a more efficient means for defining irregular "hotspots" on images than the ISMAP function of the World Wide Web, which uses polygonal outlines to define objects in images. A MetaMAP-capable client-based image browser application can be embedded, together with an associated image, within a hypermedia document, allowing objects within the MetaMAP-processed image to have URL addresses associated with them. When a user clicks with a mouse upon an object within the MetaMAP-processed image, the MetaMAP client application relays the relevant URL back to the hypermedia browser application, which then retrieves the HTML file or hypermedia object which corresponds to that URL.

The various processes in the system of the present invention communicate through a custom API called Mosaic/External Application Program Interface MEAPI. The MEAPI set of predefined messages includes those shown in Table I.

Message Function	Message Name
Messages from server to client:	
1. Server Update Done	XtNrefreshNotify
2. Server Ready	XtNpanelStartNotify
3. Server Exiting	XtNpanelExitNotify
Messages from client to server:	
4. Area Shown	XtNmapNotify
5. Area Hidden	XtNunmapNotify
6. Area Destroyed	XtNexitNotify

Table I

The messages in Table I are defined in the file protocol_lib.h in Appendix B. The functions of the MEAPI are provided in protocol_lib.c of Appendix B. Thus, by using MEAPI a server process communicates to a client application program to let the client application know when the server has finished updating information, such as an image frame buffer, or pixmap (Message 1); when the server is ready to start processing messages (Message 2) and when the server is exiting or stopping computation related to the server application program.

For client to server communications, MEAPI provides for the client informing the server when the image display window area is visible, when the area is hidden and when the area is destroyed. Such information allows the server to decide whether to allocate computing resources for, e.g., rendering and viewing transformation tasks where the server is running an application program to generate new views of a multi dimensional object. Source code for MEAPI fundamental functions such as handle_client_msg, register_client, register_client_msg_callback and send_client_msg may be found in protocol_lib.c as part of the source code in Appendix B.

Next, a discussion of the software processes that perform parsing of a hypermedia document and launching of an application program is provided in connection with Table II and Figs. 7A, 7B, 8A and 8B.

Table II, below, shows an example of an HTML tag format used by the present invention to embed a link to an application program within a hypermedia document.

```
<EMBED
     TYPE = "type"
     HREF = "href"
     WIDTH = width
     HEIGHT = height
>
```

TABLE II

As shown in Table II, the EMBED tag includes TYPE, HREF, WIDTH and HEIGHT elements. The TYPE element is a Multipurpose Internet Mail Extensions (MIME) type. Examples of values for the TYPE element are "application/x-vis" or "video/mpeg". The type "application /x-vis" indicates that an application named "x-vis" is to be used to handle the object at the URL specified by the HREF. Other types are possible such as "application/x-inventor", "application/postscript" etc. In the case where TYPE is "application/x-vis" this means that the object at the URL address is a three dimensional image object since the program "x-vis" is a data visualization tool designed to operate on three dimensional image objects. However, any manner of application program may be specified by the TYPE element so that other types of applications, such as a spreadsheet program, database program, word processor, etc. may be used with the present invention. Accordingly, the object reference by the HREF element would be, respectively, a spreadsheet object, database object, word processor document object, etc.

On the other hand, TYPE values such as "video/mpeg", "image/gif", "video/x-sgi-movie", etc. describe the type of data that HREF specifies. This is useful where an external application program, such as a video player, needs to know what format the data is in, or where the browser client needs to determine which application to launch based on the data format. Thus, the TYPE value can specify either an application program or a data type. Other TYPE values are possible. HREF specifies a URL address as discussed above for a data object. Where TYPE is "application/x-vis" the URL address specifies a multi-dimensional image object. Where TYPE is "video/mpeg" the URL address specifies a video object.

WIDTH and HEIGHT elements specify the width and height dimensions, respectively, of a Distributed Hypermedia Object Embedding (DHOE) window to display an external application object such as the three dimensional image object or video object discussed above.

Fig. 7A is a flowchart describing some of the functionality within the HTMLparse.c file of routines. The routines in HTMLparse.c perform the task of parsing a hypermedia document and detecting the EMBED tag. In a preferred embodiment, the enhancements to include the EMBED tag are made to an HTML library included in public domain NCSA Mosaic version 2.4. ~~These files are included as source code in Appendix A attached to this specification~~. Note that much of the source code ~~in Appendix A~~ is pre-existing NCSA Mosaic code. Only those portions of the source code that relate to the new functionality discussed in this specification should be considered as part of the invention. The new functionality is identifiable as being set off from the main body of source code by conditional compilation macros such as "#ifdef ... #endif" as will be readily apparent to one of skill in the art.

In general, the flowcharts in this specification illustrate one or more software routines executing in a computer system such as computer system 1 of Fig. 1. The routines may be implemented by any means as is known in the art. For example, any number of computer programming languages, such as "C", Pascal, FORTRAN, assembly language, etc., may be used. Further, various programming approaches such as procedural, object oriented or artificial intelligence techniques may be employed.

The steps of the flowcharts may be implemented by one or more software routines, processes, subroutines, modules, etc. It will be apparent that each flowchart is illustrative of merely the broad logical flow of the method of the present invention and that steps may be added to, or taken away from, the flowcharts without departing from the scope of the invention. Further, the order of execution of steps in the flowcharts may be changed without departing from the scope of the invention. Additional considerations in implementing the method described by the flowchart in software may dictate changes in the selection and order of steps. Some considerations are event handling by interrupt driven, polled, or other schemes. A multiprocessing or multitasking environment could allow steps to be executed "concurrently." For ease of discussion the implementation of each flowchart may be referred to as if implemented in a single "routine".

The modifications to NCSA Mosaic version 2.4 software files HTMLparse.c, HTMLformat.c, HTMLwidget.c and HTML.c will next be discussed, in turn.

Returning to Fig. 7, it is assumed that a hypermedia document has been obtained at a user's client computer and that a browser program executing on the client computer displays the document and calls a first routine in the HTMLparse.c file called "HTMLparse". This first routine, HTMLparse, is entered at step 252 where a pointer to the start of the document portion is passed. Steps 254, 256 and 258 represent a loop where the document is parsed or scanned for HTML tags or other symbols. While the file HTMLparse.c includes routines to handle all possible tags and symbols that may be encountered, Fig. 7A, for simplicity, only illustrates the handling of EMBED tags.

Assuming there is more text to parse, execution proceeds to step 256 where routines in HTMLparse.c obtain the next item (e.g., word, tag or symbol) from the document. At step 258 a check is made as to whether the current tag is the EMBED tag. If not, execution returns to step 254 where the next tag in the document is obtained. If, at step 258, it is determined that the tag is the EMBED tag, execution proceeds to step 260 where an enumerated type is assigned for the tag. Each occurrence of a valid EMBED tag specifies an embedded object. HTMLParse calls a routine "get_mark" in HTMLparse.c to put sections of HTML document text into a "markup" text data structure. Routine get_mark, in turn, calls ParseMarkType to assign an enumerated type. The enumerated type is an identifier with a unique integer associated with it that is used in later processing described below.

Once all of the hypermedia text in the text portion to be displayed has been parsed, execution of HTMLparse.c routines terminates at step 262.

Fig. 7B is a flowchart of routines in file HTMLformat.c to process the enumerated type created for the EMBED tag by routines in HTMLparse.c. In the X-Window implementation of a preferred embodiment, the enumerated type is processed as if it is a regular Motif/XT widget. For details on X-Window development see, e.g., "Xlib Programming Manual," "X Toolkit Intrinsics Programming Manual" and "Motif Programming Manual" published by O'Reilly & Associates, Inc. HTMLformat is entered at step 270 where a pointer to the enumerated type to process is passed.

At step 272 the parameters of the structure are initialized in preparation for inserting a DrawingArea widget on an HTML page. This includes providing values for the width and height of a window on the display to contain an image, position of the window, style, URL of the image object, etc. Various codes are also added by routines in HTMLformat.c (such as TriggerMarkChanges) to insert an internal representation of the HTML statement into an object list maintained internally by the browser. In the X-Window application corresponding to the source code of Appendix A, the browser is NCSA Mosaic version 2.4.

Fig. 8A is a flowchart for routine HTMLwidget. HTMLwidget creates display data structures and launches an external application program to handle the data object specified by the URL in the EMBED tag.

HTMLwidget is entered at step 280 after HTMLformat has created the internal object representation of the EMBED tag. HTMLwidget is passed the internal object and performs its processing on the object. At step 282 the DrawingArea widget is created according to the type of the internal representation, from HTMLformat, specified in the internal object. Similarly, at step 284 a pixmap area for backing storage is defined.

At step 286 a check is made as to whether the type attribute of the object, i.e., the value for the TYPE element of the EMBED tag, is an application. If so, step 290 is executed to launch a predetermined application. In a preferred embodiment an application is launched according to a user-defined list of application type/application pairs. The list is defined as a user-configurable XResource as described in "Xlib Programming Manual." An alternative embodiment could use the MIME database as the source of the list of application type/application pairs. The routine "vis_start_external_application" in file HTMLformat.c is invoked to match the application type and to identify the application to launch.

The external application is started as a child process of the current running process (Mosaic), and informed about the window ID of the DrawingArea created in HTMLformat. The external application is also passed information about the ID of the pixmap, the data URL and dimensions. Codes for communication such as popping-up/iconifying, start notification, quit notification and refresh notification with external applications and DrawingArea refreshing are also added. Examples of such codes are (1) "setup/start" in vis_register_client and vis_get_panel_window in HTMLwidgets.c; (2) "handle messages from external applications" in vis_handle_panel_msg in HTMLwidgets.c; (3) "send messages to external applications" in vis_send_msg in HTMLwidgets.c; (4) "terminate external applications" in vis_exit in HTMLwidgets.c which calls vis_send_msg to send a quit message; and (5) "respond to refresh msgs" in vis_redraw in HTMLwidgets.c.

If, at step 286, the type is determined not to be an application object (e.g., a three dimensional image object in the case of application "x-vis") a check is made at step 288

216

to determine if the type is a video object. If so, step 292 is executed to launch a video player application. Parameters are passed to the video player application to allow the player to display the video object within the DrawingArea within the display of the portion of hypermedia document on the client's computer. Note that many other application objects types are possible as described above.

Fig. 8B is a flowchart for routine HTML. Routine HTML takes care of "shutting down" the objects, data areas, etc. that were set up to launch the external application and display the data object. HTML is entered at step 300 and is called when the display or other processing of the EMBED tag has been completed. At step 302 the display window is removed and the memory areas for the pixmap and internal object structure is made free for other uses. Completion of processing can be by user command or by computer control.

The present invention allows a user to have interactive control over application objects such as three dimensional image objects and video objects. In a preferred embodiment, controls are provided on the external applications' user interface. In the case of a VIS/panel application, a process, "panel" creates a graphical user interface (GUI) thru which the user interacts with the data. The application program, VIS, can be executing locally with the user's computer or remotely on a server, or on one or more different computers, on the network. The application program updates pixmap data and transfers the pixmap data (frame image data) to a buffer to which the browser has access. The browser only needs to respond to the refresh request to copy the contents from the updated pixmap to the DrawingArea. The Panel process sends messages as "Msg" sending performed by routines such as vis_send_msg and vis_handle_panel_msg to send events (mousemove, keypress, etc.) to the external application.

Fig. 9 is a screen display of the invention showing an interactive application object (in this case a three dimensional image object) in a window within a browser window. In Fig. 9, the browser is NCSA Mosaic version 2.4. The

31

processes VIS, Panel and VRServer work as discussed above. Fig. 9 shows screen display 356 Mosaic window 350 containing image window 352 and a portion of a panel window 354. Note that image window 352 is within Mosaic window 350 while panel window 354 is external to Mosaic window 350. Another possibility is to have panel window 354 within Mosaic window 350. By using the controls in panel window 354 the user is able to manipulate the image within image window 352 in real time do perform such operations as scaling, rotation, translation, color map selection, etc. In Fig. 9, two Mosaic windows are being used to show two different views of an embryo image. One of the views is rotated by six degrees from the other view so that a stereoscopic effect can be achieved when viewing the images. Communication between Panel and VIS is via "Tooltalk" described in, e.g., "Tooltalk 1.1.1 Reference Manual," from SunSoft.

Fig. 10 is an illustration of the processes VIS, Panel and VRServer discussed above. As shown in Fig. 10, the browser process, Mosaic, communicates with the Panel process via inter-client communication mechanisms such as provided in the X-Window environment. The Panel process communicates with the VIS process through a communications protocol (ToolTalk, in the preferred embodiment) to exchange visualization command messages and image data. The image data is computed by one or more copies of a process called VRServer that may be executing on remote computers on the network. VRServer processes respond to requests such as rendering requests to generate image segments. The image segments are sent to VIS and combined into a pixmap, or frame image, by VIS. The frame image is then transferred to the Mosaic screen via communications between VIS, Panel and Mosaic. A further description of the data transfer may be found in the paper "Integrated Control of Distributed Volume Visualization Through the World-Wide-Web," referenced above.

In the foregoing specification, the invention has been described with reference to a specific exemplary embodiment thereof. It will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the appended claims. For example, various programming languages and techniques can be used to implement the disclosed invention. Also, the specific logic presented to accomplish tasks within the present invention may be modified without departing from the scope of the invention. Many such changes or modifications will be readily apparent to one of ordinary skill in the art. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense, the invention being limited only by the provided claims.

WHAT IS CLAIMED IS:

Sub B1

1. A method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment;

displaying, on said client workstation, at least a portion of a hypermedia document received over said network from said server, wherein said hypermedia document includes an embedded controllable application; and

interactively controlling said embedded controllable application from said client workstation via communications sent over said distributed hypermedia environment.

2. The method of claim 1, wherein the step of displaying is performed by using a hypermedia browser application.

3. The method of claim 2, wherein instructions for controlling said embedded controllable application reside on said network server, wherein said step of interactively controlling said embedded controllable application includes the following substeps:

issuing, from the client workstation, one or more commands to the network server;

executing, on the network server, one or more instructions in response to said commands;

sending information from said network server to said client workstation in response to said executed instructions; and

processing said information at the client workstation to interactively control said embedded controllable application.

30

4. The method of claim 2, wherein instructions for controlling said embedded controllable application reside on said client workstation.

5. The method of claim 2, wherein the communications to interactively control said embedded controllable application from said client workstation continue to be exchanged between the controllable application and the hypermedia browser even after the controllable application program has been launched.

6. The method of claim 3, wherein said embedded controllable application is a multi-dimensional viewer.

7. The method of claim 3, wherein said embedded controllable application is a spreadsheet program.

8. The method of claim 3, wherein said embedded controllable application is a database program.

9. The method of claim 3, wherein said embedded controllable application is a word processor.

10. The method of claim 3, wherein said substeps of issuing and sending are via an open protocol.

11. The method of claim 10, wherein said open protocol is an International Standards Organization (ISO) protocol.

12. The method of claim 11, wherein said ISO protocol is Transfer Control Protocol/Internet Protocol (TCP/IP) and said network is the Internet.

13. The method of claim 12, wherein HyperText Transfer Protocol is used to transfer said hypermedia document between said client workstation and said server.

14. The method of claim 13, wherein HyperText Markup Language is used to specify said embedded controllable application within said hypermedia document.

15. A method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, said network including a plurality of general purpose workstations, wherein said network environment is a distributed hypermedia environment;

displaying, on said client workstation, at least a portion of a hypermedia document received over said network from said server, wherein said hypermedia document includes at least a first embedded multi-dimensional data visualization application; and

interactively controlling said embedded multi-dimensional data visualization application from said client workstation via communications sent over said distributed hypermedia environment wherein data image rendering is performed by said plurality of general purpose workstations using distributed processing.

16. The method of claim 15, wherein the step of displaying is performed by using a hypermedia browser application.

17. The method of claim 15, wherein the multi-dimensional data visualization includes volume visualization.

18. The method of claim 15, wherein the multi-dimensional data visualization includes two dimensional image processing.

19. The method of claim 15, wherein the multi-dimensional data visualization includes image analysis.

20. The method of claim 15, wherein the multi-dimensional data visualization includes the display of animated sequences.

21. The method of claim 15, wherein the multi-dimensional data visualization includes a geometric data viewer to display computer aided design files.

22. The method of claim 15, wherein the multi-dimensional data visualization includes displaying molecular modeling data.

~~Sub B3~~

23. The method of claim 15, wherein a hypermedia browser is executing on the client workstation, wherein communications to interactively control said embedded controllable application from said client workstation continue to be exchanged between the controllable application and the hypermedia browser even after the controllable application program has been launched.

24. A method for interactively controlling an embedded object in a document displayed on a client computer, wherein the client computer includes a processor coupled to a display device and to a user input device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a document, wherein the document includes an embedded object, wherein an application program for manipulating the embedded object resides on a first additional computer, the method comprising the following steps:

transferring, over the network, at least a portion of the document from the server computer to the client computer;

accepting first signals from the user input device that indicate that the embedded object is to be manipulated;

issuing commands from the client computer to the first additional computer in response to the first signals;

executing, by using the first additional computer, instructions in the application program in response to the issued commands, wherein the executed instructions generate information about manipulating the embedded object;

communicating, via the network, the information about manipulating the embedded object from the first additional computer to the client computer; and

using the client computer to manipulate the embedded object according to the communicated information.

25. The method of claim 24, wherein said document is a hypermedia document.

Sub A4

26. The method of claim 24, further comprising the steps of executing instructions in a second application program on a second additional computer in response to the issued commands, wherein the instructions executed by the second additional computer result in information about manipulating the embedded object being generated more quickly.

27. The method of claim 26, wherein said document is a hypermedia document.

Sub A5

28. The method of claim 26, wherein the embedded object is a multi-dimensional image displayable in any of a plurality of orientations.

29. The method of claim 28, wherein said document is a hypermedia document.

30. The method of claim 28, wherein the executed instructions perform three dimensional display transformations to determine the second orientation of the multi-dimensional image object.

31. The method of claim 30, wherein said document is a hypermedia document.

32. The method of claim 28, wherein the executed instructions perform image rendering to determine an orientation of the multi-dimensional image.

33. The method of claim 32, wherein said document is a hypermedia document.

Sub A6

34. A method for displaying a three dimensional image object on a client computer, wherein the client computer includes a processor coupled to a display device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a hypermedia document, wherein the hypermedia document includes a three dimensional image object embedded within the hypermedia document, wherein the three dimensional image object is displayable in a plurality of orientations, the method comprising the following steps:

transferring, over the network, at least a portion of the hypermedia document from the server computer to the client computer;

displaying on the display device, by using the processor, at least a portion of the hypermedia document, wherein the displayed portion of the hypermedia document includes the three dimensional image object displayed in a first orientation;

using the client computer to issue commands over the network;

executing instruction on a first additional computer in response to the issued commands, wherein the executed instructions determine a second orientation for display of the three dimensional image object;

communicating, via the network, information about the second orientation from the first additional computer to the client computer; and

using the client computer to redisplay the three dimensional image object in the second orientation.

35. The method of claim 34, wherein said network is a distributed hypermedia environment.

36. The method of claim 34, further comprising the steps of executing instructions on a second additional computer in response to the issued commands, wherein the instructions executed by the second computer enable the second orientation to be determined more quickly than when only the first additional computer executes instructions.

37. The method of claim 36, wherein said network is a distributed hypermedia environment.

38. The method of claim 36, wherein the executed instructions perform volume rendering to determine the second orientation of the three dimensional image object.

39. The method of claim 38, wherein said network is a distributed hypermedia environment.

40. The method of claim 36, wherein the executed instructions perform three dimensional display transformations to determine the second orientation of the three dimensional image object.

41. The method of claim 40, wherein said network is a distributed hypermedia environment.

42. The method of claim 34, wherein the client computer includes a user input device coupled to the processor, the method further comprising the following steps:

accepting signals from the user input device, wherein the accepted signals indicate that the second orientation is to be determined.

43. The method of claim 42, wherein said network is a distributed hypermedia environment.

08/324,443

EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

ABSTRACT OF THE DISCLOSURE

A system allowing a user of a browser program on a computer connected to an open distributed hypermedia system to access and execute an embedded program object. The program object is embedded into a hypermedia document much like data objects. The user may select the program object from the screen. Once selected the program object executes on the user's (client) computer or may execute on a remote server or additional remote computers in a distributed processing arrangement. After launching the program object, the user is able to interact with the object as the invention provides for ongoing interprocess communication between the application object (program) and the browser program. One application of the embedded program object allows a user to view large and complex multi-dimensional objects from within the browser's window. The user can manipulate a control panel to change the viewpoint used to view the image. The invention allows a program to execute on a remote server or other computers to calculate the viewing transformations and send frame data to the client computer thus providing the user of the client computer with interactive features and allowing the user to have access to greater computing power than may be available at the user's client computer.

Attorney Docket No. 02307-553

DECLARATION

As a below named inventor, I declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: **EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS** the specification of which _X_ is attached hereto or ___ was filed on ________ as Application Serial No. ______ and was amended on ___ (if applicable).

I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56. I claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign applications(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Country	Application No.	Date of Filing	Priority Claimed Under 35 USC 119
			Yes ____ No ____
			Yes ____ No ____

I claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, section 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial No.	Date of Filing	Status
		___ Patented ___ Pending ___ Abandoned
		___ Patented ___ Pending ___ Abandoned

Full Name of Inventor 1	Last Name **Doyle**	First Name **Michael**	Middle Name or Initial	
Residence & Citizenship	City **Alameda**	State/Foreign Country **California**	Country of Citizenship **USA**	
Post Office Address	Post Office Address **5 Remmel Court**	City **Alameda**	State/Country **California**	Zip Code **94502**
Full Name of Inventor 2	Last Name **Martin**	First Name **David**	Middle Name or Initial	
Residence & Citizenship	City **San Jose**	State/Foreign Country **California**	Country of Citizenship **USA**	
Post Office Address	Post Office Address **5572 Makati Circle**	City **San Jose**	State/Country **California**	Zip Code **95123**
Full Name of Inventor 3	Last Name **Ang**	First Name **Cheong**	Middle Name or Initial	
Residence & Citizenship	City **Pacifica**	State/Foreign Country **California**	Country of Citizenship **Malaysia**	
Post Office Address	Post Office Address **658 Hickey Blvd.**	City **Pacifica**	State/Country **Pacifica**	Zip Code **94044**

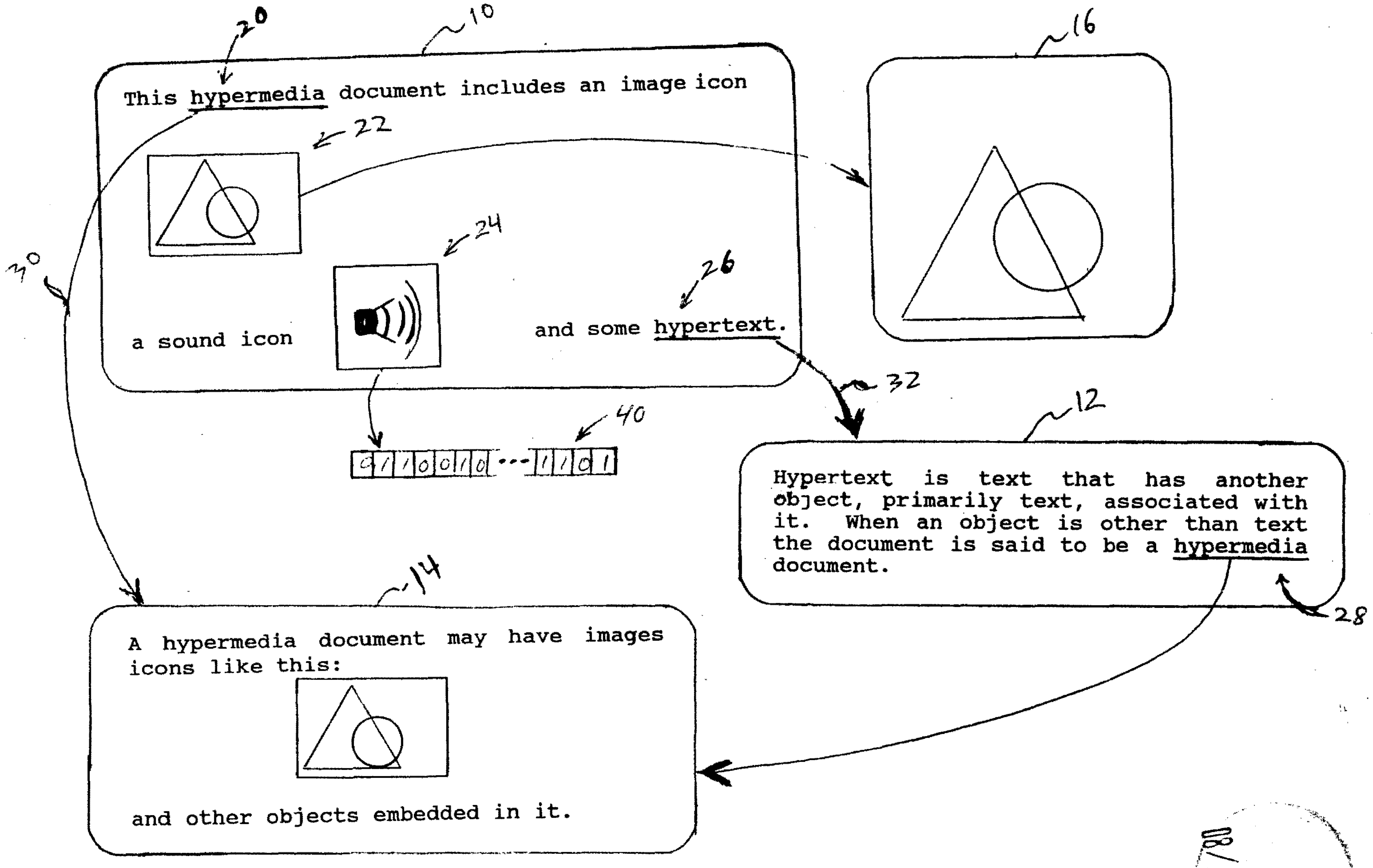
I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature of Inventor 1 **Michael Doyle**	Signature of Inventor 2 **David Martin**	Signature of Inventor 3 **Cheong Ang**
Date	Date	Date

DEC.MRG 1/93

(Page 1 of 1)

PRIOR ART



This hypermedia document includes an image icon

a sound icon

and some hypertext.

Hypertext is text that has another object, primarily text, associated with it. When an object is other than text the document is said to be a hypermedia document.

A hypermedia document may have images icons like this:

and other objects embedded in it.

Fig. 1

PRIOR ART

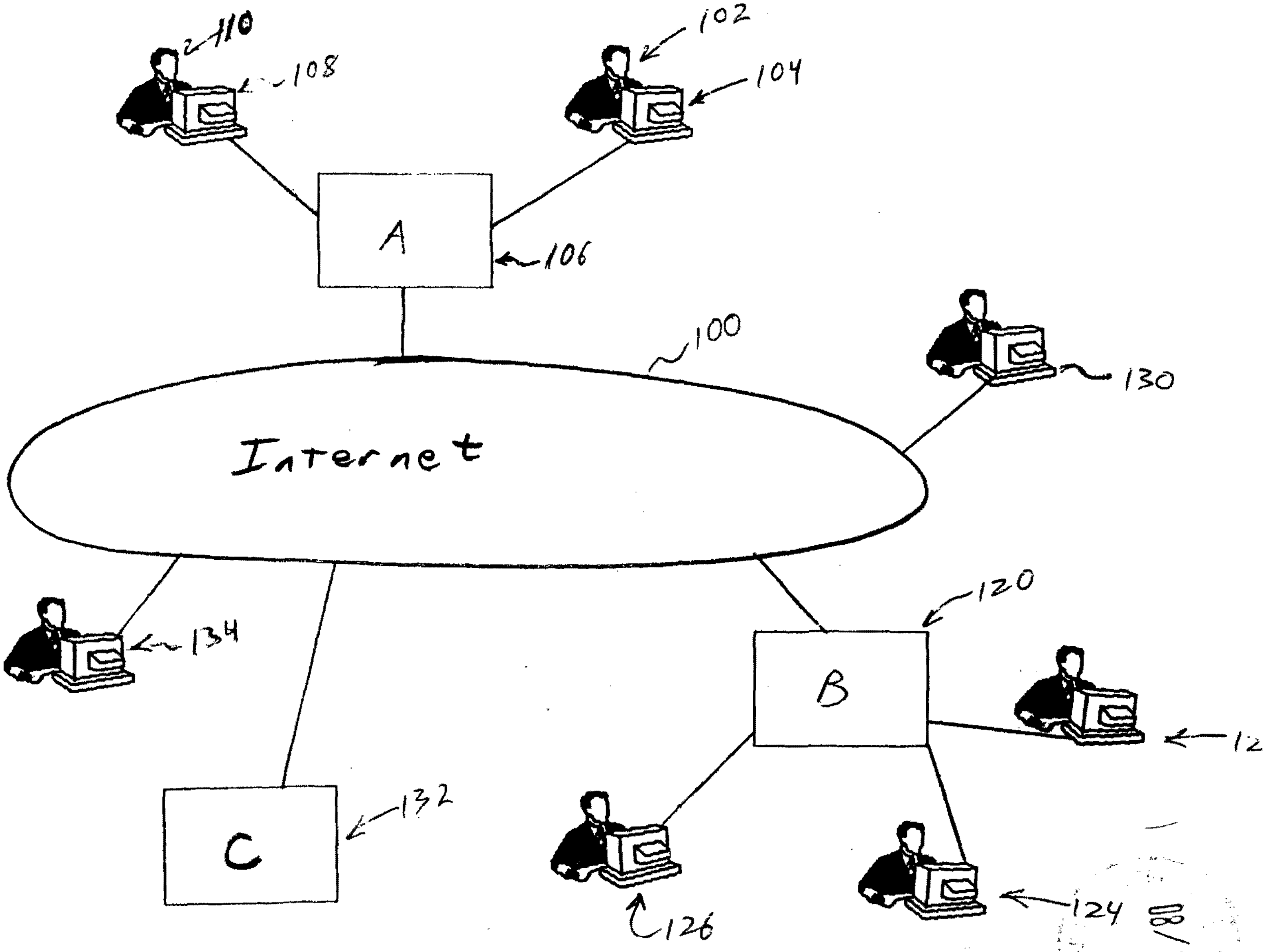


Fig. 2

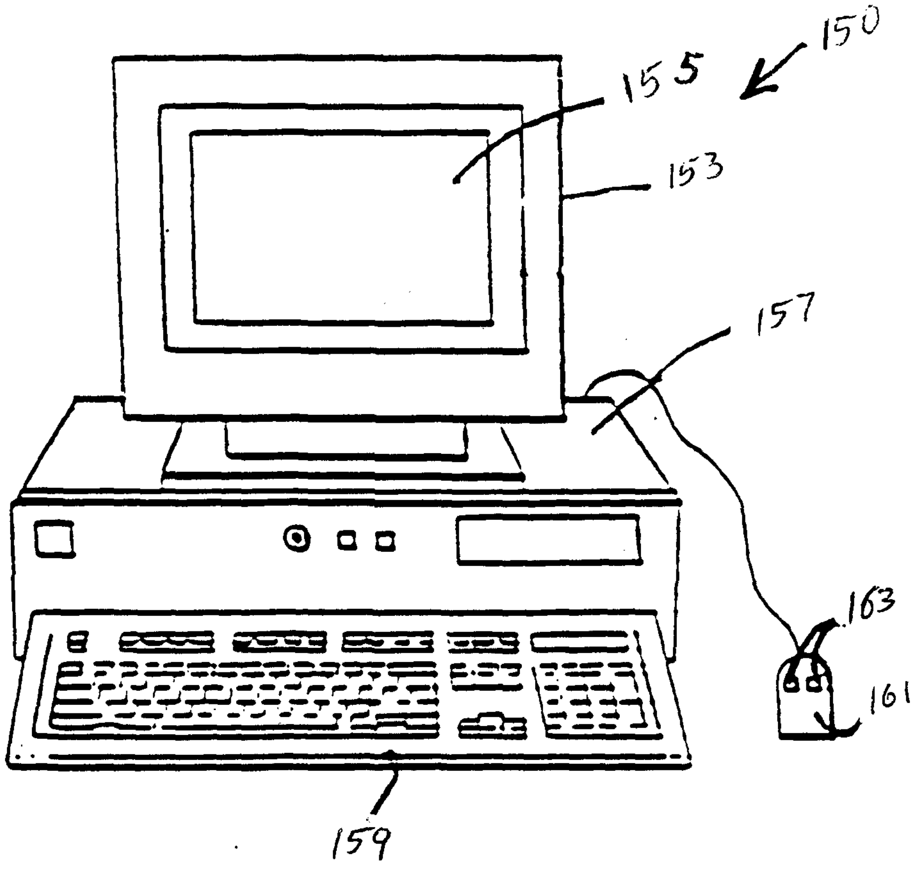
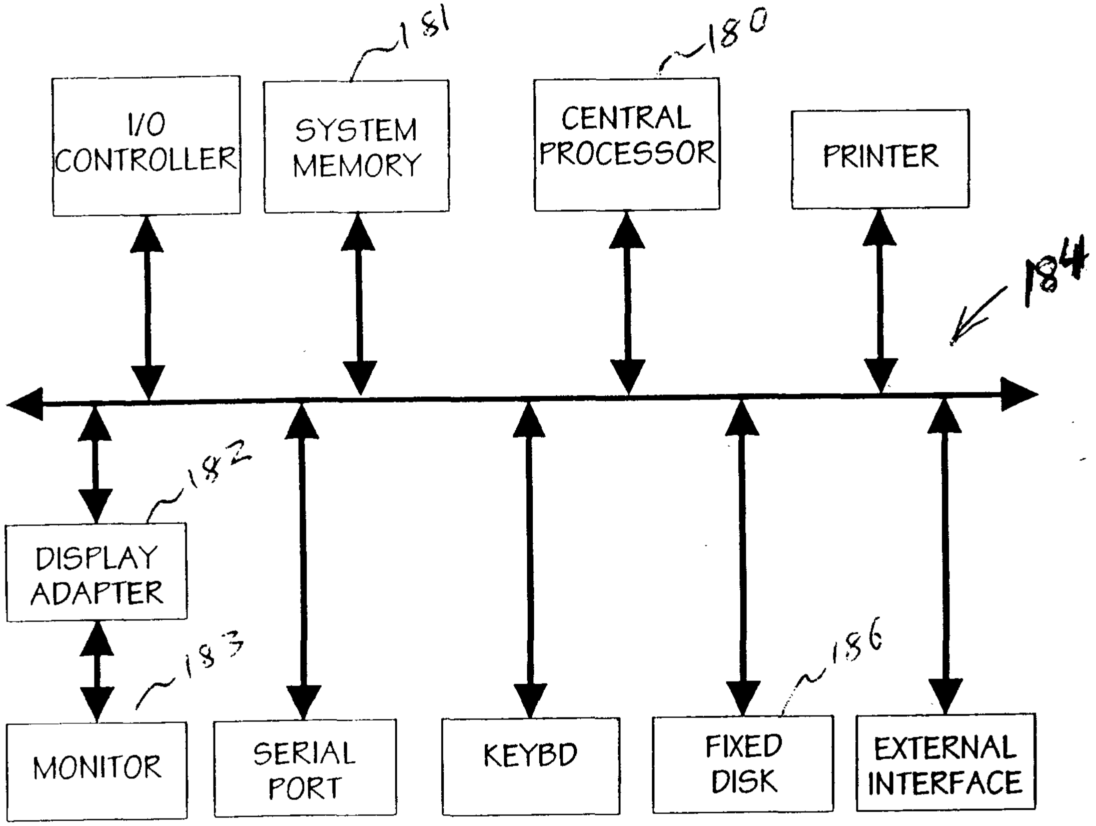


Fig 3



181
180
I/O CONTROLLER
SYSTEM MEMORY
CENTRAL PROCESSOR
PRINTER
184
182
DISPLAY ADAPTER
183
186
MONITOR
SERIAL PORT
KEYBD
FIXED DISK
EXTERNAL INTERFACE

Fig. 4

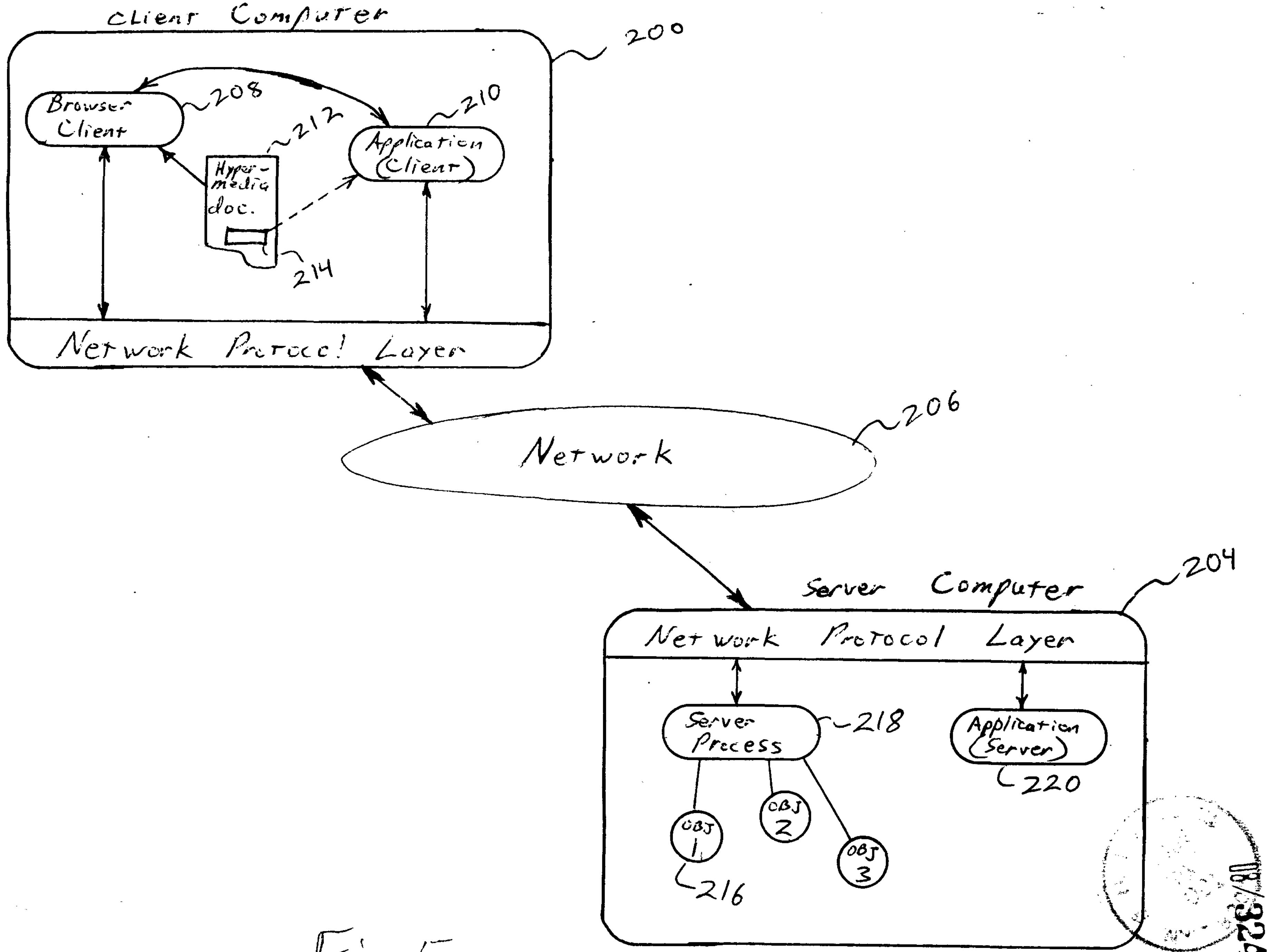

Client Computer
200
Browser Client
208
212
Application (Client)
210
Hyper-media doc.
214
Network Protocol Layer
Network
206
Server Computer
204
Network Protocol Layer
Server Process
218
Application (Server)
220
OBJ 1
216
OBJ 2
OBJ 3

Fig. 5

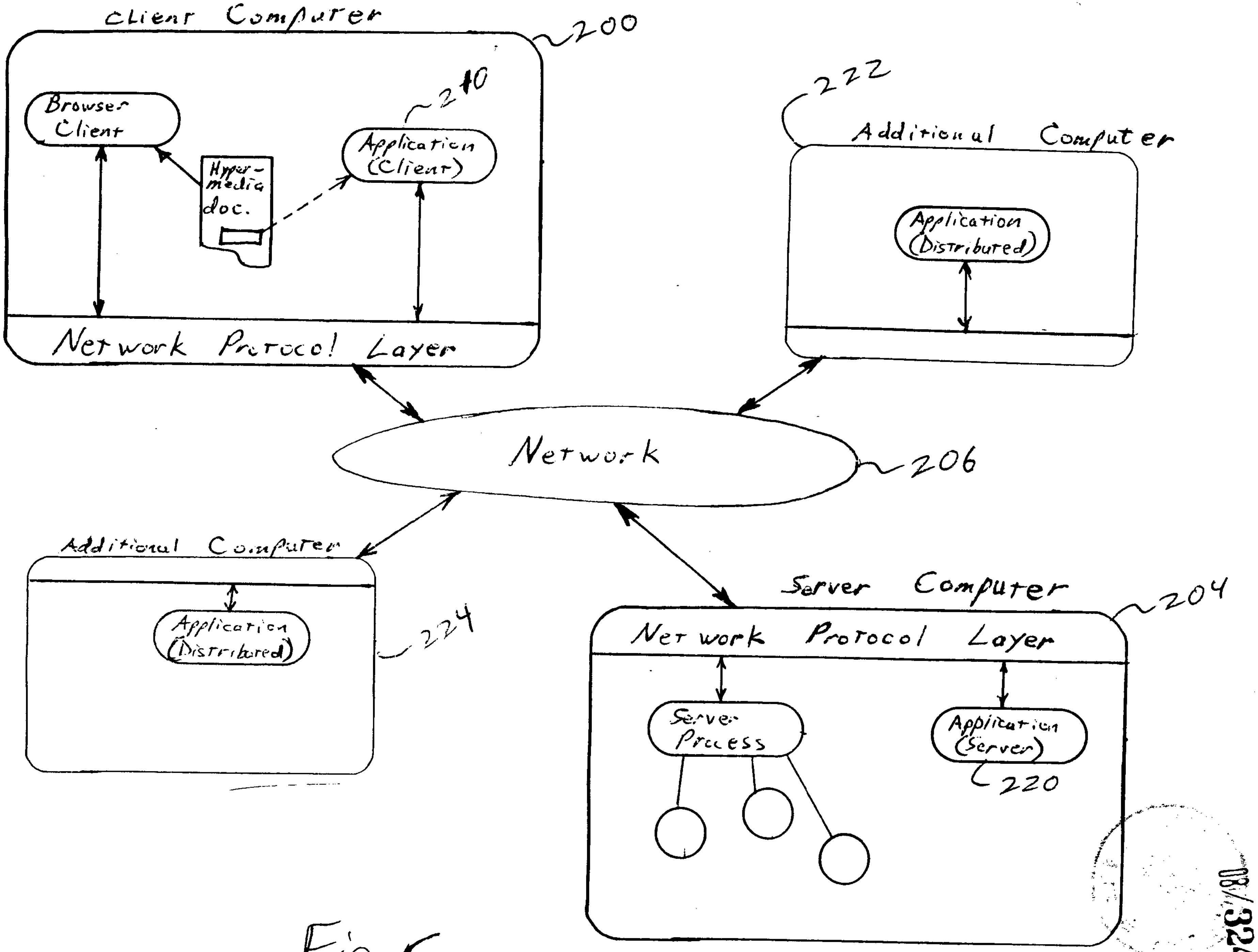
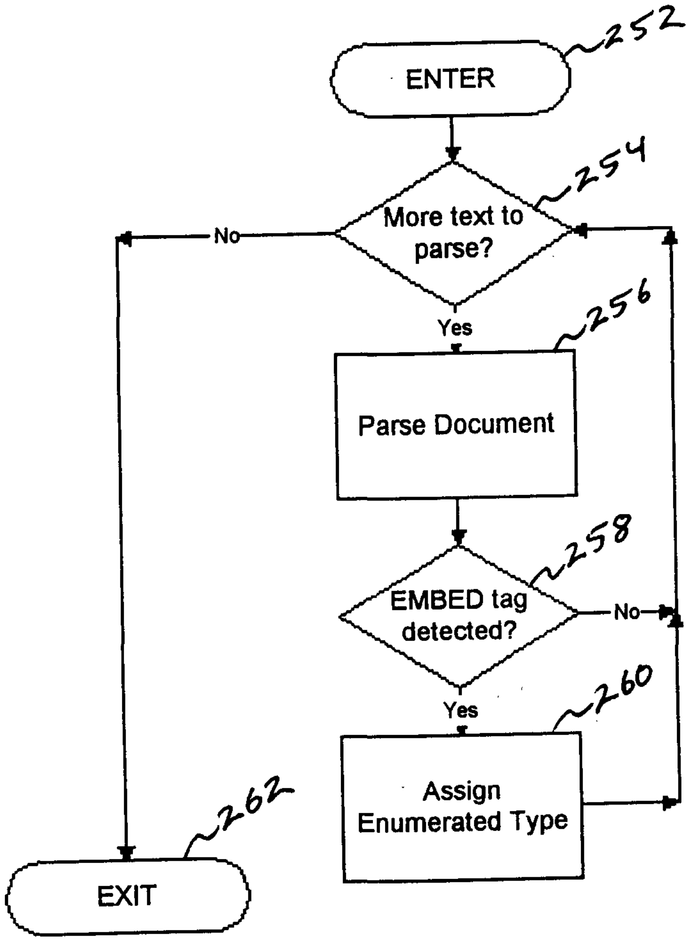

Client Computer
200
Browser Client
Hyper-media doc.
210
Application (Client)
Network Protocol Layer
222
Additional Computer
Application (Distributed)
Network
206
Additional Computer
Application (Distributed)
224
Server Computer
204
Network Protocol Layer
Server Process
Application (Server)
220

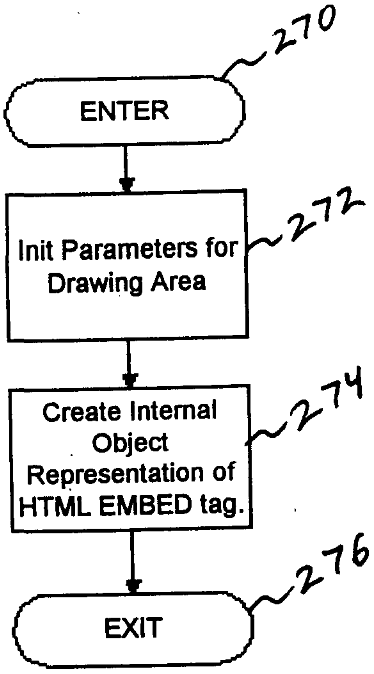
Fig. 6

08/324443



HTMLparse.c

Fig. 7A



ENTER
270
Init Parameters for Drawing Area
272
Create Internal Object Representation of HTML EMBED tag.
274
EXIT
276

HTMLFormat.c

Fig. 7B

HTMLwidget.c

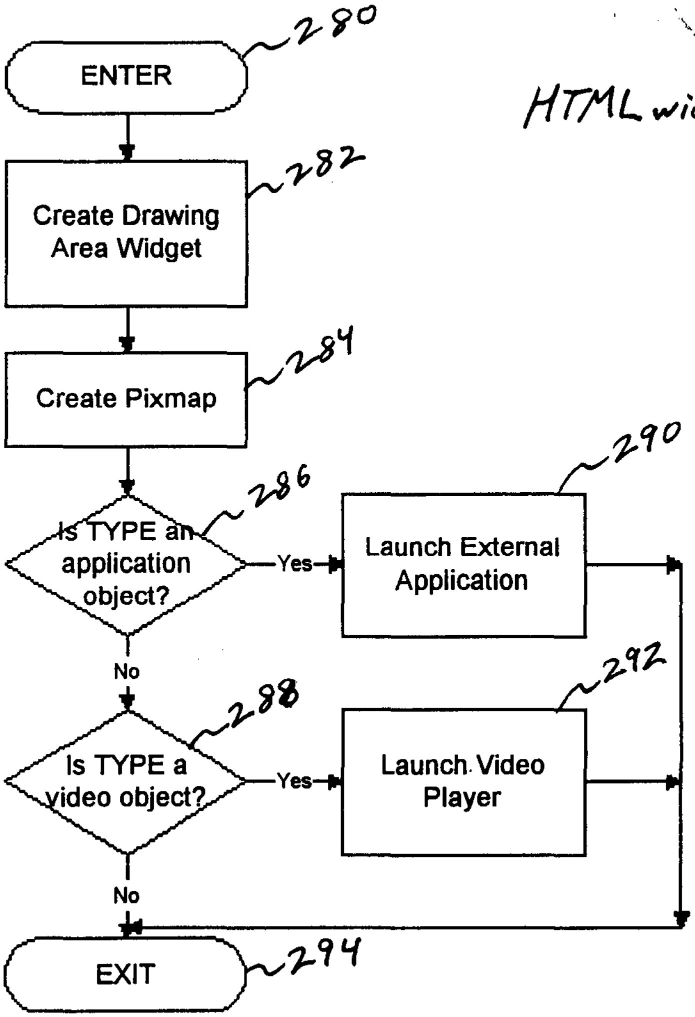


Fig. 8A

HTML.c

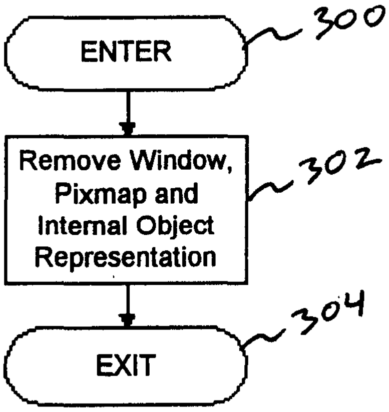
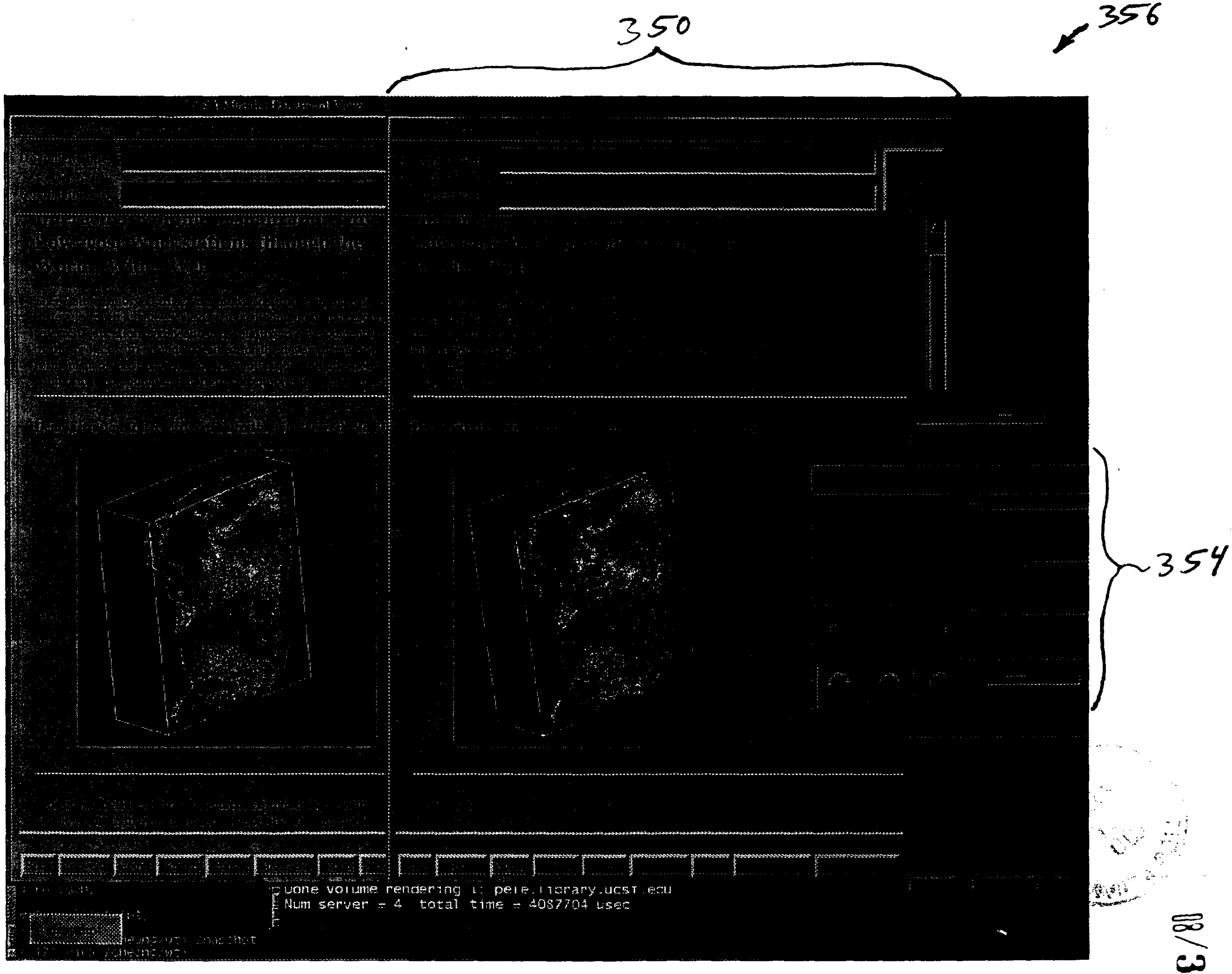


Fig. 8B

08/324443



Done volume rendering in pele.library.ucsf.edu
Num server = 4 total time = 4087704 usec

Fig. 9

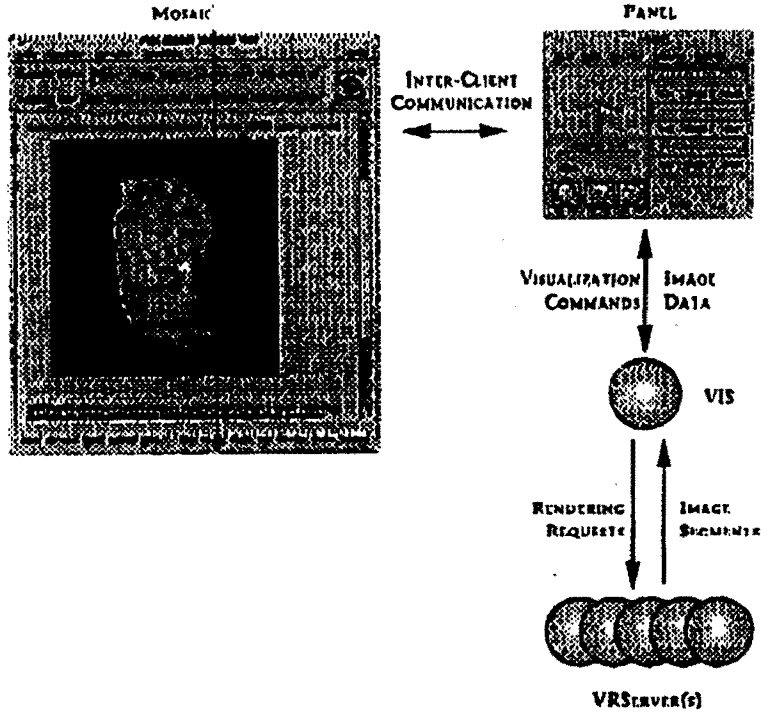

Mosaic
Panel
Inter-Client Communication
Visualization Commands
Image Data
VIS
Rendering Requests
Image Segments
VRServer(s)

Fig. 10

906 PH Ex. 2

UNITED STATI , DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
08/324,443	10/17/94	DOYLE	M 02307553

EXAMINER	
DINH,D	
ART UNIT	**PAPER NUMBER** 9Q
2317	4

B3M1/0506

TOWNSEND AND TOWNSEND
KHOURIE AND CREW
STEUART STREET TOWER
ONE MARKET PLAZA
SAN FRANCISCO CA 94105

DATE MAILED: 05/06/96

This is a communication from the examiner in charge of your application.
COMMISSIONER OF PATENTS AND TRADEMARKS

☒ This application has been examined ☐ Responsive to communication filed on ________ ☐ This action is made final.

A shortened statutory period for response to this action is set to expire 3 month(s), 0 days from the date of this letter.
Failure to respond within the period for response will cause the application to become abandoned. 35 U.S.C. 133

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☒ Notice of References Cited by Examiner, PTO-892.
2. ☒ Notice of Draftsman's Patent Drawing Review, PTO-948.
3. ☒ Notice of Art Cited by Applicant, PTO-1449.
4. ☐ Notice of Informal Patent Application, PTO-152.
5. ☐ Information on How to Effect Drawing Changes, PTO-1474.
6. ☐ ________.

Part II SUMMARY OF ACTION

1. ☒ Claims 1-43 are pending in the application.

   Of the above, claims ________ are withdrawn from consideration.

2. ☐ Claims ________ have been cancelled.

3. ☐ Claims ________ are allowed.

4. ☒ Claims 1-43 are rejected.

5. ☐ Claims ________ are objected to.

6. ☐ Claims ________ are subject to restriction or election requirement.

7. ☒ This application has been filed with informal drawings under 37 C.F.R. 1.85 which are acceptable for examination purposes.

8. ☐ Formal drawings are required in response to this Office action.

9. ☐ The corrected or substitute drawings have been received on ________. Under 37 C.F.R. 1.84 these drawings are ☐ acceptable; ☐ not acceptable (see explanation or Notice of Draftsman's Patent Drawing Review, PTO-948).

10. ☐ The proposed additional or substitute sheet(s) of drawings, filed on ________. has (have) been ☐ approved by the examiner; ☐ disapproved by the examiner (see explanation).

11. ☐ The proposed drawing correction, filed ________, has been ☐ approved; ☐ disapproved (see explanation).

12. ☐ Acknowledgement is made of the claim for priority under 35 U.S.C. 119. The certified copy has ☐ been received ☐ not been received ☐ been filed in parent application, serial no. ________ ; filed on ________.

13. ☐ Since this application apppears to be in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under Ex parte Quayle, 1935 C.D. 11; 453 O.G. 213.

14. ☐ Other

EXAMINER'S ACTION

Part III DETAILED ACTION

The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless --
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

Claims 1-4 and 15-16 are rejected under 35 U.S.C. § 102(a) as being anticipated by the University of Southern California's Mercury Project (in public used from Sept. 15, 1994 to March 31, 1995 at http://www.usc.edu/dept/raiders/).

As per claim 1, the Mercury Project operated using a method essentially as claimed:

providing client workstation [user computer browsing the WWW] and network server [the Mercury Project server]coupled to a distributed hypermedia network environment [World Wide Web;

displaying at the client workstation a portion of a hyper media document [HTML document] wherein the document includes an embedded controllable application [controlling a robot arm and air pulse - see page 1 of "USC Mercury Project: Interface"];

interactively controlling said embedded controllable application from said client workstation via communications sent over said distributed hypermedia environment [see page 1 of "USC Mercury Project: Interface"].

As per claim 2, the Mercury project was accessed using multimedia browser (specifically Mosaic - see p.2 of "USC PRESS RELEASE") running on the client.

As per claim 3, the steps recited is inherent in using the Mercury Project:

issuing from the client one or more commands to the server [HTTP message conveying information resulting from a user clicking on the robot control panel image];

executing, on the network server, instructions in response to said commands [interpreting and responding to the HTTP message];

sending information [HTML document showing the updated robot position and camera image] from the server to the client;

processing said information [parsing the HTML document] at the client to interactively control said embedded controllable application.

As per claim 4, since the browser is controlling the displaying the HTML document and the browser's code/instructions is running on the client computer, the 'instructions' for controlling said embedded application [i.e. interpreting user input and sending HTTP message to the server] resides on the client.

As per claims 15-16, they are rejected under similar rationale as for claims 1-2 above. The Mercury Project is a 'multi-dimensional' data visualization application because it provides live-image of a real (3D) environment and position of the robot arm.

The following is a quotation of 35 U.S.C. § 103 which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.
>
> Subject matter developed by another person, which qualifies as prior art only under subsection (f) or (g) of section 102 of this title, shall not preclude patentability under this section where the subject matter and the claimed invention were, at the time the invention was made, owned by the same person or subject to an obligation of assignment to the same person.

Claims 1-43 are rejected under 35 U.S.C. § 103 as being unpatentable over Applicant's disclosed prior art and further in view of Hansen "Enhancing documents with embedded programs: How Ness extends insets in the Andrew ToolKit".

As per claim 1, Applicant's disclosed prior art has the limitation essentially as claimed - client workstation, network server, receiving and displaying hypermedia document. It is known at the time of the invention that the client and server interact via communications sent over the distributed hypermedia environment (transferring of HTML document, processing links, forms, mapped images, etc.). The disclosed prior art does not have embedded controllable application [executable/ interpretable/ 'launchable' program instructions/ codes] in the hypermedia document].

Hansen teaches enhancing hypertext documents by embedding programs in the documents to allows the documents to responds to readers for applications like animation, simulations, interactive examples etc.[Abstract, p.23 col.1]. It would have been obvious for one of ordinary skill in the art to combine the teaching of Hansen to the disclosed prior art because it would have improved hypermedia information delivery and applications.

As per claim 2, the disclosed prior art teaches a hypermedia browser [Mosaic].

As per claim 3, it is known in the disclosed prior art to issue command from the client to trigger execution on the server and sending response back to the client [processing forms and mapped images, etc.]. Hence, the steps would have been apparent in the method as modified.

As per claim 4, Hansen teaches that the embedded program is an object within the hypermedia document. The document is being displayed by a "browser" in the client workstation. Hence, it is apparent that the instructions for activating/controlling the embedded program would reside on the client workstations (i.e. within the browser and/or within the client workstation operating environment).

As per claim 5, it is apparent from p.24 col.2 "response to event", and p.25 4th paragraph -simulating for mouse hit, menu selections, and keystrokes that communication continue between the embedded program and the "browser" to accept and act on trigger events.

As per claims 6-10, Hansen, p.23 col.1, suggested multi-dimensional viewer [simulation, animation], spreadsheet and database [addresses, appointments, course records], and word processing [parameterized letter] applications.

As per claims 11-13, the recited limitations - ISO, TCP/IP, HTTP - are inherent the disclosed prior art.

As per claim 14, Hansen teaches using markup language to specify the embedded program [p.25 col.2].

As per claims 15-16, they are rejected under similar rationale as for claims 1-2 above.

As per claims 17-22, the recited limitations - volume visualization, 2d image, image analysis, animated sequences, geometric viewer, and molecular modeling - would have been a matter of design choice because they are merely a listing of possible uses and does not affect the method of providing controlling of the embedded program. Hansen disclosed applying the 'enhanced document' in simulations and animation [p.24 col.1]. Hence, the applications recited would have been obvious to one of ordinary skill in the art.

As per claim 23, it is rejected under similar rationale as for claim 4 above.

As per claim 24, it is rejected under similar rationale as for claim 1 above. Hansen does not specifically disclose the step of transferring ..., accepting ..., executing ..., communicating ..., using ... The steps recited is inherent in the disclosed prior art as modified because:

> It is well known in the art, at the time of the invention, that HTML documents can contains links, mapped (clickable) images, fill-in forms, etc. It is known that HTML documents transfers involves HTTP protocol messages. The process involves:
>
> transferring, over the network, a hypermedia document [the HTML document] with embedded objects [URL links, mapped

images, fill-in forms, etc.] from a server computer to the client computer;

accepting first signals from the user input device [clicking on an URL link, or a mapped image, or a form's 'submit' button]

issuing commands [HTTP message with the linked URL, or coordinates where the mapped image was clicked, or the form's content] from the client computer to a first computer in response to the signal [it is known that an HTTP message in an HTML document can direct to any computer connected to the Internet that accept HTTP protocol];

executing instructions by the first computer and generate information about manipulating the embedded object [retrieving or generating a HTML document in response to the HTTP message];

communicating the information to the client [sending the resulting HTML document];

using the client to manipulate the embedded object [displaying the result HTML document by the browser] according to the communicated information [HTML tags in the received HTML document].

As per claims 25, 27, 29, 31, 33, an HTML document is a hypermedia document.

As per claim 26, it is well known in the art to run application on multiple computer for faster performance. It would have been obvious for one of ordinary skill in the art to run the application on multiple computers because it would have reduce processing time.

As per claim 28, the recited limitation is one of many possible applications for the method of claim 24. It would have been a matter of design choice to provide multi-dimensional image visualization application because the particular recited application does not affect the method of claim 24. It is well known in the art to provide multi-dimensional image visualization in the scientific community. Hence, it would have been obvious for one of ordinary skill in the art to provide multi-dimensional image visualization application in the prior art as modified because it would enable wide spread access to scientists to remotely analyze the images.

As per claims 30, 32, 38, 40, the recited methods are well known in the art for manipulating 3D images.

As per claim 34, it is a combination of claims 24 and 28. Hence, claim 34 is rejected under similar rationales as stated for claims 24 and 28 above.

As per claims 35, 37, 39, 41, 43, the World Wide Web is a distributed hypermedia environment.

As per claim 36, it is rejected under similar rationale as for claim 26 above.

As per claim 42, the method would have been inherent in the prior art as modified. It would have been obvious to one of ordinary skill in the art to accepted input indicating desired orientation in a multi-dimensional image visualization application because it would enable the user to effectively visualize and manipulate the multi-dimensional image.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Tim Berners-Lee, "Hypertext Markup Language (HTML)", June 1993, discusses and discloses features of HTML.

Berstein et al., US patent 5,204,947, disclose method for providing links in hypermedia services without modifying the documents.

Vertelney et al., US patent 5,202,828, teach user interface icon with embedded applications.

Caro, US patent 4,949,248, teaches a system for shared remote access with application programs executing on more than one computer.

Swanson, US patent 5,390,314, teaches resource script that can be execute on various different computer systems.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dung Dinh whose telephone number is (703) 305-9655. The examiner can normally be reached on Monday-Thursday from 7:00 AM - 4:30 PM. The examiner can also be reached on alternate Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas Lee can be reached at (703) 305-9717. The fax phone number for this group is (703) 308-5359.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

D Dinh

D Dinh
Patent Examiner
May 3, 1996

Notice of References Cited

Application No. 08/324,443	Applicant(s) Doyle et al.	
Examiner D. Dinh	Group Art Unit 2317	Page 1 of 1

U.S. PATENT DOCUMENTS

	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
A	5,390,314	02/14/95	Swanson Filed 10/09/92	395	500
B	4,949,248	08/14/90	Caro	395	200.03
C	5,204,947	04/20/93	Bernstein et al.	395	157
D	5,202,828	04/12/93	Vertelney et al.	364	419
E					
F					
G					
H					
I					
J					
K					
L					
M					

FOREIGN PATENT DOCUMENTS

	DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
N						
O						
P						
Q						
R						
S						
T						

NON-PATENT DOCUMENTS

	DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
U	Berners-Lee "Hypertext Markup Language (HTML)", HTML Internet Draft, IIIR working Group.	06/93
V	University of Southern California's Mercury Project - "USC Mercury Project:Interface", "Project Milestones", "USC PRESS RELEASE" - obtained from Internet, http://www.usc.edu/dept/raiders/	No date
W	Hansen, Wilfred "Enhancing documents with embedded programs: How Ness extends insets in the Andrew ToolKit", IEEE Computer Languages, 1990 International Conference.	
X		

LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT	Docket No.:2307U-553 S[illegible]al No.: 08/324,443 Applicant(s): Michael Doyle, et al. Group: TBA Filing Date: October 17, 1994 Page No: 1

U.S. PATENT DOCUMENTS

Examiner Initial	Ref No.	Document Number	Document Date	Name	Class/ Subclass	Filing Date if approp.
	AA	4,847,604	07/11/89	Doyle	340/706	
	AB	5,307,499	04/26/94	Yin	395/700	
	AC	5,321,806	06/14/94	Meinerth, et al.	395/162	
	AD	5,321,808	06/14/94	Rupp, et al.	395/164	

FOREIGN PATENT DOCUMENTS

Examiner Initial	No.	Document Number	Date	Country	Class/ Subclass	Translation Yes	No

OTHER REFERENCES

Examiner Initial	No.	Author, Title, Date, Pertinent Pages, Etc.
	AE	Tani, M., et al., "Object-Oriented Video: Interaction with Real-World Objects Through Live Video", May 1992, pp. 593-598.
	AF	Crowley, T., et al., "MMConf: An Infrastructure for Building Shared Multimedia Applications", CSCW 90 Proceedings, October 1990, pp. 329-342
	AG	Davis, H., et al., "Towards An Integrated Information Environment With Open Hypermedia System", ACM ECHT Conference, December 1992, pp. 181-190.
	AH	Ferrara, F., "The KIM Query System", Abstract, SIGCHI Bulletin, Vol. 6, No. 3, July 1994, pp. 30-39.

Examiner: D. DINH	Date Considered: 4/16/96

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP §609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

SUBSTITUTE DISCLOSURE STATEMENT FORM (PTO-1449)

LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT	Docket No.:2307U-553 S ial No.: 08/324,443 Applicant(s): Michael Doyle, et al. Group: TBA Filing Date: October 17, 1994 Page No: 2

AI Gibbs, S., "Composite Multimedia and Active Objects", OOPSLA '91, pp. 97-112.

AJ Davis, H., et al., "Microcosm: An Open Hypermedia System", Interchi '93, April 1993, p. 526.

AK Vaziri, A., "Scientific Visualization in High-Speed Network Environments", Computer Networks and ISDN Systems 22, 1991, pp. 111-129.

AL Cullen, J., et al., "The Use of FTAM to access graphical pictures across wide area networks", Computer Networks and ISDN Systems, 1992, pp. 337-383.

AM Lashkari, Y.Z., et al., "PLX: A Proposal to Implement a General Broadcasting Facility in a Distributed Environment Running X Windows", Comput. & Graphics, Vol. 16, No. 2, pp. 143-149, 1992.

AN Kirste, T., "Spacepicture-An Interactive Hypermedia Satellite Image Archival System", Comput. & Graphics, Vol. 17, No. 3, pp. 251-260, 1993.

AO Coulson, G., et al., "Extensions to ANSA for Multimedia Computing", Computer Networks and ISDN Systems 25, 1992, pp. 305-323.

AP Huynh, Duong Le, et al., "PIX: An Object-Oriented Network Graphics Environment", Comput. & Graphics, Vol. 17, No. 3, pp 295-304, 1993.

AQ Berners-Lee, T.J., et al., "The World-Wide Web, Computer Networks and ISDN Systems 25, 1992, pp. 454-459.

AR Shackelford, D.E., et al., "The Architecture and Implementation of a Distributed Hypermedia Storage System", Hypertext '93 Proceedings, November 1993, pp. 1-13.

AS Labriola, D., "Remote Possibilities", PC Magazine, June 14, 1994, pp. 223-228.

Examiner: D. DINH	Date Considered: 4/16/96

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP §609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

SUBSTITUTE DISCLOSURE STATEMENT FORM (PTO-1449)

LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT

Docket No.:2307U-553 S. .al No.: 08/324,443
Applicant(s): Michael Doyle, et al. Group: TBA
Filing Date: October 17, 1994 Page No: 3

AT Udell, J., "Visual Basic Custom Controls Meet OLE", Byte Magazine, March 1994, pp. 197-200.

AU Sarna, D.E., et al., "OLE Gains Without (Much) Pain", Datamation Magazine, June 15, 1994, pp. 31 and 113.

AV Rizzo, J., "What's OpenDoc?", MacUser Magazine, April 1994, pp.119-123.

AW Fogarty, K., et al., "Microsoft's OLE can be network Trojan Horse", Network World Magazine, June 27, 1994, Vol. 11, No. 26, pp. 1 and 75.

AX "Cello WWW Browser Release 1.01a", Article obtained from the Internet, ftp.law.cornell.edu/pub/L11/Cello no DDE, March 16, 1994, pp. 2-9.

AY "OLE 2.0: Death to Monoliths", Byte Magazine, March 1994, p. 122.

Examiner: D. DINH	Date Considered: 4/16/96

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP §609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

SUBSTITUTE DISCLOSURE STATEMENT FORM (PTO-1449)

#4

Form PTO 948 (Rev. 10-94) U.S. DEPARTMENT OF COMMERCE - Patent and Trademark Office Application No. 08/324443.

NOTICE OF DRAFTSPERSON'S PATENT DRAWING REVIEW

PTO Draftpersons review all originally filed drawings regardless of whether they are designated as formal or informal. Additionally, patent Examiners will review the drawings for compliance with the regulations. Direct telephone inquiries concerning this review to the Drawing Review Branch, 703-305-8404.

The drawings filed (insert date) 10/17/94, are
A. ___ not objected to by the Draftsperson under 37 CFR 1.84 or 1.152.
B. ✓ objected to by the Draftsperson under 37 CFR 1.84 or 1.152 as indicated below. The Examiner will require submission of new, corrected drawings when necessary. Corrected drawings must be submitted according to the instructions on the back of this Notice.

1. DRAWINGS. 37 CFR 1.84(a): Acceptable categories of drawings: Black ink. Color.
   ___ Not black solid lines. Fig(s)___
   ___ Color drawings are not acceptable until petition is granted. Fig(s)___
2. PHOTOGRAPHS. 37 CFR 1.84(b)
   ___ Photographs are not acceptable until petition is granted. Fig(s)___
   ___ Photographs not properly mounted (must use brystol board or photographic double-weight paper). Fig(s)___
   ___ Poor quality (half-tone). Fig(s)___
3. GRAPHIC FORMS. 37 CFR 1.84 (d)
   ___ Chemical or mathematical formula not labeled as separate figure. Fig(s)___
   ___ Group of waveforms not presented as a single figure, using common vertical axis with time extending along horizontal axis. Fig(s)___
   ___ Individuals waveform not identified with a separate letter designation adjacent to the vertical axis. Fig(s)___
4. TYPE OF PAPER. 37 CFR 1.84(c)
   ___ Paper not flexible, strong, white, smooth, nonshiny, and durable. Sheet(s)___
   ___ Erasures, alterations, overwritings, interlineations, cracks, creases, and folds copy machine marks not accepted. Fig(s)___
   ___ Mylar, velum paper is not acceptable (too thin). Fig(s)___
5. SIZE OF PAPER. 37 CFR 1.84(f): Acceptable sizes:
   21.6 cm. by 35.6 cm. (8 l/2 by 14 inches)
   21.6 cm. by 33.1 cm. (8 l/2 by 13 inches)
   21.6 cm. by 27.9 cm. (8 l/2 by 11 inches)
   21.0 cm. by 29.7 cm. (DIN size A4)
   ___ All drawing sheets not the same size. Sheet(s)___
   ___ Drawing sheet not an acceptable size. Sheet(s)___
6. MARGINS. 37 CFR 1.84(g): Acceptable margins:

Paper size

	21.6 cm. X 35.6 cm. (8 l/2 X 14 inches)	21.6 cm X 33.1 cm. (8 l/2 X 13 inches)	21.6 cm. X 27.9 cm. (8 1/2 X 11 inches)	21.0 cm. X 29.7 cm. (DIN Size A4)
T	5.1 cm. (2")	2.5 cm. (1")	2.5 cm. (1")	2.5cm.
L	.64 cm. (1/4")	.64 cm. (1/4")	.64 cm. (1/4")	2.5 cm.
R	.64 cm. (1/4")	.64 cm. (1/4")	.64 cm. (1/4")	1.5 cm.
B	.64 cm. (1/4")	.64 cm. (1/4")	.64 cm. (1/4")	1.0 cm.

   Margins do not conform to chart above.
   Fig ~~Sheet~~(s) 1,2,7,6,9
   ✓ Top (T) ✓ Left (L) ___ Right (R) ___ Bottom (B)
7. VIEWS. 37 CFR 1.84(h)
   REMINDER: Specification may require revision to correspond to drawing changes.
   ___ All views not grouped together. Fig(s)___
   ___ Views connected by projection lines or lead lines. Fig(s)___
   Partial views. 37 CFR 1.84(h) 2
   ___ View and enlarged view not labled separatly or properly. Fig(s)___
   Sectional views. 37 CFR 1.84 (h) 3
   ___ Hatching not indicated for sectional portions of an object. Fig(s)___
   ___ Cross section not drawn same as view with parts in cross section with regularly spaced parallel oblique strokes. Fig(s)___
8. ARRANGEMENT OF VIEWS. 37 CFR 1.84(i)
   ___ Words do not appear on a horizontal, left-to-right fashion when page is either upright or turned so that the top becomes the right side, except for graphs. Fig(s)___
9. SCALE. 37 CFR 1.84(k)
   ___ Scale not large enough to show mechanism with crowding when drawing is reduced in size to two-thirds in reproduction. Fig(s)___
   ___ Indication such as "actual size" or scale 1/2" not permitted. Fig(s)___
10. CHARACTER OF LINES, NUMBERS, & LETTERS. 37 CFR 1.84(l)
   ✓ Lines, numbers & letters not uniformly thick and well defined, clean, durable, and black (except for color drawings). Fig(s) 1–10
11. SHADING. 37 CFR 1.84(m)
   ___ Solid black shading areas not permitted. Fig(s)___
   ___ Shade lines, pale, rough and blurred. Fig(s)___
12. NUMBERS, LETTERS, & REFERENCE CHARACTERS. 37 CFR 1.84(p)
   ✓ Numbers and reference characters not plain and legible. 37 CFR 1.84(p)(l) Fig(s) 1–10.
   ___ Numbers and reference characters not oriented in same direction as the view. 37 CFR 1.84(p)(l) Fig(s)___
   ___ English alphabet not used. 37 CFR 1.84(p)(2) Fig(s)___
   ✓ Numbers, letters, and reference characters do not measure at least .32 cm. (1/8 inch) in height. 37 CFR(p)(3) Fig(s) 1–10.
13. LEAD LINES. 37 CFR 1.84(q)
   ___ Lead lines cross each other. Fig(s)___
   ___ Lead lines missing. Fig(s)___
14. NUMBERING OF SHEETS OF DRAWINGS. 37 CFR 1.84(t)
   ___ Sheets not numbered consecutively, and in Arabic numerals, beginning with number 1. Sheet(s)___
15. NUMBER OF VIEWS. 37 CFR 1.84(u)
   ___ Views not numbered consecutively, and in Arabic numerals, beginning with number 1. Fig(s)___
   ___ View numbers not preceded by the abbreviation Fig. Fig(s)___
16. CORRECTIONS. 37 CFR 1.84(w)
   ___ Corrections not made from prior PTO-948. Fig(s)___
17. DESIGN DRAWING. 37 CFR 1.152
   ___ Surface shading shown not appropriate. Fig(s)___
   ___ Solid black shading not used for color contrast. Fig(s)___

COMMENTS:

ATTACHMENT TO PAPER NO.___ REVIEWER Tiny DATE 2/27/95

PTO Copy

906 PH Ex. 3

GB
2317

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231,

8-6-96

TOWNSEND and TOWNSEND and CREW LLP

By dl Bullont

PATENT

Attorney Docket No. 02307I-553

#5/A
BH
10-9-96

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

MICHAEL D. DOYLE et al.

Application No.: 08/324,443

Filed: 10/17/94

For: EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

Examiner: D. Dinh

Art Unit: 2317

AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Responsive to the Office Action mailed May 6, 1996, please amend the above identified application as follows:

IN THE CLAIMS: ✓

Please cancel claim 16.

✓ Please amend the following claims:

1. (Amended) A method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment;

executing, at said client workstation, a browser application, that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processes specified by the text format;

utilizing said browser to display[ing], on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein said first hypermedia document is displayed within a first browser-controlled window on said client workstation and wherein said first distributed hypermedia document includes an embed text format that specifies the location of an object external to the first distributed hypermedia document and that specifies type information utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document;

invoking, with said browser application, said executable application to display and process said object within the first browser-controlled window while a portion of said first distributed hypermedia document continues to be displayed within said browser-controlled window [an embedded controllable application; and

interactively controlling said embedded controllable application from said client workstation via communications sent over said distributed hypermedia environment].

2. (Amended) The method of claim 1, wherein said executable application is a controllable application and further comprising the step of:

interactively controlling said controllable application from said client workstation via communications sent over said distributed hypermedia environment [displaying is performed by using a hypermedia browser application].

3. (Amended) The method of claim 2, wherein instructions for controlling said [embedded] controllable application reside on said network server, wherein said step of interactively controlling said [embedded] controllable application includes the following substeps:

issuing, from the client workstation, one or more commands to the network server;

executing, on the network server, one or more instructions in response to said commands;

sending information from said network server to said client workstation in response to said executed instructions; and

processing said information at the client workstation to interactively control said [embedded] controllable application.

4. (Amended) The method of claim 2, wherein instructions for controlling said [embedded] controllable application reside on said client workstation.

5. (Amended The method of claim 2, wherein the communications to interactively control said [embedded] controllable application from said client workstation continue to be exchanged between the controllable application and the hypermedia browser even after the controllable application program has been launched.

6. (Amended) The method of claim 3, wherein said [embedded] controllable application is a multi-dimensional viewer.

7. (Amended) The method of claim 3, wherein said [embedded] controllable application is a spreadsheet program.

8. (Amended) The method of claim 3, wherein said [embedded] controllable application is a database program.

9. (Amended) The method of claim 3, wherein said [embedded] controllable application is a word processor.

14. (Amended) The method of claim 13, wherein HyperText Markup Language is used to specify said [embedded] controllable application within said hypermedia document.

15. (Amended) A method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, said network including a plurality of general purpose workstations, wherein said network environment is a distributed hypermedia environment;

executing, at said client workstation, a browser application, that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processes specified by the text format;

utilizing said browser to display[ing], on said client workstation, at least a portion of a hypermedia document received over said network from said server, wherein said hypermedia document is displayed within a browser controlled window and includes at least a first [embedded] text format that identifies and locates a multi-dimensional data visualization application external to the distributed hypermedia document;

invoking, with said browser application, said multi-dimensional data visualization aplication; and

interactively controlling said [embedded] multi-dimensional data visualization application from within said browser window at said client workstation via communications sent over said distributed hypermedia environment wherein data image rendering is performed by said plurality of general purpose workstations using distributed processing.

23. (Amended) The method of claim 15, wherein a hypermedia browser is executing on the client workstation, wherein communications to interactively control said [embedded] controllable application from said client workstation continue to be exchanged between the controllable application and the hypermedia browser even after the controllable application program has been launched.

24. (Amended) A computer controlled method for interactively controlling an [embedded] external object referred to in a document displayed in a document window on a client computer, wherein the client computer includes a processor coupled to a display device and to a user input device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a document, wherein the document includes a reference to an [embedded] external object, wherein an application program for manipulating the [embedded] object resides on a first additional computer, the method comprising the following steps:

transferring, over the network, at least a portion of the document from the server computer to the client computer;

parsing said document to locate a reference to the external object included in the document, with the reference identifying and locating the external object;

accepting first signals from the user input device that indicate that the [embedded] external object is to be manipulated;

issuing commands from the client computer to the first additional computer in response to the first signals;

executing, by using the first additional computer, instructions in the application program in response to the issued commands, wherein the executed instructions generate information about manipulating the [embedded] external object;

communicating, via the network, the information about manipulating the [embedded] external object from the first additional computer to the client computer;

displaying communicated information in said document window at the client computer; and

using the client computer to manipulate the [embedded] object according to the communicated information displayed.

26. (Amended) The method of claim 24, further comprising the steps of executing instructions in a second application program on a second additional computer in response to the issued commands, wherein the instructions executed by the second additional computer result in information about manipulating the [embedded] object being generated more quickly.

28. (Amended) The method of claim 26, wherein the [embedded] object is a multi-dimensional image displayable in any of a plurality of orientations.

34. (Amended) A method for displaying a three dimensional image object on a client computer, wherein the client computer includes a processor coupled to a display device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a hypermedia document, wherein the hypermedia document includes a reference indicating the identity and location of a three dimensional image object [embedded within] external to the hypermedia document, wherein the three dimensional image object is displayable in a plurality of orientations, the method comprising the following steps:

transferring, over the network, at least a portion of the hypermedia document from the server computer to the client computer;

displaying in a document window on the display device, by using the processor, at least a portion of the hypermedia document, wherein the displayed portion of the hypermedia document includes the three dimensional image object displayed in a first orientation;

using the client computer to issue commands over the network;

executing instruction on a first additional computer in response to the issued commands, wherein the executed instructions determine a second orientation for display of the three dimensional image object;

communicating, via the network, information about the second orientation from the first additional computer to the client computer; and

using the client computer to redisplay the three dimensional image object in the second orientation.

Please add the following claims:

44. (New) A computer program product for use in a system having at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment, the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein for causing a client workstation to invoke an external executable application referenced by a hypermedia document to display and process an external object referenced by the hypermedia document, said computer program product further comprising:

> computer readable program code for causing said client workstation to execute a browser application to parse a distributed hypermedia document to identify text formats included in the distributed hypermedia document and to respond to predetermined text formats to initiate processes specified by the text format;
>
> computer readable program code for causing said client workstation to utilize said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein said first hypermedia document is displayed within a first browser-controlled window on said client workstation and wherein said first distributed hypermedia document includes an embed text format that specifies the location of an object external to the first distributed hypermedia document and that specifies type information utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document;

computer readable program code for causing said client workstation to invoke, with said browser application, said executable application to display and process said object within the first browser-controlled window while a portion of said first distributed hypermedia document continues to be displayed within said browser-controlled window.

45. (New) The computer program product of claim 44, wherein said executable application is a controllable application and further comprising:

computer readable program code for causing said client workstation to interactively control said controllable application from said client workstation via communications sent over said distributed hypermedia environment.

46. (New) The computer program product of claim 45, wherein instructions for controlling said controllable application reside on said network server, wherein said step of interactively controlling said controllable application includes:

computer readable program code for causing said client workstation to issue, from the client workstation, one or more commands to the network server;

computer readable program code for causing said network server to execute one or more instructions in response to said commands;

computer readable program code for causing said network sever to send information to said client workstation in response to said executed instructions; and

computer readable program code for causing said client workstation to process said information at the client workstation to interactively control said controllable application.

47. (New) The computer program product of claim 45, wherein instructions for controlling said controllable application reside on said client workstation.

48. (New) The computer program product of claim 45, wherein the communications to interactively control said controllable application from said client workstation continue to be exchanged between the controllable application and the hypermedia browser even after the controllable application program has been launched.

49. (New) The computer program product of claim 46, wherein said controllable application is a multi-dimensional viewer.

50. (New) The computer program product of claim 46, wherein said controllable application is a spreadsheet program.

51. (New) The computer program product of claim 46, wherein said controllable application is a database program.

52. (New) The computer program product of claim 46, wherein said controllable application is a word processor.

53. (New) The computer program product method of claim 52, wherein HyperText Markup Language is used to specify said controllable application within said hypermedia document.

54. (New) A computer program product for use in a system having at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment, the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein for running an application program in a network environment being a distributed hypermedia environment to process an external object referenced by the hypermedia document, said computer program product further comprising:

computer readable program code for causing said client work station to execute a browser application, that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and to respond to predetermined text formats to initiate processes specified by the text format;

computer readable program code for causing said client work station to utilize said browser to display at least a portion of a hypermedia document received over said network from said server, wherein said hypermedia document is displayed within a browser controlled window and includes at least a first text format that identifies and locates a multi-dimensional data visualization application external to the distributed hypermedia document;

computer readable program code for causing said client work station to invoke, through said browser application, said multi-dimensional data visualization aplication; and

computer readable program code for causing said client work station to interactively control said multi-dimensional data visualization application from within said browser window at said client workstation via communications sent over said distributed hypermedia environment wherein data image rendering is performed by said plurality of general purpose workstations using distributed processing.

55. (New) A computer program product for use in a system including a client computer which includes a processor coupled to a display device and to a user input device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a hypermedia document, wherein the hypermedia document includes a reference to an external object, wherein an application program for manipulating the

object resides on a first additional computer, the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein for interactively controlling an external object referred to in a hypermedia document displayed in a document window on a client computer, said computer program product further comprising:

computer readable program code for causing said network to transfer at least a portion of the hypermedia document from the server computer to the client computer;

computer readable program code for causing said client work station to parse said hypermedia document to locate a reference to the external object included in the hypermedia document, with the reference identifying and locating the external object;

computer readable program code for causing said client work station to accept first signals from the user input device that indicate that the external object is to be manipulated;

computer readable program code for causing said client work station to issue commands from the client computer to the first additional computer in response to the first signals;

computer readable program code for causing said first additional computer to execute instructions in the application program in response to the issued commands, wherein the executed instructions generate information about manipulating the external object;

computer readable program code for causing said first additional computer to communicate, via the network, the information about manipulating the external object from the first additional computer to the client computer;

computer readable program code for causing said client work station to display communicated information in said document window at the client computer; and

computer readable program code for causing said client work station to manipulate the object according to the communicated information displayed.

56. (New) A computer program product for use in a system including a client computer which includes a processor coupled to a display device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a hypermedia document, wherein the hypermedia document includes a reference indicating the identity and location of a three dimensional image object external to the hypermedia document, wherein the three dimensional image object is displayable in a plurality of orientations, the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein for displaying a three dimensional image external object referred to in a document displayed in a document window on a client computer, said computer program product further comprising:

computer readable program code for causing said network to transfer at least a portion of the hypermedia document from the server computer to the client computer;

computer readable program code for causing said client workstation to display, in a document window on the display device, by using the processor, at least a portion of the hypermedia document, wherein the displayed portion of the hypermedia document includes the three dimensional image object displayed in a first orientation;

computer readable program code for using the client computer to issue commands over the network;

computer readable program code for executing instruction on a first additional computer in response to the issued commands, wherein the executed instructions determine a second orientation for display of the three dimensional image object;

computer readable program code for communicating, via the network, information about the second orientation from the first additional computer to the client computer; and

computer readable program code for using the client computer to redisplay the three dimensional image object in the second orientation.

REMARKS

Claims 1-43 have been examined, claims 1-9, 14, 15, 23, 24, 26, 28 and 34 are amended, claim 16 is cancelled, and claims 44-56 have been added. Accordingly, claims 1-15 and 17-56 are now pending in the application. Reexamination and reconsideration are requested.

Claims 1-4 and 15-16 are rejected under 35 U.S.C. Sec. 102(a) as being anticipated by the University of Southern California's Mercury Project (in public use from Sept. 15, 1994 to March 31, 1995 at http://www.usc.edu/dept/raiders/).

The present invention, as defined for example in amended claim 1, includes the steps of executing, at the client workstation, a browser application that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to text formats to initiate processes specified by that text format. The browser is also utilized to display at least a portion of the distributed hypermedia document within a browser-controlled window.

The distributed hypermedia document includes an embed text format the specifies the location of an object external to the distributed hypermedia document and that specifies type information utilized by the browser to identify and locate an executable application external to the distributed hypermedia document. The browser invokes the executable application to display and process the object within the browser window.

The Mercury Project is an interactive Web page that utilizes CGI (Common Gateway Interface) scripts and the HTML <FORM> tag to facilitate interaction between the user and the Web page.

CGI is a protocol that allows a Web server and an external program to communicate. An example is a form which utilizes the following HTML format:

```
<FORM
ACTION="identity of program to invoke on server"
METHOD="specify method of reading info"
>.
<INPUT
TYPE="type"
NAME="name"
SIZE="size"
>


etc.
</FORM>
```

Information is entered by the user and passed through the Web server to the CGI server identified by the ACTION= attribute. The CGI program generates a new result Web page which is sent back to the browser and displayed in place of the original page displayed in the browser-window.

For example, at the bottom of page 2 of the cited document it is stated that after clicking on the map the arm will move and a new image will be returned.

The HTML of the page 2 of the cited Mercury Project document is appended to this amendment. The page has been decommissioned so the <FORM> tags have been removed. However, it is apparent that all images are in-line GIF and MPEG images identified by their extensions. Also, appended is a page linked to the Mercury page, the Telegarden, page that utilizes the <FORM> tag to execute control.

The <FORM> tag causes the browser to send a string of characters, entered into a form in the original HTML document, to the Web server application. The Web server invokes a CGI application identified by the ACTION= attribute of the tag and passes the string to the CGI application. In response to the character string the server application generates and sends a new HTML document to be displayed by the browser in place of the ordinal HTML document.

For example, in the Mercury Project when the arm is moved by clicking on an in-line image included in a first HTML document, displayed in a first browser-controlled window, a string is sent to the Web server with information on how far the arm is moved. The Web server invokes the CGI server which generates a new HTML document with an in-line image showing the arm in the new position and sends the new document to the browser to be displayed in place of the first HTML document. Some browsers would cache the first HTML document so that the first and new HTML documents could be viewed in different windows.

In the Telegarden page, a new HTML document including a revised image of the garden is sent by the server and displayed by the browser in place of the original document. Also, forms are very popular in search pages where a new HTML document is sent by the sever and displayed by the browser in place of the original document to show the results of the search.

The examiner states that the Mercury Project operated using a method substantially as claimed. This rejection is respectfully traversed for the following reasons.

There is no disclosure of the claimed step of utilizing a browser to display a first hypermedia document in a first window with the hypermedia document including a tag format specifying the location of an external object and an external executable application. As described above, the Mercury Project utilizes CGI where a <FORM> tag identifies a program on the server but not an external object. Additionally, the claimed step of invoking the executable object to display and process the object within the first browser-controlled window while a portion of the first hypermedia document is displayed is not disclosed.

In the Mercury Project a new HTML document is generated at the server and displayed by the browser in a new window.

The claimed combination is fundamentally different from the Mercury Project. In the claimed combination, the external object and executable object are embedded by reference in the HTML document and the object is displayed and processed within the same window where a portion of the original document is displayed. In the Mercury Project information is passed back to the server and a new document is generated and displayed. There is no display and processing the external object within the window in which a portion of the original document is displayed.

A major difference between CGI and the claimed combination is that in CGI there is no enforced continuity between the documents. The CGI responds to form information by generating new documents each being a static document independent of the previous document which generated the form string passed to the Web server. For example, in the Mercury Project separate, independent HTML documents are generated for each position of the arm. There is no disclosure of the claimed step of invoking the executable application to display and process said object within the window while a portion of the first distributed hypermedia document is displayed in the window.

Thus, unlike CGI, the claimed executable application does not generate a static HTML document to be displayed in place of the first document but displays and processes the object in a portion of the window.

Claims 1-43 are rejected under 35 U.S.C. Sec. 103 as being unpatentable over Applicant's disclosed prior art and further in view of Hansen "Enhancing Documents with Embedded Programs: How Ness Extends Insets in the Andrew ToolKit".

The Hansen reference discloses embedding an executable script in a document. The Ness script is a sequence of attribute specifications, i.e., declarations of global variables, global functions, and **extend** blocks. An extend block associates a set of contained attributes with some named object and has the following syntax:

extend <name>

```
                <attributes>
        end extend
```

where <name> must be a string constant giving the name of the associated object. (Page 25). One attribute is an event specification such a mouse click.

The operation of Ness scripts is illustrated in the "Extended Birthday Card" example at pages 30 and 31. Note that the object named is: **extend** "visible cake". Both the executable script and the object to be manipulated are within the document.

In view of the above, it is believed that the claims are not obvious over the disclosed prior art in view of Hansen. There is no disclosure in the references, singly or in combination, of displaying a hypermedia document in a first window including a text format specifying the location of an external object and identifying an external executable application or of invoking the external application to display and process the external object within the first window.

The system of Ness provides for interaction with an object embedded in a document by executing code embedded in the document. However, there is no teaching or suggestion of the claimed system of utilizing a browser to invoke an external application identified by an original document, being displayed by a browser within a first window, to display and process an external object within the first window.

In view of the above, claim 1 is deemed to be patentable over the cited references.

Claim 15 recites that the browser application invokes an external application which is a multi-dimensional visualization application. The multi-dimensional visualization application is controlled from within the browser window via communications sent over the distributed hypermedia environment.

As described in the application, a 3-D image, located external to the original document, may be displayed within the first window and processed by an executable application external to the browser.

Thus, unlike the Mercury Project, a multi-dimensional object may be displayed and manipulated within the browser window by an application running on a remote computer. Also, Ness is distinguished because in the Ness system the application code is included in the hypermedia document. Thus, the combination claimed in claim 15 is a significant advance over systems such as disclosed in Hanson or the Mercury Project and is not suggested by any of the cited references.

Claim 24 recites accepting first signals from a user input device indicating that the external object, referenced in a hypermedia document displayed in a document window on a client computer, is to be manipulated. Commands are issued to a first additional computer in response to the first signal and the first additional computer responds to the commands by executing instructions in the application program to generate information about manipulating the external object. The information is communicated to the client computer which displays the information in the document window and is used to manipulate the object according to the information displayed.

As described above, in the Mercury Project an object in a document window can not be manipulated. The user can indicate a desired change to the object, e.g., move the arm, which is communicated to the Web server and transferred to the CGI server. The CGI server then generates a new document with an image showing the arm moved and transfers the new document back to the client which displays the new document in place of the original document. Thus, the Mercury Project does not provide for interaction with an object displayed in browser window. Additionally, as discussed in the specification, the requirement of sending a new hypermedia document to display each image change consumes bandwidth on the network.

The Ness disclosure does not teach using an application on a first additional computer to manipulate an object external to the hypermedia document within document window. In Ness both the object and application are included within the document displayed. While this feature enhances interactivity with documents, the application is executed on the computer displaying the document.

In contrast, in the combination of claim 24 the first computer could be much more powerful than the client computer to run an application performing, for example, 3-D visualization or CAD/CAM programs, and the results could be displayed in the document window of client computer in the form of a PC or network computer.

Accordingly, the subject matter of claim 24 is not disclosed or suggested by the cited references, singly or in combination.

In claim 24, the hypermedia document displayed in a document window on the client computer references an external object which is a three dimensional object to be displayed in a plurality of orientations. The external object is displayed in a first orientation within the document window. Commands are issued to a first additional computer in response to the first signal and the first additional computer responds to the commands by execute instructions in the application program to generate information about a second orientation of the external object. The information is communicated to the client computer which redisplays the external object in the second orientation within the document window.

The subject matter of claim 34 differs from the cited references for reasons similar to those discussed above with reference to claim 24. Again, the claimed system allows for a much more powerful first additional computer to perform calculations to reorient an object with the results communicated to the client computer to redisplay the object in a new orientation.

The claims added herein are of similar scope to the examined claims but are recited in the form of computer readable code, embodied on a computer readable medium, for causing a computer to effect the novel steps recited in the method claims previously examined.

In view of the foregoing, Applicants believe all claims now pending in this application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at (415) 576-0200.

Respectfully submitted,

Michael E. Woods

Charles E. Krueger
Reg. No. 30,077

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
(415) 576-0200
Fax (415) 576-0300
CEK:db
s:\02307I\553\amend.01

Amendment

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, CA 94111-3834
(415) 576-0200

Atty. Docket No. 02307I-055300US

Date August 6, 1996

In re application of MICHAEL D. DOYLE et al.

Appln. No. 08/324,443

Filed 10/17/94

Group Art Unit 2317
For EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

I hereby certify that this is being deposited with the United States Postal Service as first class mail in an envelope addressed to:

Assistant Commissioner for Patents
Washington, D. C. 20231.

Date: 8-6-96

VCBullock

THE ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Sir:

Transmitted herewith is an amendment in the above-identified application.

[ ] Enclosed is a petition to extend time to respond.

[X] Small entity status of this application under 37 CFR 1.9 and 1.27 has been established by a verified statement previously submitted.

[ ] A verified statement to establish small entity status under 37 CFR 1.9 and 1.27 is enclosed.

[X] A document entitled *USC Mercury Project: Interface*

If any extension of time is needed, then this response should be considered a petition therefor.

The filing fee has been calculated as shown below:

	(Col. 1)		(Col. 2)	(Col. 3)	SMALL ENTITY			OTHER THAN A SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NO. PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDIT. FEE	OR	RATE	ADDIT. FEE
TOTAL	56*	MINUS	43**	= 13	x11=	$143		x22=	$
INDEP.	8*	MINUS	4***	= 4	x39=	$156		x78=	$
[ ] FIRST PRESENTATION OF MULTIPLE DEP. CLAIM					+125=	$		+250=	$
					TOTAL ADDIT. FEE	$299	OR	TOTAL	$

* If the entry in Col. 1 is less than the entry in Col. 2, write "0" in Col. 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, write "20" in this space.

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, write "3" in this space.
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found from the equivalent box in Col. 1 of a prior amendment or the number of claims originally filed.

[ ] No fee is due.

Please charge Deposit Account No. 20-1430 as follows:

[x] Claims fee $ 299.00

[X] Any additional fees associated with this paper or during the pendency of this application.

TWO extra copies of this sheet are enclosed.

TOWNSEND and TOWNSEND and CREW LLP

Michael E. Woods

Michael E. Woods / Reg. No.: 33,466
Attorneys for Applicant

Law Offices
TOWNSEND and TOWNSEND and CREW LLP
Patents, Trademarks, Copyrights, Antitrust, Commercial Litigation

TWO EMBARCADERO CENTER, 8TH FLOOR
SAN FRANCISCO, CALIFORNIA 94111-3834
TEL (415) 576-0200 FAX (415) 576-0300

Copy of #5

TELECOPIER COVER SHEET

IMPORTANT: This message is intended only for the use of the individual or entity to which it is addressed and may contain information that is privileged, confidential, and/or exempt from disclosure by applicable law or court order. If the reader of this message is not the intended recipient, or the employee or agent responsible for delivering the message to the intended recipient, you are hereby notified that any dissemination, distribution, or copying of this communication is **strictly prohibited**. If you have received this communication in error, please notify us immediately by telephone and return the original message to us at the above San Francisco address via the United States Postal Service. Thank you.

Phone: (415) 576-0200, FAX: (415) 576-0300

September 10, 1996

To: Examiner Dinh, Group 2317
From: Charles E. Krueger
Client Number: 023071-553 At FAX Number: **703/308-5359**
Number of Pages (including this page): 29
If you have any problems with reception, please call **Irene Rodas** at extension 4659, or **Deborah Bullock** at extension 4557.

Re: Appln. No. 08/324,443 / Doyle et al.

Dear Examiner Dinh:

Attached hereto is a copy of an amendment in the referenced application, as mailed to the Patent Office on August 6, 1996. As discussed with my secretary, please let us know if for any reason this amendment will not be considered as timely filed.

Very truly yours,

Charles E. Krueger

CEK:db
Encl.

08/324443
23B1

FAXED______________

RETURN TO______________

Amendment

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, CA 94111-3834
(415) 576-0200

Atty. Docket No. 023071-055300US

Date August 6, 1996

In re application of MICHAEL D. DOYLE et al.

Appln. No. 08/324,443

Filed 10/17/94

Group Art Unit 2317
For EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

I hereby certify that this is being deposited with the United States Postal Service as first class mail in an envelope addressed to:

Assistant Commissioner for Patents
Washington, D. C. 20231.

Date: 8-6-96

THE ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Sir:

Transmitted herewith is an amendment in the above-identified application.

[ ] Enclosed is a petition to extend time to respond.

[X] Small entity status of this application under 37 CFR 1.9 and 1.27 has been established by a verified statement previously submitted.

[ ] A verified statement to establish small entity status under 37 CFR 1.9 and 1.27 is enclosed.

[X] A document entitled *USC Mercury Project: Interface*

If any extension of time is needed, then this response should be considered a petition therefor.

The filing fee has been calculated as shown below:

	(Col. 1)		(Col. 2)	(Col. 3)	SMALL ENTITY			OTHER THAN A SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NO. PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDIT. FEE	OR	RATE	ADDIT. FEE
TOTAL	56*	MINUS	43**	= 13	x11=	$143		x22=	$
INDEP.	8*	MINUS	4***	= 4	x39=	$156		x78=	$
[ ] FIRST PRESENTATION OF MULTIPLE DEP. CLAIM					+125=	$		+250=	$
					TOTAL ADDIT. FEE	$299	OR	TOTAL	$

* If the entry in Col. 1 is less than the entry in Col. 2, write "0" in Col. 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, write "20" in this space.

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, write "3" in this space.
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found from the equivalent box in Col. 1 of a prior amendment or the number of claims originally filed.

[ ] No fee is due.

Please charge Deposit Account No. 20-1430 as follows:

[x] Claims fee $ 299.00

[X] Any additional fees associated with this paper or during the pendency of this application.

TWO extra copies of this sheet are enclosed.

TOWNSEND and TOWNSEND and CREW LLP

Michael E. Woods

Michael E. Woods / Reg. No.: 33,466
Attorneys for Applicant

AMEND.TRN 4/96

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231, on 8-6-96

TOWNSEND and TOWNSEND and CREW LLP

By

PATENT

Attorney Docket No. 023071-553

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

MICHAEL D. DOYLE et al.

Application No.: 08/324,443

Filed: 10/17/94

For: EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

Examiner: D. Dinh

Art Unit: 2317

AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Responsive to the Office Action mailed May 6, 1996, please amend the above identified application as follows:

IN THE CLAIMS:

Please cancel claim 16.

Please amend the following claims:

1. (Amended) A method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment;

executing, at said client workstation, a browser application, that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processes specified by the text format;

utilizing said browser to display[ing], on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein said first hypermedia document is displayed within a first browser-controlled window on said client workstation and wherein said first distributed hypermedia document includes an embed text format that specifies the location of an object external to the first distributed hypermedia document and that specifies type information utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document;

invoking, with said browser application, said executable application to display and process said object within the first browser-controlled window while a portion of said first distributed hypermedia document continues to be displayed within said browser-controlled window [an embedded controllable application; and

interactively controlling said embedded controllable application from said client workstation via communications sent over said distributed hypermedia environment].

2. (Amended) The method of claim 1, wherein said executable application is a controllable application and further comprising the step of:

interactively controlling said controllable application from said client workstation via communications sent over said distributed hypermedia environment [displaying is performed by using a hypermedia browser application].

3. (Amended) The method of claim 2, wherein instructions for controlling said [embedded] controllable application reside on said network server, wherein said step of interactively controlling said [embedded] controllable application includes the following substeps:

issuing, from the client workstation, one or more commands to the network server;

executing, on the network server, one or more instructions in response to said commands;

sending information from said network server to said client workstation in response to said executed instructions; and

processing said information at the client workstation to interactively control said [embedded] controllable application.

4. (Amended) The method of claim 2, wherein instructions for controlling said [embedded] controllable application reside on said client workstation.

5. (Amended The method of claim 2, wherein the communications to interactively control said [embedded] controllable application from said client workstation continue to be exchanged between the controllable application and the hypermedia browser even after the controllable application program has been launched.

6. (Amended) The method of claim 3, wherein said [embedded] controllable application is a multi-dimensional viewer.

7. (Amended) The method of claim 3, wherein said [embedded] controllable application is a spreadsheet program.

8. (Amended) The method of claim 3, wherein said [embedded] controllable application is a database program.

9. (Amended) The method of claim 3, wherein said [embedded] controllable application is a word processor.

14. (Amended) The method of claim 13, wherein HyperText Markup Language is used to specify said [embedded] controllable application within said hypermedia document.

15. (Amended) A method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, said network including a plurality of general purpose workstations, wherein said network environment is a distributed hypermedia environment;

executing, at said client workstation, a browser application, that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processes specified by the text format;

utilizing said browser to display[ing], on said client workstation, at least a portion of a hypermedia document received over said network from said server, wherein said hypermedia document is displayed within a browser controlled window and includes at least a first [embedded] text format that identifies and locates a multi-dimensional data visualization application external to the distributed hypermedia document;

invoking, with said browser application, said multi-dimensional data visualization aplication; and

interactively controlling said [embedded] multi-dimensional data visualization application from within said browser window at said client workstation via communications sent over said distributed hypermedia environment wherein data image rendering is performed by said plurality of general purpose workstations using distributed processing.

23. (Amended) The method of claim 15, wherein a hypermedia browser is executing on the client workstation, wherein communications to interactively control said [embedded] controllable application from said client workstation continue to be exchanged between the controllable application and the hypermedia browser even after the controllable application program has been launched.

24. (Amended) A computer controlled method for interactively controlling an [embedded] external object referred to in a document displayed in a document window on a client computer, wherein the client computer includes a processor coupled to a display device and to a user input device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a document, wherein the document includes a reference to an [embedded] external object, wherein an application program for manipulating the [embedded] object resides on a first additional computer, the method comprising the following steps:

transferring, over the network, at least a portion of the document from the server computer to the client computer;

parsing said document to locate a reference to the external object included in the document, with the reference identifying and locating the external object;

accepting first signals from the user input device that indicate that the [embedded] external object is to be manipulated;

issuing commands from the client computer to the first additional computer in response to the first signals;

executing, by using the first additional computer, instructions in the application program in response to the issued commands, wherein the executed instructions generate information about manipulating the [embedded] external object;

communicating, via the network, the information about manipulating the [embedded] external object from the first additional computer to the client computer;

displaying communicated information in said document window at the client computer; and

using the client computer to manipulate the [embedded] object according to the communicated information displayed.

26. (Amended) The method of claim 24, further comprising the steps of executing instructions in a second application program on a second additional computer in response to the issued commands, wherein the instructions executed by the second additional computer result in information about manipulating the [embedded] object being generated more quickly.

28. (Amended) The method of claim 26, wherein the [embedded] object is a multi-dimensional image displayable in any of a plurality of orientations.

34. (Amended) A method for displaying a three dimensional image object on a client computer, wherein the client computer includes a processor coupled to a display device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a hypermedia document, wherein the hypermedia document includes a reference indicating the identity and location of a three dimensional image object [embedded within] external to the hypermedia document, wherein the three dimensional image object is displayable in a plurality of orientations, the method comprising the following steps:

transferring, over the network, at least a portion of the hypermedia document from the server computer to the client computer;

displaying in a document window on the display device, by using the processor, at least a portion of the hypermedia document, wherein the displayed portion of the hypermedia document includes the three dimensional image object displayed in a first orientation;

using the client computer to issue commands over the network;

executing instruction on a first additional computer in response to the issued commands, wherein the executed instructions determine a second orientation for display of the three dimensional image object;

communicating, via the network, information about the second orientation from the first additional computer to the client computer; and

using the client computer to redisplay the three dimensional image object in the second orientation.

Please add the following claims:

-- 44. (New) A computer program product for use in a system having at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment, the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein for causing a client workstation to invoke an external executable application referenced by a hypermedia document to display and process an external object referenced by the hypermedia document, said computer program product further comprising:

computer readable program code for causing said client workstation to execute a browser application to parse a distributed hypermedia document to identify text formats included in the distributed hypermedia document and to respond to predetermined text formats to initiate processes specified by the text format;

computer readable program code for causing said client workstation to utilize said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein said first hypermedia document is displayed within a first browser-controlled window on said client workstation and wherein said first distributed hypermedia document includes an embed text format that specifies the location of an object external to the first distributed hypermedia document and that specifies type information utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document;

computer readable program code for causing said client workstation to invoke, with said browser application, said executable application to display and process said object within the first browser-controlled window while a portion of said first distributed hypermedia document continues to be displayed within said browser-controlled window.

45. (New) The computer program product of claim 44, wherein said executable application is a controllable application and further comprising:

computer readable program code for causing said client workstation to interactively control said controllable application from said client workstation via communications sent over said distributed hypermedia environment.

46. (New) The computer program product of claim 45, wherein instructions for controlling said controllable application reside on said network server, wherein said step of interactively controlling said controllable application includes:

computer readable program code for causing said client workstation to issue, from the client workstation, one or more commands to the network server;

computer readable program code for causing said network server to execute one or more instructions in response to said commands;

computer readable program code for causing said network sever to send information to said client workstation in response to said executed instructions; and

computer readable program code for causing said client workstation to process said information at the client workstation to interactively control said controllable application.

47. (New) The computer program product of claim 45, wherein instructions for controlling said controllable application reside on said client workstation.

48. (New) The computer program product of claim 45, wherein the communications to interactively control said controllable application from said client workstation continue to be exchanged between the controllable application and the hypermedia browser even after the controllable application program has been launched.

49. (New) The computer program product of claim 46, wherein said controllable application is a multi-dimensional viewer.

50. (New) The computer program product of claim 46, wherein said controllable application is a spreadsheet program.

51. (New) The computer program product of claim 46, wherein said controllable application is a database program.

52. (New) The computer program product of claim 46, wherein said controllable application is a word processor.

53. (New) The computer program product method of claim 52, wherein HyperText Markup Language is used to specify said controllable application within said hypermedia document.

54. (New) A computer program product for use in a system having at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment, the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein for running an application program in a network environment being a distributed hypermedia environment to process an external object referenced by the hypermedia document, said computer program product further comprising:

computer readable program code for causing said client work station to execute a browser application, that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and to respond to predetermined text formats to initiate processes specified by the text format;

computer readable program code for causing said client work station to utilize said browser to display at least a portion of a hypermedia document received over said network from said server, wherein said hypermedia document is displayed within a browser controlled window and includes at least a first text format that identifies and locates a multi-dimensional data visualization application external to the distributed hypermedia document;

computer readable program code for causing said client work station to invoke, through said browser application, said multi-dimensional data visualization aplication; and

computer readable program code for causing said client work station to interactively control said multi-dimensional data visualization application from within said browser window at said client workstation via communications sent over said distributed hypermedia environment wherein data image rendering is performed by said plurality of general purpose workstations using distributed processing.

55. (New) A computer program product for use in a system including a client computer which includes a processor coupled to a display device and to a user input device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a hypermedia document, wherein the hypermedia document includes a reference to an external object, wherein an application program for manipulating the

object resides on a first additional computer, the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein for interactively controlling an external object referred to in a hypermedia document displayed in a document window on a client computer, said computer program product further comprising:

computer readable program code for causing said network to transfer at least a portion of the hypermedia document from the server computer to the client computer;

computer readable program code for causing said client work station to parse said hypermedia document to locate a reference to the external object included in the hypermedia document, with the reference identifying and locating the external object;

computer readable program code for causing said client work station to accept first signals from the user input device that indicate that the external object is to be manipulated;

computer readable program code for causing said client work station to issue commands from the client computer to the first additional computer in response to the first signals;

computer readable program code for causing said first additional computer to execute instructions in the application program in response to the issued commands, wherein the executed instructions generate information about manipulating the external object;

computer readable program code for causing said first additional computer to communicate, via the network, the information about manipulating the external object from the first additional computer to the client computer;

computer readable program code for causing said client work station to display communicated

PATENT

information in said document window at the client computer; and

computer readable program code for causing said client work station to manipulate the object according to the communicated information displayed.

56. (New) A computer program product for use in a system including a client computer which includes a processor coupled to a display device, wherein the processor is further coupled to a computer network, wherein the computer network is coupled to a server computer and one or more additional computers, wherein the server computer includes a local storage device containing a hypermedia document, wherein the hypermedia document includes a reference indicating the identity and location of a three dimensional image object external to the hypermedia document, wherein the three dimensional image object is displayable in a plurality of orientations, the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein for displaying a three dimensional image external object referred to in a document displayed in a document window on a client computer, said computer program product further comprising:

computer readable program code for causing said network to transfer at least a portion of the hypermedia document from the server computer to the client computer;

computer readable program code for causing said client workstation to display, in a document window on the display device, by using the processor, at least a portion of the hypermedia document, wherein the displayed portion of the hypermedia document includes the three dimensional image object displayed in a first orientation;

computer readable program code for using the client computer to issue commands over the network;

computer readable program code for executing instruction on a first additional computer in response to the issued commands, wherein the executed instructions determine a second orientation for display of the three dimensional image object;

computer readable program code for communicating, via the network, information about the second orientation from the first additional computer to the client computer; and

computer readable program code for using the client computer to redisplay the three dimensional image object in the second orientation. --

REMARKS

Claims 1-43 have been examined, claims 1-9, 14, 15, 23, 24, 26, 28 and 34 are amended, claim 16 is cancelled, and claims 44-56 have been added. Accordingly, claims 1-15 and 17-56 are now pending in the application. Reexamination and reconsideration are requested.

Claims 1-4 and 15-16 are rejected under 35 U.S.C. Sec. 102(a) as being anticipated by the University of Southern California's Mercury Project (in public use from Sept. 15, 1994 to March 31, 1995 at http://www.usc.edu/dept/raiders/).

The present invention, as defined for example in amended claim 1, includes the steps of executing, at the client workstation, a browser application that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to text formats to initiate processes specified by that text format. The browser is also utilized to display at least a portion of the distributed hypermedia document within a browser-controlled window.

The distributed hypermedia document includes an embed text format the specifies the location of an object external to the distributed hypermedia document and that specifies type information utilized by the browser to identify and locate an executable application external to the distributed hypermedia document. The browser invokes the executable application to display and process the object within the browser window.

The Mercury Project is an interactive Web page that utilizes CGI (Common Gateway Interface) scripts and the HTML <FORM> tag to facilitate interaction between the user and the Web page.

CGI is a protocol that allows a Web server and an external program to communicate. An example is a form which utilizes the following HTML format:

```
<FORM
ACTION="identity of program to invoke on server"
METHOD="specify method of reading info"
>.
<INPUT
TYPE="type"
NAME="name"
SIZE="size"
>

etc.
</FORM>
```

Information is entered by the user and passed through the Web server to the CGI server identified by the ACTION= attribute. The CGI program generates a new result Web page which is sent back to the browser and displayed in place of the original page displayed in the browser-window.

For example, at the bottom of page 2 of the cited document it is stated that after clicking on the map the arm will move and a new image will be returned.

The HTML of the page 2 of the cited Mercury Project document is appended to this amendment. The page has been decommissioned so the <FORM> tags have been removed. However, it is apparent that all images are in-line GIF and MPEG images identified by their extensions. Also, appended is a page linked to the Mercury page, the Telegarden, page that utilizes the <FORM> tag to execute control.

The <FORM> tag causes the browser to send a string of characters, entered into a form in the original HTML document, to the Web server application. The Web server invokes a CGI application identified by the ACTION= attribute of the tag and passes the string to the CGI application. In response to the character string the server application generates and sends a new HTML document to be displayed by the browser in place of the ordinal HTML document.

For example, in the Mercury Project when the arm is moved by clicking on an in-line image included in a first HTML document, displayed in a first browser-controlled window, a string is sent to the Web server with information on how far the arm is moved. The Web server invokes the CGI server which generates a new HTML document with an in-line image showing the arm in the new position and sends the new document to the browser to be displayed in place of the first HTML document. Some browsers would cache the first HTML document so that the first and new HTML documents could be viewed in different windows.

In the Telegarden page, a new HTML document including a revised image of the garden is sent by the server and displayed by the browser in place of the original document. Also, forms are very popular in search pages where a new HTML document is sent by the sever and displayed by the browser in place of the original document to show the results of the search.

The examiner states that the Mercury Project operated using a method substantially as claimed. This rejection is respectfully traversed for the following reasons.

There is no disclosure of the claimed step of utilizing a browser to display a first hypermedia document in a first window with the hypermedia document including a tag format specifying the location of an external object and an external executable application. As described above, the Mercury Project utilizes CGI where a <FORM> tag identifies a program on the server but not an external object. Additionally, the claimed step of invoking the executable object to display and process the object within the first browser-controlled window while a portion of the first hypermedia document is displayed is not disclosed.

In the Mercury Project a new HTML document is generated at the server and displayed by the browser in a new window.

The claimed combination is fundamentally different from the Mercury Project. In the claimed combination, the external object and executable object are embedded by reference in the HTML document and the object is displayed and processed within the same window where a portion of the original document is displayed. In the Mercury Project information is passed back to the server and a new document is generated and displayed. There is no display and processing the external object within the window in which a portion of the original document is displayed.

A major difference between CGI and the claimed combination is that in CGI there is no enforced continuity between the documents. The CGI responds to form information by generating new documents each being a static document independent of the previous document which generated the form string passed to the Web server. For example, in the Mercury Project separate, independent HTML documents are generated for each position of the arm. There is no disclosure of the claimed step of invoking the executable application to display and process said object within the window while a portion of the first distributed hypermedia document is displayed in the window.

Thus, unlike CGI, the claimed executable application does not generate a static HTML document to be displayed in place of the first document but displays and processes the object in a portion of the window.

Claims 1-43 are rejected under 35 U.S.C. Sec. 103 as being unpatentable over Applicant's disclosed prior art and further in view of Hansen "Enhancing Documents with Embedded Programs: How Ness Extends Insets in the Andrew ToolKit".

The Hansen reference discloses embedding an executable script in a document. The Ness script is a sequence of attribute specifications, i.e., declarations of global variables, global functions, and **extend** blocks. An extend block associates a set of contained attributes with some named object and has the following syntax:

extend <name>

<attributes>

end extend

where <name> must be a string constant giving the name of the associated object. (Page 25). One attribute is an event specification such a mouse click.

The operation of Ness scripts is illustrated in the "Extended Birthday Card" example at pages 30 and 31. Note that the object named is: **extend** "visible cake". Both the executable script and the object to be manipulated are within the document.

In view of the above, it is believed that the claims are not obvious over the disclosed prior art in view of Hansen. There is no disclosure in the references, singly or in combination, of displaying a hypermedia document in a first window including a text format specifying the location of an external object and identifying an external executable application or of invoking the external application to display and process the external object within the first window.

The system of Ness provides for interaction with an object embedded in a document by executing code embedded in the document. However, there is no teaching or suggestion of the claimed system of utilizing a browser to invoke an external application identified by an original document, being displayed by a browser within a first window, to display and process an external object within the first window.

In view of the above, claim 1 is deemed to be patentable over the cited references.

Claim 15 recites that the browser application invokes an external application which is a multi-dimensional visualization application. The multi-dimensional visualization application is controlled from within the browser window via communications sent over the distributed hypermedia environment.

As described in the application, a 3-D image, located external to the original document, may be displayed within the first window and processed by an executable application external to the browser.

Thus, unlike the Mercury Project, a multi-dimensional object may be displayed and manipulated within the browser window by an application running on a remote computer. Also, Ness is distinguished because in the Ness system the application code is included in the hypermedia document. Thus, the combination claimed in claim 15 is a significant advance over systems such as disclosed in Hanson or the Mercury Project and is not suggested by any of the cited references.

Claim 24 recites accepting first signals from a user input device indicating that the external object, referenced in a hypermedia document displayed in a document window on a client computer, is to be manipulated. Commands are issued to a first additional computer in response to the first signal and the first additional computer responds to the commands by executing instructions in the application program to generate information about manipulating the external object. The information is communicated to the client computer which displays the information in the document window and is used to manipulate the object according to the information displayed.

As described above, in the Mercury Project an object in a document window can not be manipulated. The user can indicate a desired change to the object, e.g., move the arm, which is communicated to the Web server and transferred to the CGI server. The CGI server then generates a new document with an image showing the arm moved and transfers the new document back to the client which displays the new document in place of the original document. Thus, the Mercury Project does not provide for interaction with an object displayed in browser window. Additionally, as discussed in the specification, the requirement of sending a new hypermedia document to display each image change consumes bandwidth on the network.

The Ness disclosure does not teach using an application on a first additional computer to manipulate an object external to the hypermedia document within document window. In Ness both the object and application are included within the document displayed. While this feature enhances interactivity with documents, the application is executed on the computer displaying the document.

PATENT

In contrast, in the combination of claim 24 the first computer could be much more powerful than the client computer to run an application performing, for example, 3-D visualization or CAD/CAM programs, and the results could be displayed in the document window of client computer in the form of a PC or network computer.

Accordingly, the subject matter of claim 24 is not disclosed or suggested by the cited references, singly or in combination.

In claim 24, the hypermedia document displayed in a document window on the client computer references an external object which is a three dimensional object to be displayed in a plurality of orientations. The external object is displayed in a first orientation within the document window. Commands are issued to a first additional computer in response to the first signal and the first additional computer responds to the commands by execute instructions in the application program to generate information about a second orientation of the external object. The information is communicated to the client computer which redisplays the external object in the second orientation within the document window.

The subject matter of claim 34 differs from the cited references for reasons similar to those discussed above with reference to claim 24. Again, the claimed system allows for a much more powerful first additional computer to perform calculations to reorient an object with the results communicated to the client computer to redisplay the object in a new orientation.

The claims added herein are of similar scope to the examined claims but are recited in the form of computer readable code, embodied on a computer readable medium, for causing a computer to effect the novel steps recited in the method claims previously examined.

In view of the foregoing, Applicants believe all claims now pending in this application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

PATENT

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at (415) 576-0200.

Respectfully submitted,

Michael E. Woods

Charles E. Krueger
Reg. No. 30,077

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
(415) 576-0200
Fax (415) 576-0300
CEK:db
s:\02307I\553\amend.01

906 PH Ex. 4

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
08/324,443	10/17/94	DOYLE	M 02307553

B3M1/1213

TOWNSEND AND TOWNSEND
KHOURIE AND CREW
STEUART STREET TOWER
ONE MARKET PLAZA
SAN FRANCISCO CA 94105

EXAMINER	
ART UNIT	**PAPER NUMBER**
2317	6

DATE MAILED: 12/13/96

This is a communication from the examiner in charge of your application.
COMMISSIONER OF PATENTS AND TRADEMARKS

□ This application has been examined ☒ Responsive to communication filed on 8-9-96 ☒ This action is made final.

A shortened statutory period for response to this action is set to expire 3 month(s), 0 days from the date of this letter.
Failure to respond within the period for response will cause the application to become abandoned. 35 U.S.C. 133

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☒ Notice of References Cited by Examiner, PTO-892.
2. □ Notice of Draftsman's Patent Drawing Review, PTO-948.
3. □ Notice of Art Cited by Applicant, PTO-1449.
4. □ Notice of Informal Patent Application, PTO-152.
5. □ Information on How to Effect Drawing Changes, PTO-1474.
6. □ ______.

Part II SUMMARY OF ACTION

1. ☒ Claims 1-56 are pending in the application.

   Of the above, claims ______ are withdrawn from consideration.

2. ☒ Claims 16 have been cancelled.
3. □ Claims ______ are allowed.
4. ☒ Claims 1-15, 17-56 are rejected.
5. □ Claims ______ are objected to.
6. □ Claims ______ are subject to restriction or election requirement.
7. ☒ This application has been filed with informal drawings under 37 C.F.R. 1.85 which are acceptable for examination purposes.
8. □ Formal drawings are required in response to this Office action.
9. □ The corrected or substitute drawings have been received on ______. Under 37 C.F.R. 1.84 these drawings are □ acceptable; □ not acceptable (see explanation or Notice of Draftsman's Patent Drawing Review, PTO-948).
10. □ The proposed additional or substitute sheet(s) of drawings, filed on ______. has (have) been □ approved by the examiner; □ disapproved by the examiner (see explanation).
11. □ The proposed drawing correction, filed ______, has been □ approved; □ disapproved (see explanation).
12. □ Acknowledgement is made of the claim for priority under 35 U.S.C. 119. The certified copy has □ been received □ not been received □ been filed in parent application, serial no. ______ ; filed on ______.
13. □ Since this application apppears to be in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under Ex parte Quayle, 1935 C.D. 11; 453 O.G. 213.
14. □ Other

EXAMINER'S ACTION

Part III DETAILED ACTION

Applicant's arguments filed 08-09-96 have been considered but are moot in view of the new grounds of rejection.

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> *(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.*

Claim 1 and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vetter "Mosaic and the World-Wide Web" and further in view of Hansen "Andrew as a Multiparadigm Environment for Visual Languages".

As per claim 1, Vetter disclosed a method for running an application program in a computer network environment essentially as claimed, comprising:

providing at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment [the World-Wide Web];

executing, at said client workstation, a browser application [Mosaic], that parses a distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processes specified by the text format;

utilizing said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein said first hypermedia document is displayed within a first browser-controlled window on said client workstation and wherein said first distributed hypermedia document includes an embed text format [p.52 col.1 URL] that specifies the location of an object external to the first distributed hypermedia document and that specifies type information utilized by said browser to identify and locate an executable application [CGI script and external viewer] external to the first distributed hypermedia document;

invoking, with said browser application, said executable application [it is known that that Mosaic can launch CGI scripts on the server and external viewer applications on the user's workstation] to display and process said object.

Mosaic does not display and process said object within the first browser-controlled window while a portion of said first distributed hypermedia document continues to be displayed within said browser-controlled window. The external viewer is launched into a separate window to process the object.

Hansen teaches "it may be adequate to display each sublanguage element in a separate window, but this runs the risk of chaotic imagery among which it is difficult to discern the relationships among program segments. Instead, the author should have the power to organize the program fragments for perusal by the reader. The organization itself, together with commentary, aids the reader in comprehending the program." [p.256 col.1]

Vetter discloses that Mosaic's functionality can be extended by having custom servers and by letting other applications control its display remotely [p.50 2nd paragraph]. Hence, it would have been obvious for one of ordinary skill in the art to extend Mosaic's functionality to enable external application to display and process the object within the browser-controlled window because it would have improved the system by reducing clustering of the display and aiding the reader comprehension of the hypermedia document.

As per claim 44, it is rejected under similar rationale as for claim 1 above.

Claims 2-5, 10-14, 24-27, 45-48, and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vetter "Mosaic and the World-Wide Web", Hansen "Andrew as a Multiparadigm Environment for Visual Languages", and further in view of Filepp et al. US patent 5,347,632.

As per claim 2, Vetter does not specifically disclose interatively controlling the executable application via communication over the network.

Filepp discloses a distributed system for interactive control of object through communication over network wherein the controllable applications communicate over the network to provide user interaction of the object within the current document ['Page' -see col.5 lines 5-18 and col.8 lines 15-28]. Vetter disclosed that Mosaic and the WWW currently lack direct support for application-specific data and support for controlling the presentation of nontext data [p.53 col.3]. Hence, it would have been obvious for one of ordinary skill in the art to adapt Filepp's teaching to work the WWW because it would have improved the system to provide more presentation and control functionality's. Vetter suggested letting other applications control Mosaic display remotely [p.50 2nd paragraph]. Hence, it is apparent that the system as modified would interactively controlling said controllable application from said client workstation via communications sent over said distributed hypermedia environment.

As per claim 3, it is well known in distributed computing to executing, on the network server, one or more instructions in response to commands from a client and sending information from said network server to the client workstation in response to said executed instructions. It is apparent in the system as modified that the client would process the information and interactively control said controllable application.

As per claim 4, Filepp teaches the instruction for controlling the object reside on the client workstation [col.7 line 19].

As per claim 5, it is apparent in the system as modified that communication continue to be exchange between the controllable application and the browser in order for the controllable application to control the object within the browser's window.

As per claims 10-14, the methods recited are apparent in the system as modified.

As per claims 45-48, they are rejected under similar rationale as for claims 2-5 above.

As per claim 24, it is rejected under similar rationale as for claims 1+2 above. The references do not specifically disclose the step of transferring ..., accepting ..., executing ..., communicating ..., using ... The steps recited is inherent in the prior art as modified because:

> It is well known in the art, at the time of the invention, that HTML documents contains links specified by URL's. It is known that HTML documents transfers involves HTTP protocol messages. The process involves:

transferring, over the network, a hypermedia document [the HTML document] with embedded objects [URL links, mapped images, fill-in forms, etc.] from a server computer to the client computer;

parsing the document by the browser to locate reference to external objects [URL's, images, etc.];

accepting first signals from the user input device [clicking on an URL link, or a mapped image, or a form's 'submit' button]

issuing commands [HTTP message with the linked URL, or coordinates where the mapped image was clicked, or the form's content] from the client computer to a first computer in response to the signal [it is known that an HTTP message in an HTML document can direct to any computer connected to the Internet that accept HTTP protocol];

executing instructions by a first additional computer [Filepp - col.4 lines 35-43 , "multiple servers"] and generate information about manipulating the embedded object; communicating the information to the client; and using the client to manipulate the object according to the communicated information [Filepp col.8 lines 15-28].

As per claims 25, 27, the document is a hypermedia document [Vetter p.49].

As per claim 26, Filepp discloses using multiple computer to response to issued commands [col.4 lines 35-43 , "multiple servers"].

As per claim 55, it is rejected under similar rationale as for claim 24 above.

Claims 6-9, and 49-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vetter "Mosaic and the World-Wide Web", Hansen "Andrew as a Multiparadigm Environment for Visual Languages", Filepp et al. US patent 5,347,632, and further in view of Rizzo "What's OpenDoc?" (prior art submitted by applicant).

As per claims 6-9 and 49-52, Vetter, Hansen, Filepp do not specifically disclose application being a spreadsheet, a database, or word processor program. Rizzo discloses a systems that allows for embedding object of different applications (word processing, spreadsheet, database, movie) in one document and manipulation of the object within the document using functions of the corresponding application. Hence, it was well within the skill on one of ordinary skill in the art to provide controllable application for database, spreadsheet, word processing, etc. functions. The type of program provided would have been a matter of design choice.

As per claim 53, it is apparent the HyperText Markup Language would be used to specify the controllable application since the system is for use in the WWW.

Claims 28-33, 34-43, and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vetter "Mosaic and the World-Wide Web", Hansen "Andrew as a Multiparadigm Environment for Visual Languages", Filepp et al. US patent 5,347,632, and further in view of Moran "Tele-Nice-Slicer: A New Tool for the Visualization of Large Volumetric Data".

As per claims 28, 30, and 32 the references do not specifically disclose multi-dimensional image displayable in plurality of orientations, and function to determine the new orientation and rendering of image. The type of objects and functions provided would have been a matter of design choice.

Moran discloses a distributed system for interactive control and visualization of graphical object through communication over network. Moran teaches determining orientation and rendering of images by sending command comprising of text fields [p.3 col.1]. Vetter disclosed that Mosaic and the WWW currently lack direct support for application-specific data and support for controlling the presentation of nontext data [p.53 col.3]. Hence, it would have been obvious for one of ordinary skill in the art to adapt Moran teaching to the WWW because it would have improved the system to provide powerful visualization, presentation and control to WWW users.

As per claims 29, 31, 33, Vetter teaches the document is a hypermedia document [Vetter p.49].

As per claims 34 and 56, they are rejected under similar rationale as for claim 24 above. The references do not specifically disclose multi-dimensional image displayable in plurality of orientations, and function to determine the new orientation and rendering of image.

Moran discloses a distributed system for interactive control and visualization of graphical object through communication over network. Moran teaches determining orientation and rendering of images by sending command comprising of text fields to server[p.3 col.1]. Vetter disclosed that Mosaic and the WWW currently lack direct support for application-specific data and support for controlling the presentation of nontext data [p.53 col.3]. Hence, it would have been obvious for one of ordinary skill in the art to adapt Moran teaching to the WWW because it would have improved the system to provide powerful visualization, presentation and control to WWW users.

Moran does not specifically disclose a second server. However, it would have been obvious for one of ordianry skill in the art to provide plurality of servers to speed up processing.

As per claims 35, 37, 39, 41, and 43, Vetter teaches the system is a distributed hypermedia environment [p.49].

As per claim 36, Moran teaches distributing the processing on various computers [client - server]. It would have been obvious for one of ordinary skill in the art to distribute the processing to the machine such that the instructions is executed faster.

As per claims 38 and 40, Moran teaches determining orientation and rendering of images [p.2 - p.3].

As per claim 42, Moran teaches dynamically manipulate the object [p.2 - zoom]. It is apparent that the system as modified would accept signal from user input to indicate a second orientation of an object.

Claim 15, 17-23, and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vetter "Mosaic and the World-Wide Web" and further in view of Hansen "Andrew as a Multiparadigm Environment for Visual Languages" and Moran "Tele-Nice-Slicer: A New Tool for the Visualization of Large Volumetric Data".

As per claim 15, it is rejected under similar rationale as for claim 1 above. Vetter and Hansen do not specifically teach a multidimensional data visualization application.

Moran discloses a distributed system for interactive control and visualization of graphical object through communication over network. Moran teaches determining orientation and rendering of images by sending command comprising of text fields [p.3 col.1]. Vetter disclosed that Mosaic and the WWW currently lack direct support for application-specific data and support for controlling the presentation of nontext data [p.53 col.3]. Hence, it would have been obvious for one of ordinary skill in the art to adapt Moran teaching to the WWW because it would have improved the system to provide powerful visualization, presentation and control to WWW users.

As per claims 17-22, the recited limitations - volume visualization, 2d image, image analysis, animated sequences, geometric viewer, and molecular modeling - would have been a matter of design choice because they are merely well known visualization methods.

As per claim 23, it is apparent in the system as modified that communication continue to be exchange between the multidimensional data visualization application and the browser in order for the visualization application to control the object within the browser's window.

As per claim 54, it is rejected under similar rationale as for claim 15 above.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Wynne et al. "Lean Management, Group Support Systems, and Hypermedia: a combination whose time has come", discloses a hypermedia product - "HyperNet" - which has 'active link' for triggering execution of external programs [see p.113].

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for response to this final action is set to expire THREE MONTHS from the date of this action. In the event a first response is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event will the statutory period for response expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dung Dinh whose telephone number is (703) 305-9655. The examiner can normally be reached on Monday-Thursday from 7:00 AM - 4:30 PM. The examiner can also be reached on alternate Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas Lee can be reached at (703) 305-9717. The fax phone number for this group is (703) 308-5359.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

THOMAS C. LEE
SUPERVISORY PATENT EXAMINER
230

Dung Dinh
Dec. 10, 1996

Notice of References Cited	Application No. **08/324,443**	Applicant(s) **Doyle et al.**		
	Examiner **D. Dinh**		Group Art Unit **2317**	**Page 1 of 1**

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
	A	5,347,632	09/13/94	Filepp et al.	395	200.09
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
	N						
	O						
	P						
	Q						
	R						
	S						
	T						

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U	Vetter, Ronald "Mosaic and the World-Wide Web," Computer Magazine, v.27 Iss.10, pp.49-57.	10/94
	V	Wynne et al. "Lean Management, Group Support Systems, and Hypermedia: a Combination Whose Time Has Come," System Sciences, 1993 Anuall Hawaii Int'l Conf., pp.112-121.	1993
	W	Hansen, Wilfred "Andrew as a Multiparadigm Environment for Visual Languages," Visual Languages, 1993 IEEE Symposium, pp.256-260.	1993
	X	Moran, Patrick "Tele-Nicer-Slicer-Dicer: A New Tool for the Visualization of Large Volumetric Data", NCSA Technical Report (TR014), August 1993	1993

906 PH Ex. 5

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in ~~an envelope addressed to:~~ Assistant Commissioner for Patents, By FAX 703-308-5359
Washington, D.C. 20231,
on 1-8-97

TOWNSEND and TOWNSEND and CREW LLP

By

#7

PATENT

Attorney Docket No. 023071-553

OFFICIAL

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

MICHAEL D. DOYLE et al.

Application No.: 08/324,443

Filed: 10/17/94

For: EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

Examiner: D. Dinh

JAN 08 1997

Art Unit: 2317

DECLARATION OF MICHAEL D. DOYLE UNDER RULE 131

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

I, MICHAEL D. DOYLE, hereby declare that:

1. I am a co-inventor of the subject matter disclosed and claimed in U.S. Patent Application No. 08/324,443.

2. The subject matter claimed in the above patent application was reduced to practice in this country prior to October, 1994, the publication date of the cited reference entitled "Mosaic and the World-Wide Web" by Vetter.

3. The reduction to practice of the claimed invention is evidenced by ATTACHMENT A, which is a transcript of the audio portion and still photographs of a video tape presented to an audience of scientists prior to October, 1994.

4. As stated in ATTACHMENT A, starting at the bottom of page 2, interface and control software had been developed that allows the embedding of a visualization application within a Mosaic document. As is apparent from the photographs, the object is displayed and processed within the browser-controlled window. The visualization application is external to the hypermedia document displayed by the browser.

MICHAEL D. DOYLE et al.
Application No.: 08/324,443
Page 2

PATENT

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Dated: 1-2-97, 1997.

MICHAEL D. DOYLE

CEK:db
E:\02307I\553\DECL.01

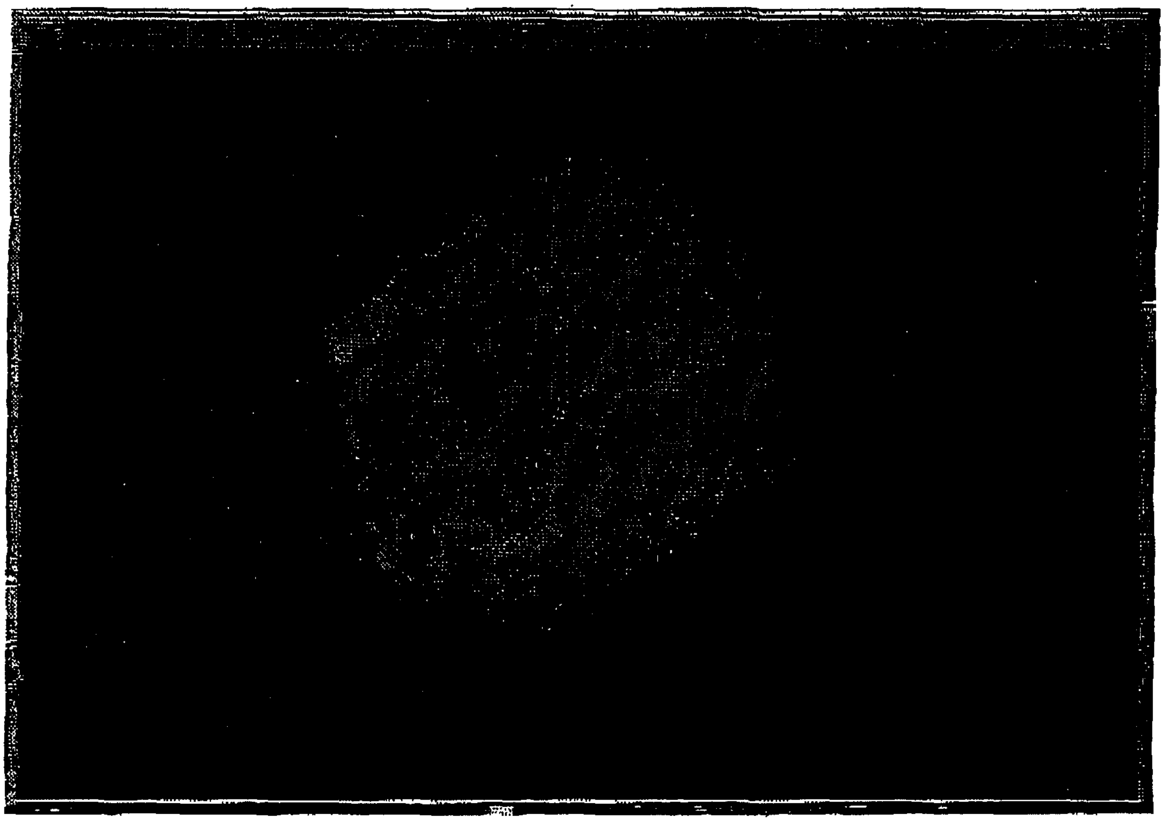
Attachment "A"

M.D. Doyle, et. al., ***The Virtual Embryo: VR Applications in Human Developmental Anatomy,*** presented at "Medicine Meets Virtual Reality II, Interactive Technology and Healthcare: Visionary Applications for Simulation, Visualization & Robotics," sponsored by the UCSD School of Medicine and the Advanced Projects Research Agency, San Diego, CA, **January 27, 1994**.

Video Presentation Transcript:

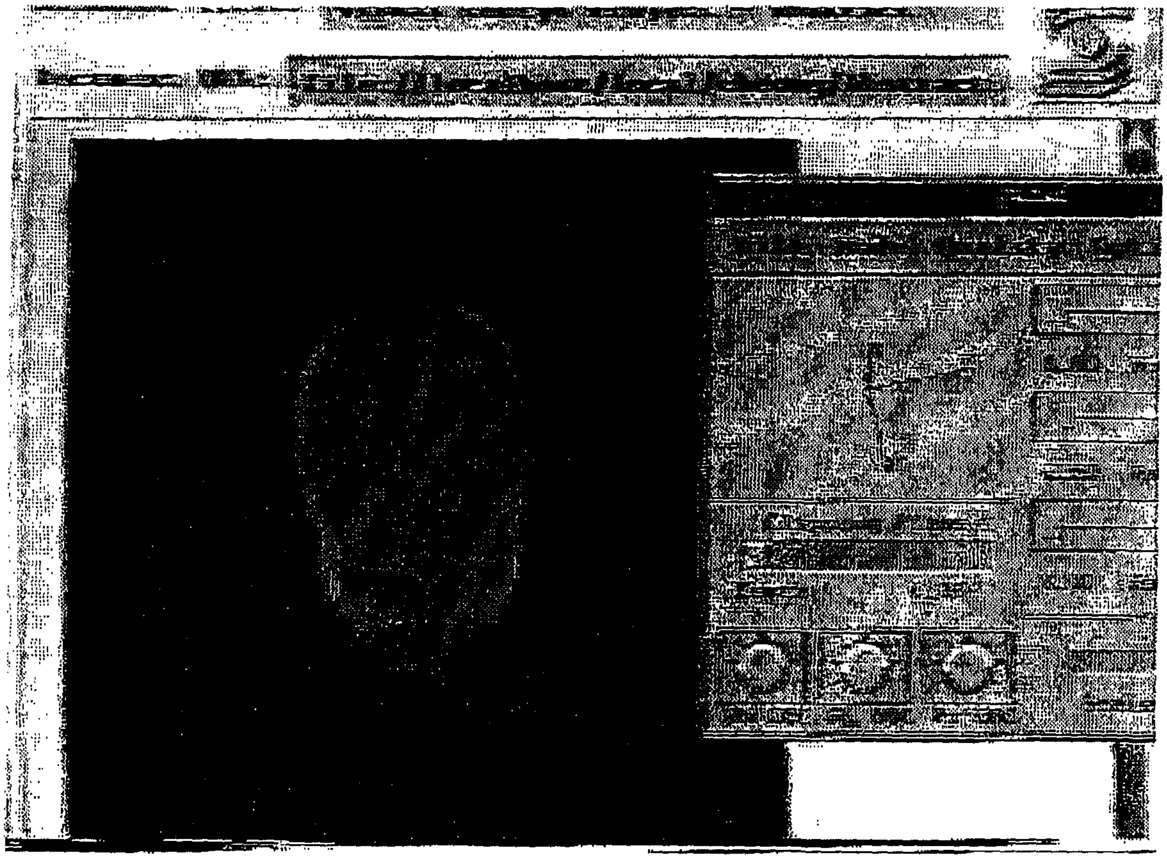
This is a status report of some of the work that's been accomplished during the first years of the Visible Embryo Project.

One of our first tasks was to develop some volume visualization software that we could use for imaging and analysis of the embryo reconstructions that we planned to create during the full term of the embryo project. One thing that was an absolute requirement was that this software be able to distribute its computational load across a network of graphics computers that weren't necessarily all in the same place. Basically we wanted to be able to have computers that could be all over the country connected by high -speed networking, able to contribute to a computation of three-dimensional datasets.



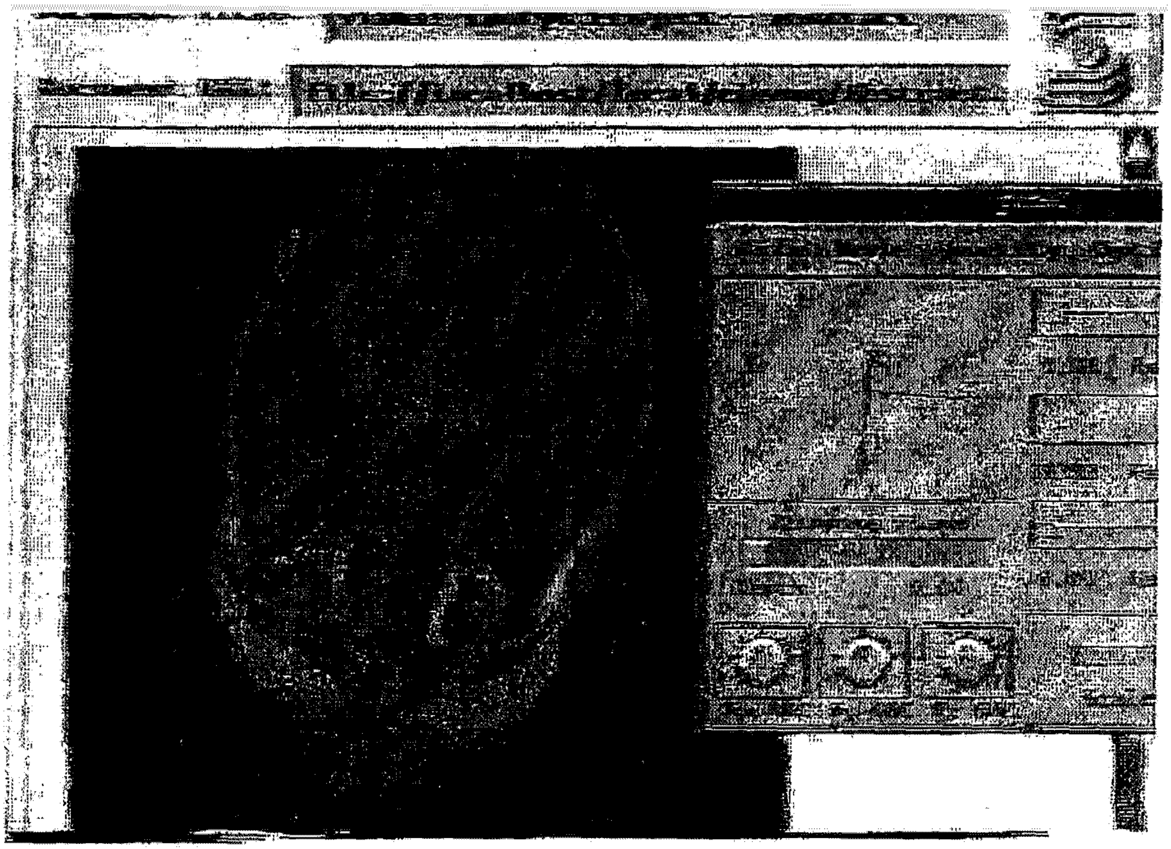
What you see here is a package called "VIS," which was developed in our group, for three-dimensional volume visualization. This is a very portable package that has been generated using as generic code as possible, although this particular image that you see here is running on a Silicon Graphics Reality Engine II, which is optimized for volume visualization. We need to use that kind of high-speed optimization to accomplish the real-time interactivity that we need to accomplish for this project. And as you see, as this rotates, and it starts rotating faster and faster, only a very powerful graphics - basically a graphics supercomputer - can accomplish this much computation on a three dimensional dataset. One of our goals is to allow anybody on the Internet with a very low-level access workstation to accomplish this kind of interactivity through their network connection, and the way that we do that is through a client-server architecture where we have a very powerful server computer accessed by a very low-end client machine.

We decided early on to use NCSA's Mosaic program and the World Wide Web to integrate access to this system. One problem with Mosaic is, as it exists today, is that the images within Mosaic are typically static or passive-playback images.



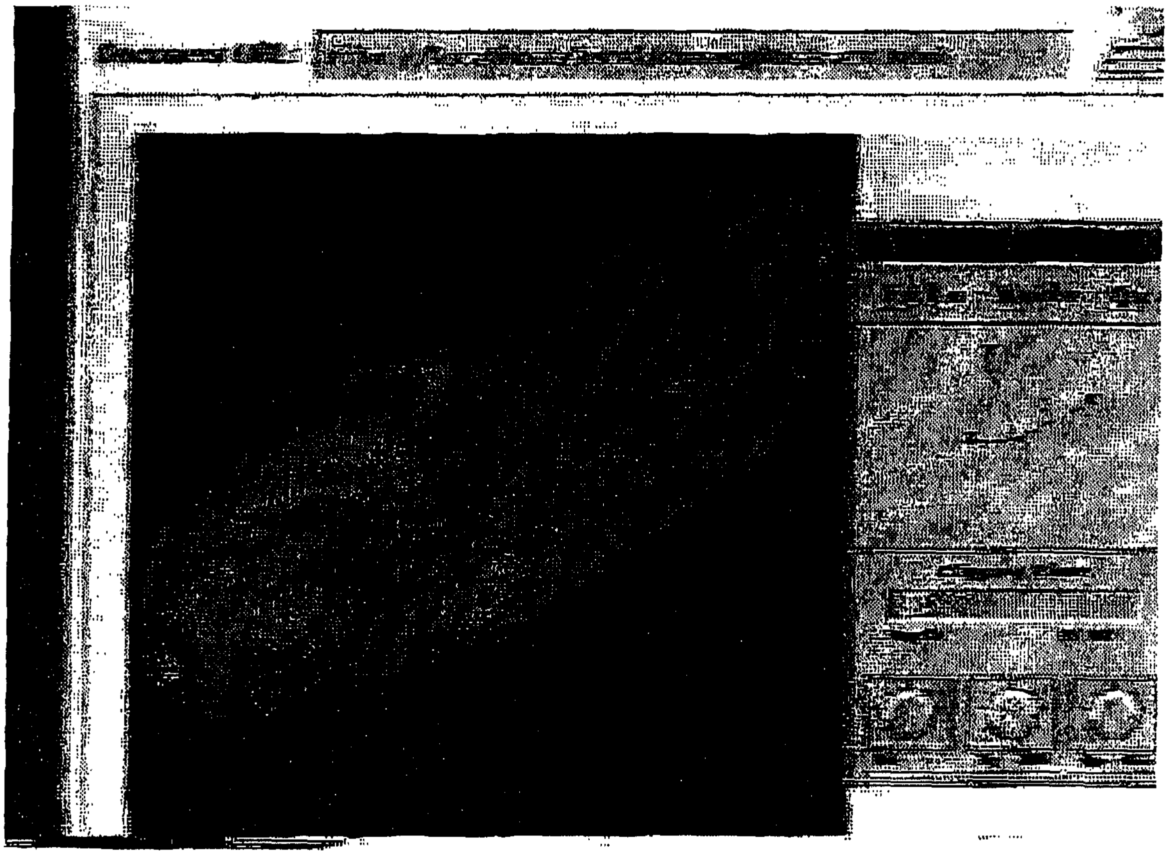
What you see here is an enhancement that we've created to the Mosaic interface and control software that allows the embedding of a dynamic real-time

visualization application within a Mosaic document. You see a head, a volume MRI model of a human head, that's being rotated in real-time interactively by the viewer. You see a little panel to the right which is, it's a control panel that's popped up - it's really external to Mosaic itself but it can talk to the internal control programs that drive the Mosaic client - and allow you to interactively control the display from within Mosaic. This is actually controlling the volume visualization software that you just saw a few minutes ago.
By moving around the controls in the control panel we can do things like rotate, we can control slices through the dataset, and so on. As you can see, there's rotation in x, y, and z planes. We can also compute arbitrary oblique sections through the data and look at the internal anatomy. Here we can see the brain of this individual; we can rotate and view that section of the dataset from a variety of vantage points. There are zooming capabilities that allow us to zoom up on the data or zoom back to look at things in more or less detail, and you can see here that once zoomed, we're moving our cutting plane down through the dataset and looking at more and more inferior levels.



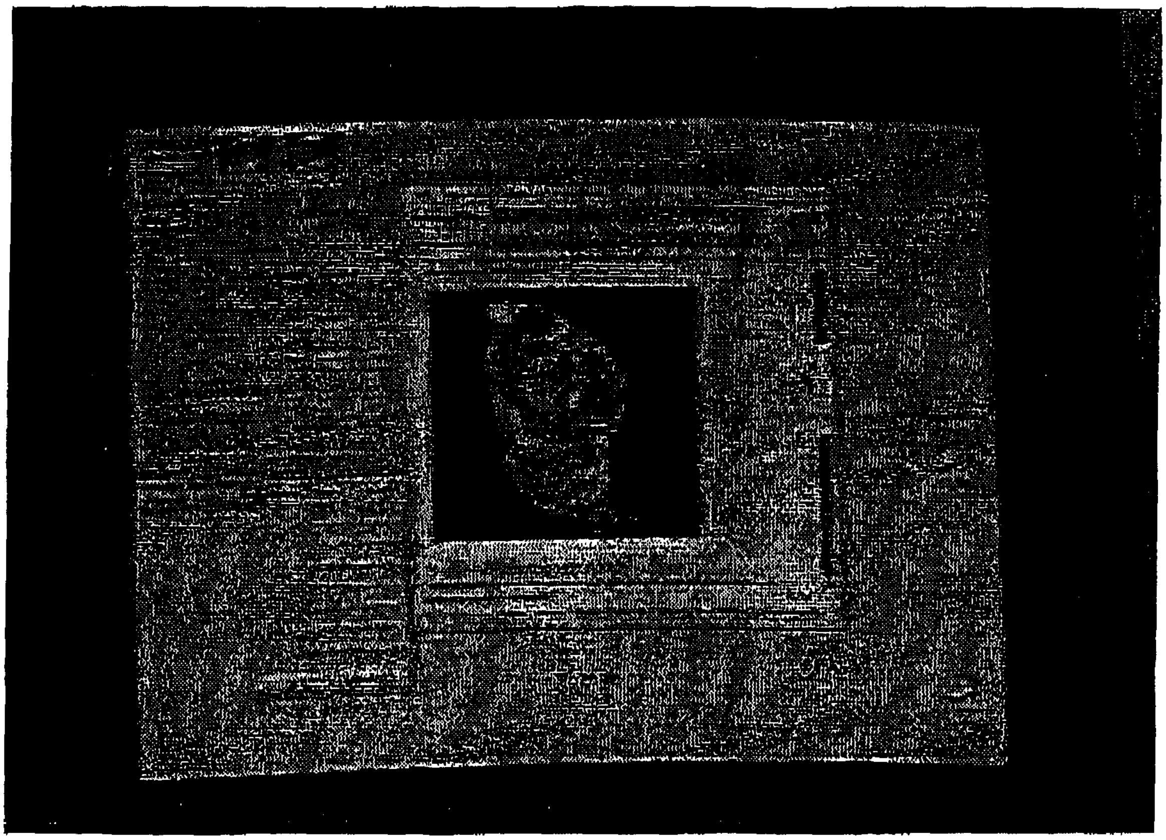
Normally graphics in Mosaic are static, as I said, but this embedding of graphic applications within Mosaic is really going to form the basis of how we integrate information access through the Visible Embryo Project. What you are about to see is our prototype system of a Visible Embryo Mosaic document that has embedded realtime visualizations within it.

We just loaded the Visible Embryo Mosaic page, the system is about to scroll down this page - it is an abstract about the Visible Embryo Project - and then we see a window. It looks like a static image just like is normally found within the World Wide Web databases that you can access today through Mosaic, but you can see that suddenly, by moving controls on the control panel, we can zoom in and see that this is a reconstruction of a seven week old human embryo. This is a reconstruction from approximately 2900 serial cross sections of an embryo sectioned about in the 1930s. It's part of the Carnegie Collection of Human Embryology.



We're looking at this in volume visualization mode, we can rotate the embryo around, we can see internal structures, neurological structures; just in the lower abdomen area, we can see the liver, the arms are very evident - so we are actually looking through the dataset. We can also slice through this dataset obliquely, and look at the internal anatomy that way as well. We can load a volume visualization table that allows us to interactively enter tissue characteristic numbers that control the translucency, transparency, or opacity of various ranges of voxal intensity. And what we've done now is we've made the exterior of the embryo a little more opaque so that as we rotate around with an oblique cutting plane we can see the difference between the cutting plane and the exterior of the embryo a little better. So now we're looking, slicing - we've rotated the embryo to an inferior view and we're looking up at the embryo from below. And we can see, we've just gone through the heart region, we're going through the liver, we're moving inferiorly and we start to get to an area where we can see the herniated gut.

Now the real key to all of this is that these are embedded visualizations. We're actually creating documents that are - I guess you'd call them currently compound documents where you have the traditional type of information, but you've also got, within that document, links to the raw data rather than just pictures generated from data. This allows you to tie together representations of data with the actual data themselves as well as with notes and different kinds of descriptive textual information based on that data.

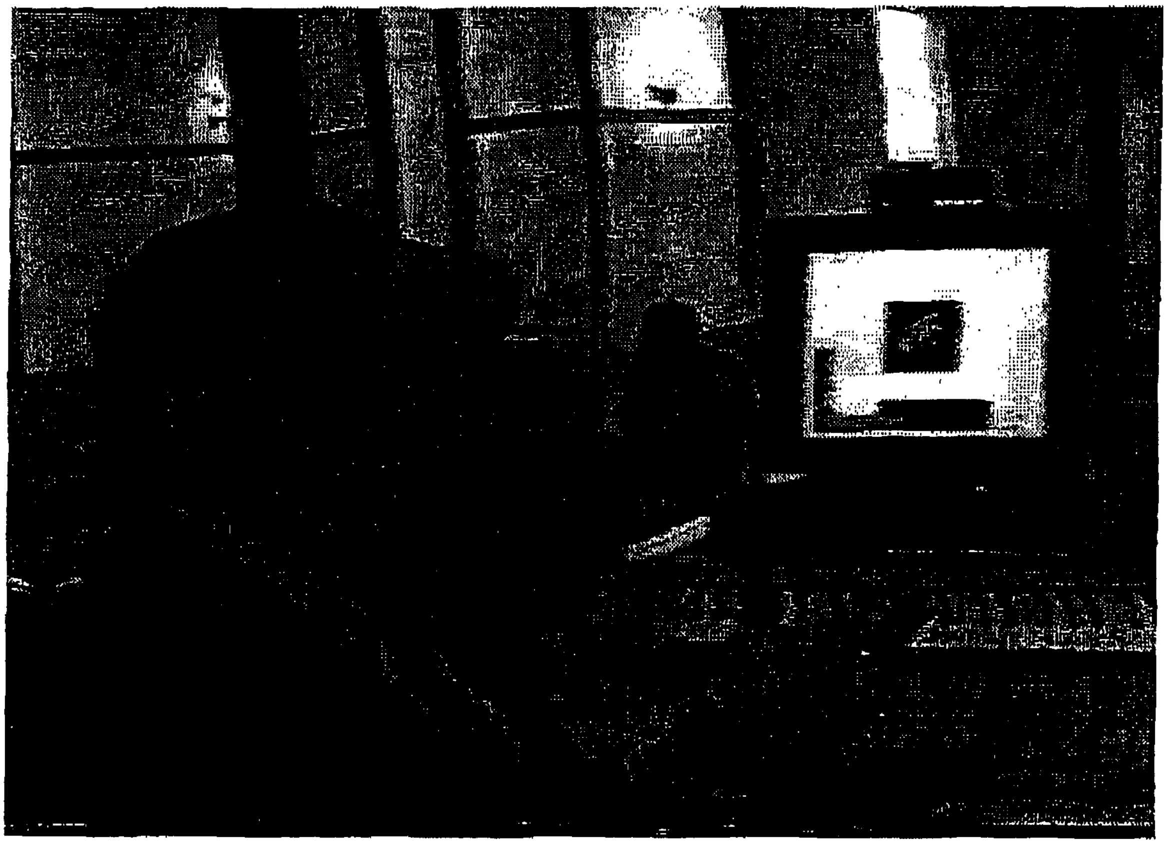


Our basic objective here is really to create what we're calling a national meta-center which is going to be a computational resource for the entire nation that allows people interested in many different areas, including developmental biology, also multi-dimensional imaging and high-speed networking, and parallel computing. All these people can access this database. And the parallel nature of the computation that's taking place is invisible to the user. They log in through a Mosaic window, and that window is giving them very high-performance control of interactive visualizations of datasets. By scrolling the window, we can see that this is actually embedded within the Mosaic document.

During a recent demonstration of this technology at the corporate briefing center at Silicon Graphics Corporation in Mountain View, California, I discussed some of the implications of this technology for researchers of human genetics.

——

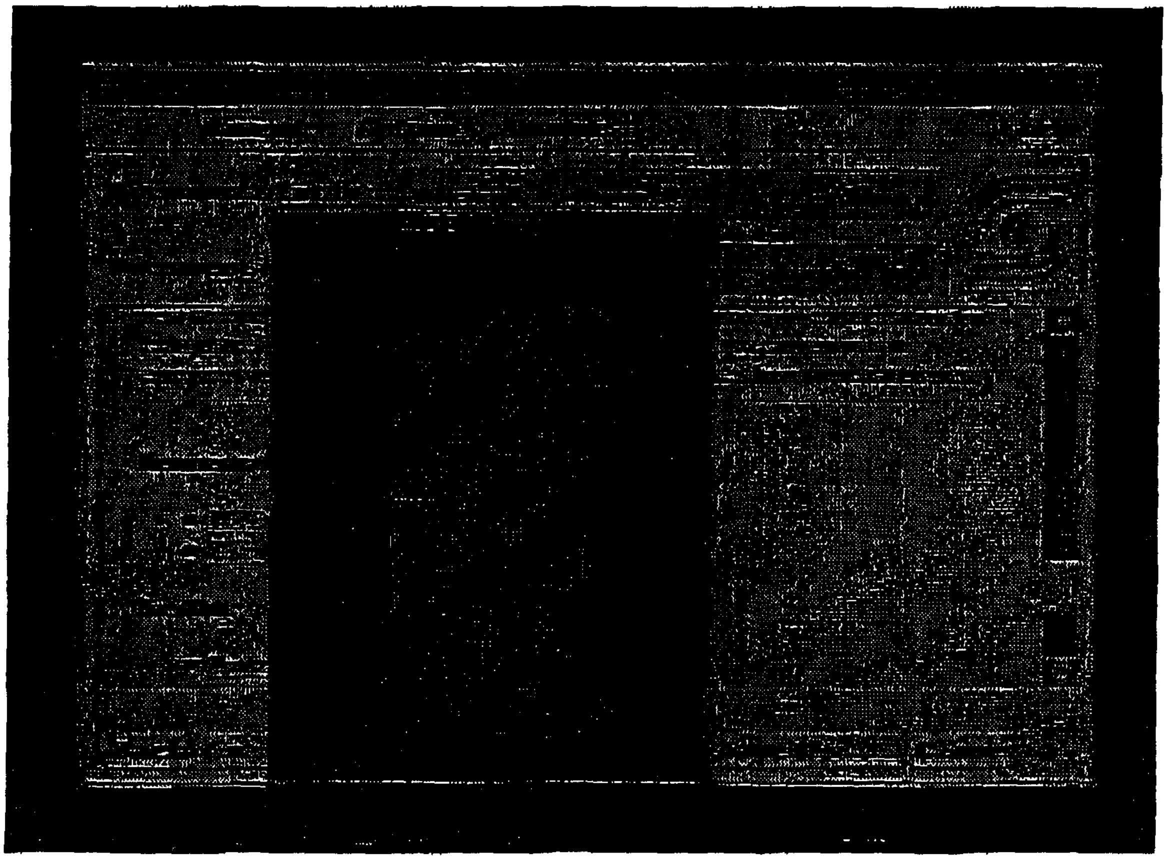
"We're also looking at using these models as a basis for creating three-dimensional maps of gene expression, which is a way to correlate the findings of the Human Genome project within a context that everyone uses. It sort of sets up a standard space within which everybody can report their findings, so that you can finally have some way of comparing studies that happen in different laboratories.



If you're studying the three-dimensional distribution of gene expression of a gene relating to heart development, what you do now is you have a little fluorescent marker that glows under an ultraviolet light, and you use confocal microscopy to develop a three-dimensional model of it, and then you say, well, it's on here and it's on there and it's on there and you try to describe it in anatomical terms but it is a qualitative description, right?.

But here there would be a standard anatomy space that people could use to describe their findings so that they could, rather than say, yeah, we saw five different studies that said that this was expressed at the bifurcation of the aorta with the Common Carotid artery - you don't have to do in terms of verbal descriptions, you can do it in terms of a true measurable Cartesian coordinate system.

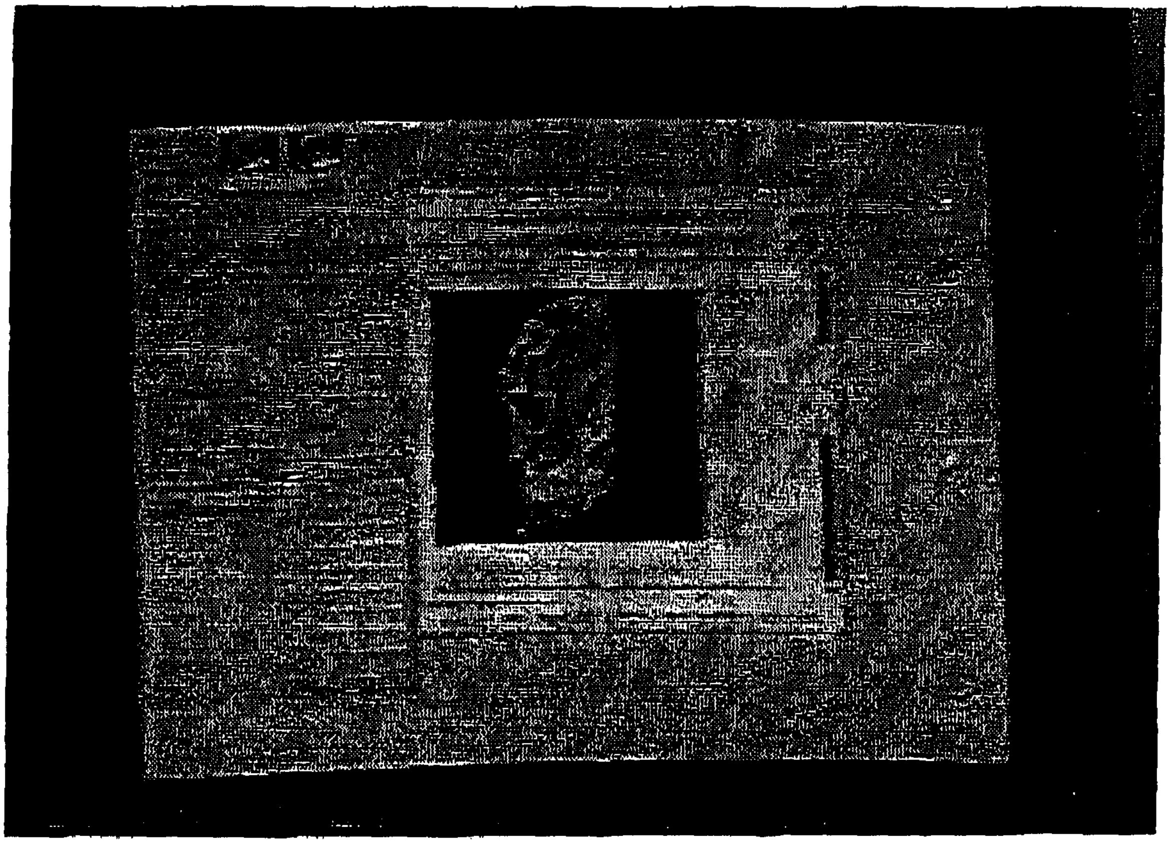
If you take your current version of Mosaic, the kind that is accessible for free through the Internet today, and log onto our home page of the Visible Embryo Project, what you'll see is a series of multi-media documents that basically give you information about the status of the Visible Embryo Project and the status of our current proposal development efforts. You'll be able to load MPEG movies of visualizations of human embryos, as you can see here.



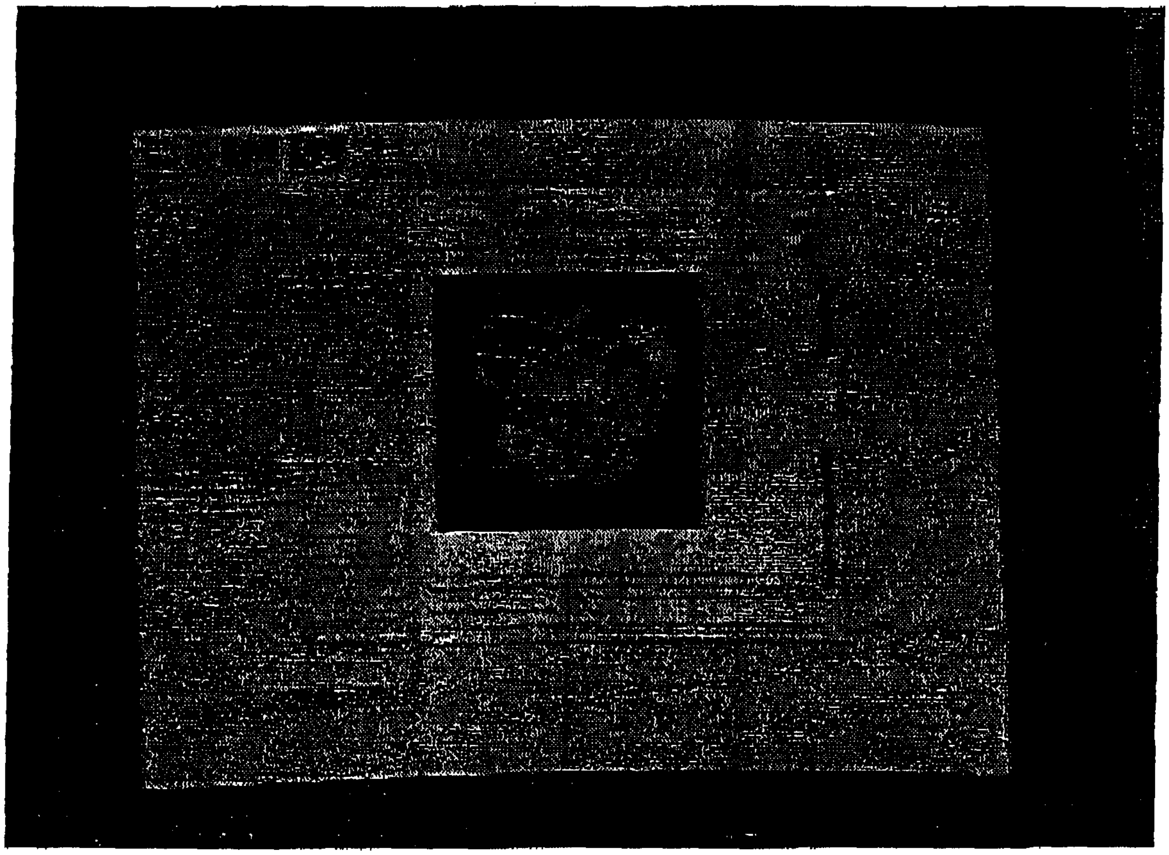
One thing you should keep in mind is that this little MPEG movie is just a canned movie, it's not interactive. Once you hit Play it just goes and it plays and then it goes away, but you can't stop it and interact with it, and rotate that embryo, for instance, to different vantage points.

Also available is an image that shows some of the early work, some screen shots representing the volume visualization tool current as of about last summer. We've come much farther and in fact a lot of the video that you saw earlier in this presentation shows you the current status of this volume visualization package. Last summer, that imaging package was separate from Mosaic, as you see it's off to the right, and Mosaic could just call it but couldn't actually embed visualizations. Now, everything is tied together into a single multi-media document.

You'll also see articles that relate to the Visible Embryo Project. This project has been going on for several years now, mostly on the coattails of other research projects, collaborators funding it wherever they could find the money. But already a significant body of literature is starting to be built up around this project.



Of course, in the very near future, you'll be able to log on through an enhanced version of Mosaic. **We're working together with the National Center for Supercomputing Applications on enhancing the standard release of Mosaic to allow these capabilities.** And you'll be able to access interactive dynamic visualizations that are being served by a network of high-end supercomputers across the country. Even if you're only accessing the system with a machine like a Macintosh or PC, you'll still be able to access the power of these supercomputers from your own location.



What I've attempted to demonstrate in the last several minutes in this presentation is that there's been a considerable amount of work already done in this project that we call the Visible Embryo Project. Many collaborators across the country have worked together to create a set of enabling technologies to allow this project to accomplish its goals. The Visible Embryo Project represents an effort to serve the needs of both the biology community as well as the information science community, in that we are attempting through current applications in information technology to break through barriers that have prevented biological researchers from asking and answering questions about the most fundamental mechanisms of human growth and development. We're also creating a technology development testbed for the information sciences that will allow researchers to push the envelope, so to speak, of information technologies to their very limits.

SPEAKER CONSENT FOR AUDIO/VIDEO RECORDINGS

We would like to audio/video tape your presentation at **Medicine Meets Virtual Reality II: Interactive Technology and Healthcare,** January 27-30, 1994.

Audio and video tapes will be available for immediate distribution to attendees and will be marketed and sold after the conference. Your colleagues will be able to benefit from your remarks by listening to cassettes and viewing tapes whether or not they were in attendance. The tapes will be copyrighted and marketed under Title 17 of the U.S. Code or other law as may be enacted, and all rights to royalties, if any, in conjunction with cassette sales shall hereby be assigned to Medicine Meets Virtual Reality. This agreement does not preclude the publication by you of your paper, speech or comments at any time and in any format.

Please cooperate with the audio/visual taping staff to obtain the best possible recordings by using the microphone at all times and repeating the questions that are asked by persons not near a microphone.

Sign this form below and return it to us at your earliest convenience.

Thank You.

I hereby give permission to ***Medicine Meets Virtual Reality*** to record and distribute audio and video tapes containing my presentation as outlined above.

Signature Michael D Doyle Date 11-30-93

Please print name Michael D. Doyle

Thank you very much for your cooperation. Remember to pick up the complimentary audiotape of your presentation before leaving the meeting.

Please return this completed form to:
MEDICINE MEETS VIRTUAL REALITY
P.O. Box 23220, San Diego, CA 92193
For further information call 619/751-8841,
or Fax 619/751-8842.

UNIVERSITY OF CALIFORNIA, SAN DIEGO
OFFICE OF CONTINUING MEDICAL EDUCATION
DISCLOSURE STATEMENT

It is the policy of the UCSD Office of Continuing Medical Education to insure balance, independence, objectivity, and scientific rigor in all its individually sponsored or jointly sponsored educational programs. All faculty participating in any UCSD sponsored programs are expected to disclose to the program audience <u>any</u> real or apparent conflict(s) of interest that may have a <u>direct bearing on the subject matter</u> of the continuing education program. This pertains to relationships with pharmaceutical companies, biomedical device manufacturers, or other corporations whose products or services are realated to the subject matter of the presenatation topic. The intent of this policy is not to prevent a speaker with a potential conflict of interest from making a presentation. It is merely intended that any potential conflict should be identified openly so that the listeners may form their own judgments about the presentation with the full disclosure of the facts. It remains for the audience to determine whether the speaker's outside interests may reflect a possible bias in either the exposition or the conclusions presented.

MEDICINE MEETS VIRTUAL REALITY II

INTERACTIVE TECHNOLOGY & HEALTHCARE

January 27-30, 1994

PRESENTER'S NAME: Michael D. Doyle
(please print or type)

I have no actual or potential conflict of interest in relation to this program.

Signature 11-30-93 (date)

I have a financial interest/arrangement or affiliation with one or more organizations that could be perceiveed as a real or apparent conflict of interest in the context of the subject of this presentation.

<u>Affiliation/Financial Interest</u>	<u>Name of Organization(s)</u>
Grant/Research Support	
Consultant	
Speakers' Bureau	
Major Stock Shareholder	
Other Financial or Material Support	

Signature (date)

Your cooperation in complying with these guidelines is appreciated.

Please return this form as soon as possible to: Medicine Meets Virtual Reality II, P.O. Box 23220, San Diego, CA 92193
Fax: 619-751-8842; Phone: 619-751-8841

906 PH Ex. 6

#8

I hereby certify that this correspondence is being
sent by facsimile transmission to: D. Dinh
Fax No.: 1-703-308-5359
Assistant Commissioner for Patents,
Washington, D.C. 20231,
on 1-8-97

TOWNSEND and TOWNSEND and CREW LLP

By Irene Rodas

PATENT

Attorney Docket No. 02307I-553

OFFICIAL

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

MICHAEL D. DOYLE et al.

Application No.: 08/324,443

Filed: 10/17/94

For: EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

Examiner: D. Dinh

Art Unit: 2317

RESPONSE AFTER FINAL

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

The following is responsive to the Office Action mailed December 13, 1996:

REMARKS

Claims 1-15 and 17-56 are now pending in the application. Reexamination and reconsideration are requested.

Claims 1 and 44 are rejected under 35 U.S.C. §103 as being unpatentable over Vetter "Mosaic and the World-Wide Web" and further in view of Hansen "Andrew as a Multiparadigm Environment for Visual Languages."

In claim 1, the distributed hypermedia document includes an embed text format that specifies the location of an object external to the distributed hypermedia document and that specifies type information utilized by the browser to identify and locate an executable application external to the distributed hypermedia document. The browser invokes the executable application to display and process the object within the browser window.

The Vetter reference states that "Users can extend Mosaic's functionality by creating custom servers and letting other applications control its display remotely."

The Hansen reference discloses a multiparadigm language environment for visual languages that supports not only graphical interaction but also file storage, printing, compilation, and execution. It is suggested that each sublanguage element be displayed in the same window.

The examiner states that Vetter discloses that Mosaic's functionality can be extended by having custom servers and letting other applications control its display remotely. Hence, it would have been obvious to extend Mosaic's functionality to enable external application to display and process the object within the browser-controlled window because it would have improved the system by reducing clustering of the display and aiding the reader comprehension of the hypermedia document. This rejection is respectfully traversed for the following reasons.

A Rule 131 Declaration is submitted showing that applicant's invention was reduced to practice prior to the publication of Vetter. Accordingly, the feature of controlling Mosaic's display remotely is not disclosed in the prior art.

Further, the feature described in Vetter was not present in Mosaic prior to the reduction to practice of the invention. The feature of remotely controlling Mosaic's display, the common-client interface (CCI), was incorporated into Mosaic based on the work done by the applicants as evidenced by Attachment I, a proposal to the NCSA, the developers of Mosaic, to modify Mosaic to allow real-time visualization tools to be embedded in Mosaic. (Michael D. Doyle, Ph.D.: Digital Libraries Proposal, 01/07/94, page 2 of 6 definition of IRV Server).

Accordingly, claims 1 and 44 are deemed allowable over the prior art.

Claims 2-5, 10-14, 24-27, 45-28, and 55 are rejected over Vetter in view of Hansen, and further in view of Filepp et al.

With regard to claim 2, the invention of claim 1 is further limited so that the executable application is a controllable application and includes the step of controlling the controllable application from said client workstation via communications sent over the distributed hypermedia network.

The disclosure of Vetter and Hansen is described above.

Filepp discloses the Prodigy system where a plurality of PCs are connected to a host server over the telephone lines and run software to receive and transmit data between the host and the PCs.

The examiner states that Vetter disclosed that Mosaic and the WWW currently lack direct support for application-specific data and support for controlling the presentation of nontext data and suggested letting other applications control Mosaic display remotely. This rejection is respectfully traversed for the following reasons.

As described above, the present invention was reduced to practice prior to the publication of Vetter and prior to the incorporation of the CCI in Mosaic. Accordingly, claim 2 is allowable because it depends on claim 1 which is allowable.

The disclosure of Filepp et al. is unrelated to Mosaic or the WWW and merely discloses a host/terminal network communication system.

The rejection of claims 3-5, 10-14, 24-27, 45-48, and 55 all rely on the teaching of Vetter and the feature of Mosaic of letting other applications control its display remotely. Because the present invention was reduced to practice prior to the publication of Vetter and the incorporation of the CCI in Mosaic, these claims are allowable over the prior art.

In view of the foregoing, Applicants believe all claims now pending in this application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

PATENT

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at (415) 576-0200.

Respectfully submitted,

Charles E. Krueger
Reg. No. 30,077

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
(415) 576-0200
Fax (415) 576-0300
CEK:db

s:\02307I\553\amend.02

Attachment I

NSF/ARPA/NASA Digital Libraries RFP Response

Title: A Knowledge Managment Environment through the World Wide Web

Principal Investigator: *Michael D. Doyle, Ph.D., UCSF Library and Center for Knowledge Managment*

Specific Aims:

1) To develop a prototype knowledge management environment for the biomedical sciences which integrates access to online representations of the scientific literature, bibliographic databases, high-performance visualization technologies, large-scale scientific databases, and tools for authoring new-generation scientific publications.

1.a) To explore and evaluate the applicability of these tools in the areas of radiology and developmental & molecular biology.

2) To provide a means for relating digital forms of spatial, functional, and conceptual information as a basis for linking the biomedical scientific literature, through the Red Sage electronic journals project, to data resources provided through the Visible Human Project, The Human Brain Project, The Visible Embryo Project, The Human Genome Project, The Protein Database, and other large-scale biomolecular and biostructural databases.

2.a) To exploit these linking strategies in the creation of a set of integrated semi-automatic front ends to varied scientific databases accessible through the Internet.

2.b) To incorporate these linking methodologies into interactive authoring and editorial tools, allowing the creation of online publications that can embed visualizations and simulations which draw data from these Internet-accessible scientific databases.

3) To develop tools which provide access to interactive visualization and analysis of massive biomedical datasets through the Internet's World Wide Web distributed hypermedia network.

3.a) To refine and extend our existing algorithms enabling distributed visualization and analysis software "engines" which can be efficiently accessed by remote users through the Internet.

3.b) To refine and extend our existing algorithms to allow the display and real-time interactive control of three-and four-dimensional data visualization and analysis tools within hypermedia documents viewed using NCSA's Mosaic graphical browser to the World Wide Web.

3.c) To develop algorithms which use novel compression technologies for the optimized interactive remote control of computationally-intensive graphical applications through the Internet.

3.d) To integrate a,b & c into a system which allows real-time remote access to distributed parallel computational applications for visualization and analysis resources within a distributed hypermedia environment.

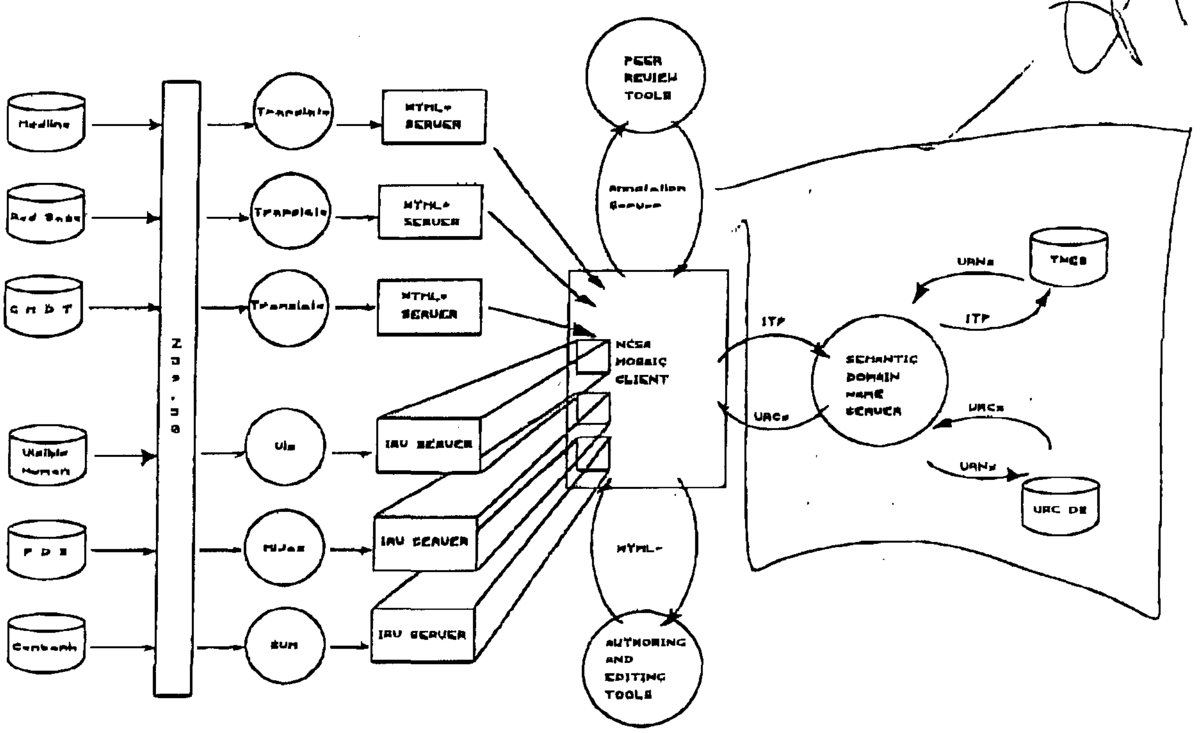
4) To explore extensions of the paradigm of scientific publishing which are made possible through use of current multimedia technologies in a networked environment, including:

4.a) publishing multidimensional datasets integrated with articles, eg: MRI and molecular data, perferred views, animations, interactive visualizations, interactive mathematical models, and

4.b) development of scientific authoring tools for publications which exist only in the networked environment.

4.b.1) This will include integration of HTML+ WYSIWYG authorial and editorial tools, multidimensional data visualization applications, molecular modelling and database management tools into an interactive scientific publishing environment.

System Diagram:



dPAC

Definitions:

HTML+: Hypertext Mark-up Language -- This is the language that World Wide Web databases are encoded in, and that Mosaic interprets.

IRV Server: UCSF CKM's Interactive Remote Visualization Server -- This allows interactive real-time visualization tools to be embedded into Mosaic documents.

Vis: UCSF CKM's distributed remote volume visualization tool

Midas: UCSF CGL's molecular visualization package

SVM: Sequence Visualization Module -- An as-yet unamed tool for graphical display of genetic sequence data.

ITP: Informal Text Phrase -- A user-entered search term, or a word or phrase of text that the user highlights from within a document.

URN: Universal Resource Name -- A persistent, location-independent identifier for an object.

URL: Universal Resource Location -- The address of an object. It contains enough information to identify a communications protocol and retrieve the object.

URC: Universal Resource Characteristics -- Any combination of one or more URNs or URLs with meta information (e.g. author, format, compression method).

Key

Description:

The system will draw from of a number of fundamental databases including bibliographic data (Medline) in the form of MARC records, journal publication data (Red Sage) in the form of SGML header and Postscript files, encyclopedic reference text data (CMDT) stored in an object-oriented SGML database, volumetric anatomical data (Visible Human Project) stored as NCSA HDF datasets, protein structure data (Protein Data Bank) stored as PDB files, and genetic sequence data (Genbank) stored as compressed ASCII strings (?, I'm guessing about Genbank).

These databases will reside behind a Z39.50 interface layer which yeilds, to the requesting client, the respective dataset in its native form. This data then goes through a translation layer where the data is either translated directly into HTML+ (Medline, Red Sage, CMDT) or loaded into a native-data visualization tool (Visible Human, PDB, Genbank). The HTML+ code is then passed to a set of HTML+ servers, which can be browsed by the Mosaic client. The visualization data is handled differently. The graphical I/O of the relevant visualization tool is passed to an interactive remote visualization (IRV) server, which handles both mapping of the display output from the visualization tool onto embedded live-visualization windows within the Mosaic-browsable HTML+ documents, as well as capture of user-entered mouse and keyboard events within the visualization windows and transmission of those mouse and keyboard events back to the relevant visualization tools. The user, browsing the system with the project's enhanced version of the Mosaic client, is presented with data and visualizations derived from these various databases, yet embedded into coherent, multimedia Mosaic documents.

For multimedia documents that have been explicitly pre-composed, the linking of these various data resources can take the form of universal resource names (URNs) that are encoded.as tags into the HTML+ documents. This is passed to the system's semantic domain name server, for resolution of the information object's location and retrieval means. The URNs are used as indices in order to look up the relevant universal resource characteristics (URCs) in a URC database, which yeilds the universal resource location (URL), or physical adress, of the information object in question.

Semi-automatic means will be provided for a user to search for arbitrary information objects on the system by either keying in a search word or phrase, or by highlighting a not-already-hyperlinked section of text that (s)he happens to be viewing within the Mosaic client at the time. This informal text phrase (ITP) is then passed to the semantic domain name server, which passes it on to a universal resource thesaurus (which will incorporate elements of the NLM's UMLS system). The thesaurus compares the ITP to its database of terms and phrases and returns a rank-ordered list of URNs that are likely to match the object in question. These URNs are then passed to the URC database for resolution of URLs that point to information objects on the Internet that are most likely to match the ITP that the user employed to initiate the search. The user is presented with a rank ordered set of textual descriptions of likely matches which are hyperlinked, via their URLs, to the data in question. Clicking upon a selection from this list loads the related data into the relevant visualization server (IRV) or HTML+ server, and a second Mosaic window pops up to allow viewing or interaction with that dataset.

A set of authoring and editing tools will be designed to allow the interactive WYSIWYG creation of HTML+ documents, as well as allowing the embedding of visualizations, etc., which can be created using the interactive remote visualization tools, and which can use data from the various scientific databases mentioned above. Alternatively, the author can use his/her own datasets, which would be uploaded to an Internet-accesible World Wide Web server. The journal editor can use the same set of tools to edit submitted articles and to communicate changes to the text with the author. This, of course, would occur in a private, access-controlled, area of the system, so that confidentiality of the material to be published can be controlled.

Other private, access-controlled HTML+ servers will be used to administer the peer review process. A modification of NCSA's Mosaic-based group annotation server will be developed to allow the jounal editor to exercise precise control and documentation of each reviewer's comments and suggestions.

Contributions:

UCSF CKM:

- Development of Z39.50-compliant experimental (subset) databases for storage of Visible Human data, PDB data, and Genbank sequence data.

- Cooperation with AT&T in the development of an object-oriented SGML-based database for the Handbook of Current Medical Diagnosis and Treatment (CMDT)

- Development of an experimental Z39.50 interface to Medline data (will be unnecessary if UC's DLA can provide such an interface to Melvyl Medline early enough into the project timeline)

- Development of translator servers to translate Medline MARC records, CMDT SGML data and Red Sage SGML/Postscript data into HTML+

- Development of a set of HTML+ documents that act as browsers to Medline, CMDT, and Red Sage

- Refinement and further development of Vis to allow better distribution of computation and better integration with Mosaic.

- Cooperation with CGL to adapt Midas for integration within Mosaic, and to identify and adapt a suitable program for graphical display of genetic sequence data.

- Refinement and further development of the interactive remote visualization server, and its incorporation (with NCSA's help) within the Mosaic environment.

- Development, in cooperation with NCSA, of an enhanced version of the Mosaic client to allow easier integration of external programs within Mosaic-readable documents.

- Development, in cooperation with Springer-Verlag and NCSA, of an interactive WYSIWYG editor for creation of HTML+ documents, and for embedding visualizations created using CKM's IRV tools, as well as development of a modified version of NCSA's group annotation server to support the peer review process.

- Development, in cooperation with AT&T, of an object-oriented SGML-based URC database

- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.

- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and Springer-Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

UCSF CGL:

- Cooperation with CKM to adapt Midas for integration within Mosaic, and to identify and adapt a suitable program for graphical display of genetic sequence data.

- Contributing to the refinement and further development of the interactive remote visualization server, and its incorporation (with NCSA's help) within the Mosaic environment.

- Development, in cooperation with UCSF's CKM, UCSF's Radiology Dept., Washington Univ., and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.
- Development, in cooperation with UCSF's CKM, UCSF's Radiology Dept., Washington Univ., and Springer-Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

Washington University:

- Development, in cooperation with UCSF's CKM, UCSF's CGL, and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.
- Development, in cooperation with UCSF's CKM, and UCSF's CGL., and Springer Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

AT&T Bell Laboratories:

- Development of Z39.50 interface to the RightPages server..
- Cooperation with CKM in the development of an object-oriented SGML-based database for the Handbook of Current Medical Diagnosis and Treatment (CMDT)
- Development, in cooperation with CKM, of an object-oriented SGML-based URC database
- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and CKM, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.

Springer-Verlag:

- Development, in cooperation with UCSF's CKM and NCSA, of an interactive WYSIWYG editor for creation of HTML+ documents, and for embedding visualizations created using CKM's IRV tools, as well as development of a modified version of NCSA's group annotation server to support the peer review process.
- Development, in cooperation with UCSF's CKM, and UCSF's CGL., and Washington Univ. of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

NCSA:

- Cooperation with CKM in developing an enhanced version of Mosaic to allow easier integration of a client modeule for CKM's interactive remote visualization server.
- Cooperation with CKM and Springer-Verlag in the modification of NCSA's group annotation server to facilitate the peer-review process.

Personnel:

Co-Investigators:

UCSF:

Library & CKM: Richard Lucier, David Martin, Zoe Stavri, Ph.D., Cheong Ang, Marc Salomon
Radiology: Tom Budinger, Ph.D.
Molecular & developmental Biology: Tom Ferrin, Ph.D., Charles Ordahl, PhD.

Washington University (molecular biology): Toni Kazic, PhD
Bell Laboratories: Ed Szurkowski, Guy Story
Springer Verlag: Bob Badger, PhD
NCSA: Joseph Hardin, PhD, & Mosaic development group
SFSU: Computer Science Dept. MS students

Timetable: 4 years

Budget: 1.2 $M/year

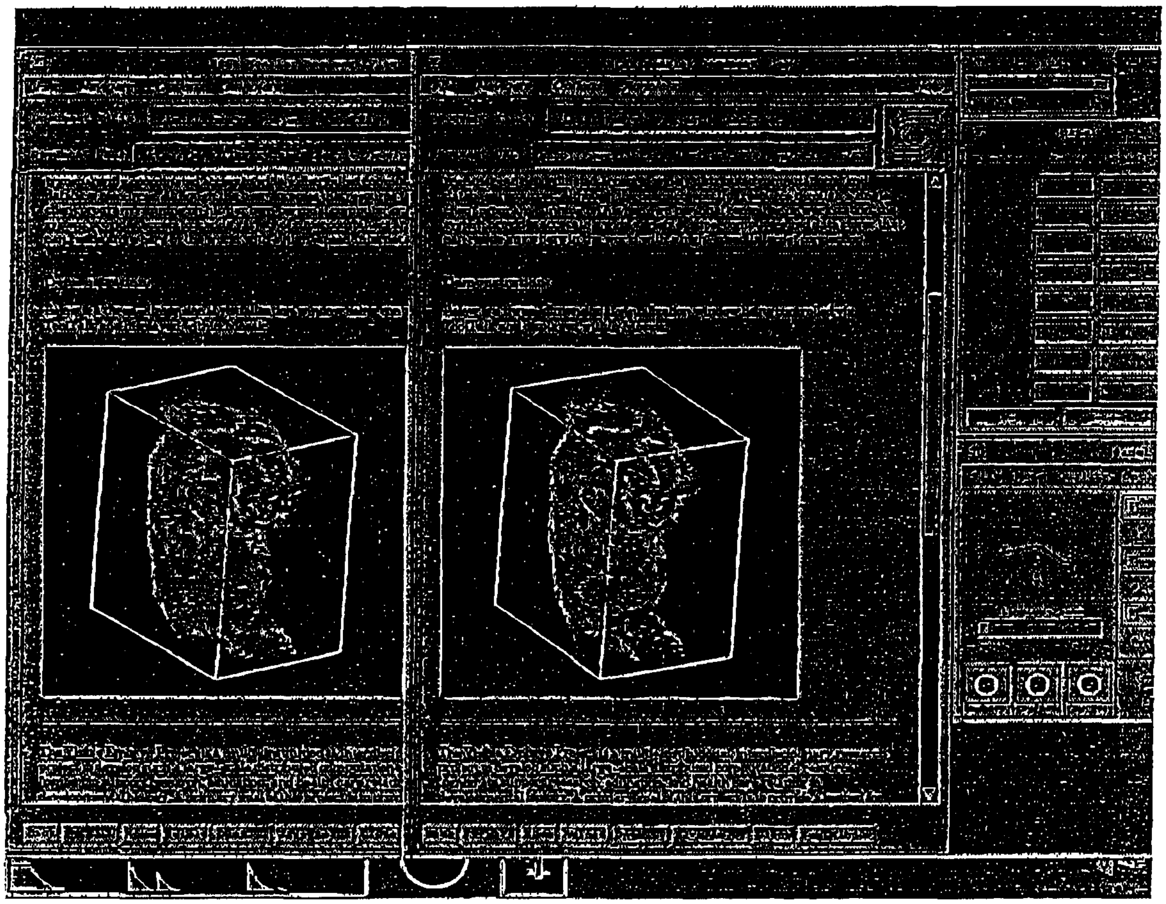


Figure 1: A stereo-pair illustration of interactive real-time 3-dimensional human embryonic volume reconstructions embedded within an NCSA Mosaic document. This technology was developed by the Center for Knowledge Management at the University of California, San Francisco, and was demonstrated there in November, 1993.

JOSEPH HARDIN

National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign
Champaign, IL 61820
217/244-6095
email: hardin@ncsa.uiuc.edu

Major work experience

University of Illinois,
Associate Director, Software Development Group, NCSA, 1992-present

University of Illinois,
Manager, Software Development Group, NCSA, 1988-1992
Coordinator, Academic Affiliates Program, NCSA, 1987 -1988
Visiting Research Associate, NCSA, 1986-1987

University of Georgia at Athens,
Visiting Instructor, Department of Speech Communication, 1985-1986

University of Illinois,
Teaching Assistant, Department of Speech Communication, 1979-1985
Teaching Assistant, Department of English-Rhetoric, 1978
Research Assistant, Department of Sociology, 1977-1978

Other work

Consultant -Business computer communications, database management, and office organization, 1982-1985

Consulting Editor/Contributor-The Champaign-Urbana Weekly news magazine, 1981-1982

Editor-The Champaign-Urbana Weekly news magazine, 1979-1980

Recent Grants and Awards

Exploratory Research and Initial Development of Software for the Analysis of Multiple Hybridization Images, 1990-1992
IBM Second Generation RISL Workstation Graphics Capabilities: Jointly Defined Effort, 1990-1992
NCSA Hierarchical Data File Software Capitalization: National Distribution and Support, 1991-1993
Research Education for Undergraduate Students Supplement, 1990-1993
Supercomputers for Biologists: Macromolecular Sequence Analysis through Distributed Computing, 1991-1993
Visualization in the MS DOS Environment: NCSA/Jackson State Collaborative Project, 1990-1993
x3d Program Development, 1992-1993

Education

Study toward Ph.D., Speech Communication, University of Illinois
Study Abroad, University of Cologne, Cologne, West Germany, 1978
B.A., History, University of Illinois, 1972

Selected Recent Conference Papers/Participation

Sigchi, Session Paper, "Scientific Visualization in a Collaborative Environment," Amsterdam, Holland, April, 1993 (in submission)

Sigchi, Session Paper, "Collaborative Hypermedia for Computational Science," Amsterdam, Holland, April, 1993 (in submission)

Oceanographic Institute, Old Dominion University, Invited Seminar, "Recent Developments in Collaboration Technologies," Norfolk, Va., Feb. 1993

Scientific Computing & Automation Conference, Invited Speaker, "Cross platform Digital Conferencing Software Development," Washington, DC, 1992

New York University, Academic Computing Facility, "Scientific Visualization in Collaborative Technologies," New York City, NY, 1992

International Institute of Ecological Economics Conference, Invited Workshop Participant, Beijer Institute, Stockholm, Sweden, 1992

American Educational Research Association Conference, "Collaborative Tools for Scientific Communication and Understanding," San Francisco, CA, 1992

Mac SciTech National Conference'92, "Interpersonal Computing for Computational Scientists: The NCSA Collage Series," San Francisco, CA, 1992

Atty Docket No. 02307I-553

PTO FAX NO.: 1-703-308-5359

ATTENTION: Examiner D. Dinh
Group Art Unit 2317

OFFICIAL COMMUNICATION

FOR THE PERSONAL ATTENTION OF

EXAMINER D. DINH

CERTIFICATION OF FACSIMILE TRANSMISSION

I hereby certify that the following Response After Final and Declaration of Michael D. Doyle Under Rule 131, in re Application of Doyle et al., Serial No. 08/324,443, filed October 17, 1994, for EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS is being facsimile transmitted to the Patent and Trademark Office on the date shown below.

Number of pages being transmitted, including this page: 26

Dated: January 8, 1997 Irene Rodas

PLEASE CONFIRM RECEIPT OF THIS PAPER BY RETURN FACSIMILE AT (415) 576-0300

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, CA 94111-3834
Telephone: (415) 576-0200
Fax: (415) 576-0300
S:\02307I\553\FAXTRNS.PTO

Copy

Atty Docket No. 02307I-553

PTO FAX NO.: 1-703-308-5359

ATTENTION: Examiner D. Dinh
Group Art Unit 2317

OFFICIAL COMMUNICATION

FOR THE PERSONAL ATTENTION OF

EXAMINER D. DINH

CERTIFICATION OF FACSIMILE TRANSMISSION

I hereby certify that the following Response After Final and Declaration of Michael D. Doyle Under Rule 131, in re Application of Doyle et al., Serial No. 08/324,443, filed October 17, 1994, for EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS is being facsimile transmitted to the Patent and Trademark Office on the date shown below.

Number of pages being transmitted, including this page: 26

Dated: January 8, 1997 Irene Rodas

PLEASE CONFIRM RECEIPT OF THIS PAPER BY RETURN FACSIMILE AT (415) 576-0300

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, CA 94111-3834
Telephone: (415) 576-0200
Fax: (415) 576-0300
S:\02307I\553\FAXTRNS.PTO

I hereby certify that this correspondence is being sent by facsimile transmission to: D. Dinh
Fax No.: 1-703-308-5359
Assistant Commissioner for Patents,
Washington, D.C. 20231,
on 1-8-97

TOWNSEND and TOWNSEND and CREW LLP

By Irene Rodas

PATENT

Attorney Docket No. 02307I-553

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:)
)
MICHAEL D. DOYLE et al.) Examiner: D. Dinh
)
Application No.: 08/324,443) Art Unit: 2317
)
Filed: 10/17/94) RESPONSE AFTER FINAL
)
For: EMBEDDED PROGRAM OBJECTS IN)
DISTRIBUTED HYPERMEDIA)
SYSTEMS)
)

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

The following is responsive to the Office Action mailed December 13, 1996:

REMARKS

Claims 1-15 and 17-56 are now pending in the application. Reexamination and reconsideration are requested.

Claims 1 and 44 are rejected under 35 U.S.C. §103 as being unpatentable over Vetter "Mosaic and the World-Wide Web" and further in view of Hansen "Andrew as a Multiparadigm Environment for Visual Languages."

In claim 1, the distributed hypermedia document includes an embed text format that specifies the location of an object external to the distributed hypermedia document and that specifies type information utilized by the browser to identify and locate an executable application external to the distributed hypermedia document. The browser invokes the executable application to display and process the object within the browser window.

The Vetter reference states that "Users can extend Mosaic's functionality by creating custom servers and letting other applications control its display remotely."

The Hansen reference discloses a multiparadigm language environment for visual languages that supports not only graphical interaction but also file storage, printing, compilation, and execution. It is suggested that each sublanguage element be displayed in the same window.

The examiner states that Vetter discloses that Mosaic's functionality can be extended by having custom servers and letting other applications control its display remotely. Hence, it would have been obvious to extend Mosaic's functionality to enable external application to display and process the object within the browser-controlled window because it would have improved the system by reducing clustering of the display and aiding the reader comprehension of the hypermedia document. This rejection is respectfully traversed for the following reasons.

A Rule 131 Declaration is submitted showing that applicant's invention was reduced to practice prior to the publication of Vetter. Accordingly, the feature of controlling Mosaic's display remotely is not disclosed in the prior art.

Further, the feature described in Vetter was not present in Mosaic prior to the reduction to practice of the invention. The feature of remotely controlling Mosaic's display, the common-client interface (CCI), was incorporated into Mosaic based on the work done by the applicants as evidenced by Attachment I, a proposal to the NCSA, the developers of Mosaic, to modify Mosaic to allow real-time visualization tools to be embedded in Mosaic. (Michael D. Doyle, Ph.D.: Digital Libraries Proposal, 01/07/94, page 2 of 6 definition of IRV Server).

Accordingly, claims 1 and 44 are deemed allowable over the prior art.

Claims 2-5, 10-14, 24-27, 45-28, and 55 are rejected over Vetter in view of Hansen, and further in view of Filepp et al.

With regard to claim 2, the invention of claim 1 is further limited so that the executable application is a controllable application and includes the step of controlling the controllable application from said client workstation via communications sent over the distributed hypermedia network.

The disclosure of Vetter and Hansen is described above.

Filepp discloses the Prodigy system where a plurality of PCs are connected to a host server over the telephone lines and run software to receive and transmit data between the host and the PCs.

The examiner states that Vetter disclosed that Mosaic and the WWW currently lack direct support for application-specific data and support for controlling the presentation of nontext data and suggested letting other applications control Mosaic display remotely. This rejection is respectfully traversed for the following reasons.

As described above, the present invention was reduced to practice prior to the publication of Vetter and prior to the incorporation of the CCI in Mosaic. Accordingly, claim 2 is allowable because it depends on claim 1 which is allowable.

The disclosure of Filepp et al. is unrelated to Mosaic or the WWW and merely discloses a host/terminal network communication system.

The rejection of claims 3-5, 10-14, 24-27, 45-48, and 55 all rely on the teaching of Vetter and the feature of Mosaic of letting other applications control its display remotely. Because the present invention was reduced to practice prior to the publication of Vetter and the incorporation of the CCI in Mosaic, these claims are allowable over the prior art.

In view of the foregoing, Applicants believe all claims now pending in this application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at (415) 576-0200.

Respectfully submitted,

Charles E. Krueger
Reg. No. 30,077

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
(415) 576-0200
Fax (415) 576-0300
CEK:db

s:\02307I\553\amend.02

Attachment I

NSF/ARPA/NASA Digital Libraries RFP Response

Title: A Knowledge Managment Environment through the World Wide Web

Principal Investigator: Michael D. Doyle, Ph.D., UCSF Library and Center for Knowledge Managment

Specific Aims:

1) To develop a prototype knowledge management environment for the biomedical sciences which integrates access to online representations of the scientific literature, bibliographic databases, high-performance visualization technologies, large-scale scientific databases, and tools for authoring new-generation scientific publications.

1.a) To explore and evaluate the applicability of these tools in the areas of radiology and developmental & molecular biology.

2) To provide a means for relating digital forms of spatial, functional, and conceptual information as a basis for linking the biomedical scientific literature, through the Red Sage electronic journals project, to data resources provided through the Visible Human Project, The Human Brain Project, The Visible Embryo Project, The Human Genome Project, The Protein Database, and other large-scale biomolecular and biostructural databases.

2.a) To exploit these linking strategies in the creation of a set of integrated semi-automatic front ends to varied scientific databases accessible through the Internet.

2.b) To incorporate these linking methodologies into interactive authoring and editorial tools, allowing the creation of online publications that can embed visualizations and simulations which draw data from these Internet-accessible scientific databases.

3) To develop tools which provide access to interactive visualization and analysis of massive biomedical datasets through the Internet's World Wide Web distributed hypermedia network.

3.a) To refine and extend our existing algorithms enabling distributed visualization and analysis software "engines" which can be efficiently accessed by remote users through the Internet.

3.b) To refine and extend our existing algorithms to allow the display and real-time interactive control of three-and four-dimensional data visualization and analysis tools within hypermedia documents viewed using NCSA's Mosaic graphical browser to the World Wide Web.

3.c) To develop algorithms which use novel compression technologies for the optimized interactive remote control of computationally-intensive graphical applications through the Internet.

3.d) To integrate a,b & c into a system which allows real-time remote access to distributed parallel computational applications for visualization and analysis resources within a distributed hypermedia environment.

4) To explore extensions of the paradigm of scientific publishing which are made possible through use of current multimedia technologies in a networked environment, including:

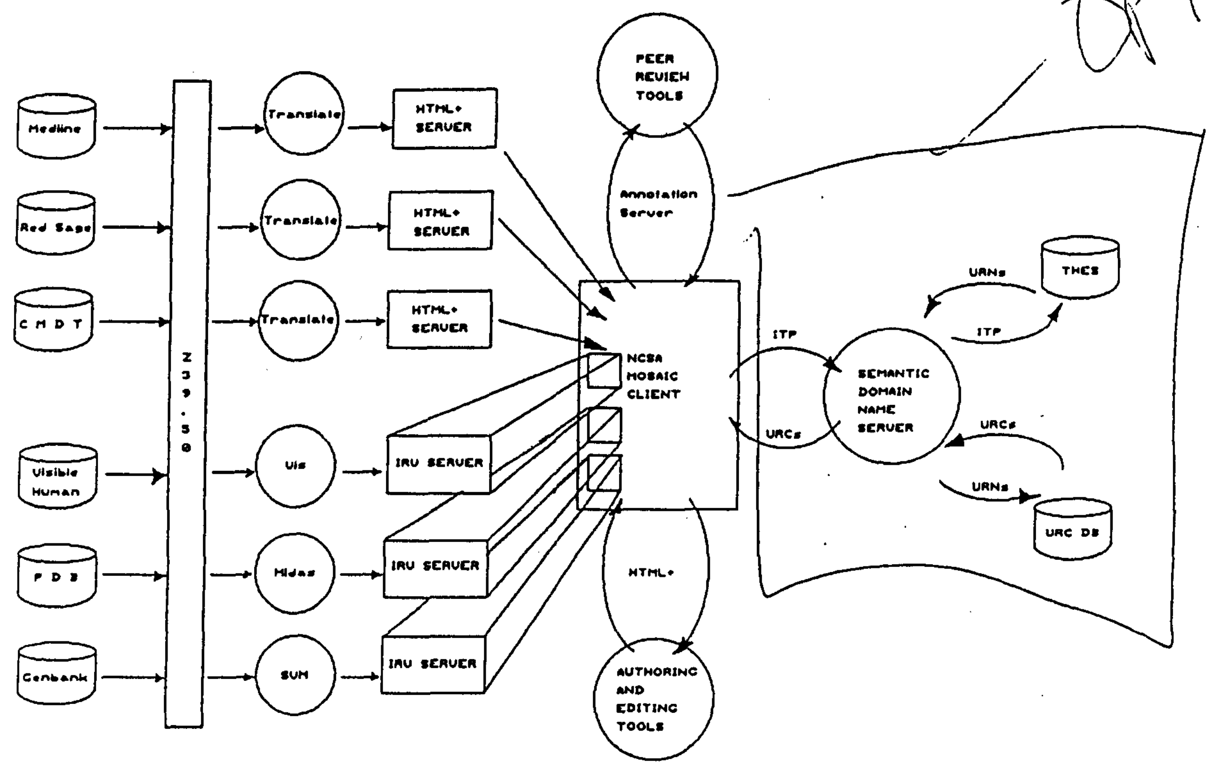
4.a) publishing multidimensional datasets integrated with articles, eg: MRI and molecular data, perferred views, animations, interactive visualizations, interactive mathematical models, and

4.b) development of scientific authoring tools for publications which exist only in the networked environment.

4.b.1) This will include integration of HTML+ WYSIWYG authorial and editorial tools, multidimensional data visualization applications, molecular modelling and database management tools into an interactive scientific publishing environment.

System Diagram:

OPAC



Definitions:

HTML+: Hypertext Mark-up Language -- This is the language that World Wide Web databases are encoded in, and that Mosaic interprets.

IRV Server: UCSF CKM's Interactive Remote Visualization Server -- This allows interactive real-time visualization tools to be embedded into Mosaic documents.

Vis: UCSF CKM's distributed remote volume visualization tool

Midas: UCSF CGL's molecular visualization package

SVM: Sequence Visualization Module -- An as-yet unamed tool for graphical display of genetic sequence data.

ITP: Informal Text Phrase -- A user-entered search term, or a word or phrase of text that the user highlights from within a document.

URN: Universal Resource Name -- A persistent, location-independent identifier for an object.

URL: Universal Resource Location -- The address of an object. It contains enough information to identify a communications protocol and retrieve the object.

URC: Universal Resource Characteristics -- Any combination of one or more URNs or URLs with meta information (e.g. author, format, compression method).

Key

Description:

The system will draw from of a number of fundamental databases including bibliographic data (Medline) in the form of MARC records, journal publication data (Red Sage) in the form of SGML header and Postscript files, encyclopedic reference text data (CMDT) stored in an object-oriented SGML database, volumetric anatomical data (Visible Human Project) stored as NCSA HDF datasets, protein structure data (Protein Data Bank) stored as PDB files, and genetic sequence data (Genbank) stored as compressed ASCII strings (?, I'm guessing about Genbank).

These databases will reside behind a Z39.50 interface layer which yeilds, to the requesting client, the respective dataset in its native form. This data then goes through a translation layer where the data is either translated directly into HTML+ (Medline, Red Sage, CMDT) or loaded into a native-data visualization tool (Visible Human, PDB, Genbank). The HTML+ code is then passed to a set of HTML+ servers, which can be browsed by the Mosaic client. The visualization data is handled differently. The graphical I/O of the relevant visualization tool is passed to an interactive remote visualization (IRV) server, which handles both mapping of the display output from the visualization tool onto embedded live-visualization windows within the Mosaic-browsable HTML+ documents, as well as capture of user-entered mouse and keyboard events within the visualization windows and transmission of those mouse and keyboard events back to the relevant visualization tools. The user, browsing the system with the project's enhanced version of the Mosaic client, is presented with data and visualizations derived from these various databases, yet embedded into coherent, multimedia Mosaic documents.

For multimedia documents that have been explicitly pre-composed, the linking of these various data resources can take the form of universal resource names (URNs) that are encoded as tags into the HTML+ documents. This is passed to the system's semantic domain name server, for resolution of the information object's location and retrieval means. The URNs are used as indices in order to look up the relevant universal resource characteristics (URCs) in a URC database, which yeilds the universal resource location (URL), or physical adress, of the information object in question.

Semi-automatic means will be provided for a user to search for arbitrary information objects on the system by either keying in a search word or phrase, or by highlighting a not-already-hyperlinked section of text that (s)he happens to be viewing within the Mosaic client at the time. This informal text phrase (ITP) is then passed to the semantic domain name server, which passes it on to a universal resource thesaurus (which will incorporate elements of the NLM's UMLS system). The thesaurus compares the ITP to its database of terms and phrases and returns a rank-ordered list of URNs that are likely to match the object in question. These URNs are then passed to the URC database for resolution of URLs that point to information objects on the Internet that are most likely to match the ITP that the user employed to initiate the search. The user is presented with a rank ordered set of textual descriptions of likely matches which are hyperlinked, via their URLs, to the data in question. Clicking upon a selection from this list loads the related data into the relevant visualization server (IRV) or HTML+ server, and a second Mosaic window pops up to allow viewing or interaction with that dataset.

A set of authoring and editing tools will be designed to allow the interactive WYSIWYG creation of HTML+ documents, as well as allowing the embedding of visualizations, etc., which can be created using the interactive remote visualization tools, and which can use data from the various scientific databases mentioned above. Alternatively, the author can use his/her own datasets, which would be uploaded to an Internet-accesible World Wide Web server. The journal editor can use the same set of tools to edit submitted articles and to communicate changes to the text with the author. This, of course, would occur in a private, access-controlled, area of the system, so that confidentiality of the material to be published can be controlled.

Other private, access-controlled HTML+ servers will be used to administer the peer review process. A modification of NCSA's Mosaic-based group annotation server will be developed to allow the jounal editor to exercise precise control and documentation of each reviewer's comments and suggestions.

Contributions:

UCSF CKM:

- Development of Z39.50-compliant experimental (subset) databases for storage of Visible Human data, PDB data, and Genbank sequence data.
- Cooperation with AT&T in the development of an object-oriented SGML-based database for the Handbook of Current Medical Diagnosis and Treatment (CMDT)
- Development of an experimental Z39.50 interface to Medline data (will be unnecessary if UC's DLA can provide such an interface to Melvyl Medline early enough into the project timeline)
- Development of translator servers to translate Medline MARC records, CMDT SGML data and Red Sage SGML/Postscript data into HTML+
- Development of a set of HTML+ documents that act as browsers to Medline, CMDT, and Red Sage
- Refinement and further development of Vis to allow better distribution of computation and better integration with Mosaic.
- Cooperation with CGL to adapt Midas for integration within Mosaic, and to identify and adapt a suitable program for graphical display of genetic sequence data.
- Refinement and further development of the interactive remote visualization server, and its incorporation (with NCSA's help) within the Mosaic environment.
- Development, in cooperation with NCSA, of an enhanced version of the Mosaic client to allow easier integration of external programs within Mosaic-readable documents.
- Development, in cooperation with Springer-Verlag and NCSA, of an interactive WYSIWYG editor for creation of HTML+ documents, and for embedding visualizations created using CKM's IRV tools, as well as development of a modified version of NCSA's group annotation server to support the peer review process.
- Development, in cooperation with AT&T, of an object-oriented SGML-based URC database
- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.
- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and Springer-Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

UCSF CGL:

- Cooperation with CKM to adapt Midas for integration within Mosaic, and to identify and adapt a suitable program for graphical display of genetic sequence data.
- Contributing to the refinement and further development of the interactive remote visualization server, and its incorporation (with NCSA's help) within the Mosaic environment.

- Development, in cooperation with UCSF's CKM, UCSF's Radiology Dept., Washington Univ., and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.
- Development, in cooperation with UCSF's CKM, UCSF's Radiology Dept., Washington Univ., and Springer-Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

Washington University:

- Development, in cooperation with UCSF's CKM, UCSF's CGL, and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.
- Development, in cooperation with UCSF's CKM, and UCSF's CGL., and Springer Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

AT&T Bell Laboratories:

- Development of Z39.50 interface to the RightPages server..
- Cooperation with CKM in the development of an object-oriented SGML-based database for the Handbook of Current Medical Diagnosis and Treatment (CMDT)
- Development, in cooperation with CKM, of an object-oriented SGML-based URC database
- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and CKM, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.

Springer-Verlag:

- Development, in cooperation with UCSF's CKM and NCSA, of an interactive WYSIWYG editor for creation of HTML+ documents, and for embedding visualizations created using CKM's IRV tools, as well as development of a modified version of NCSA's group annotation server to support the peer review process.
- Development, in cooperation with UCSF's CKM, and UCSF's CGL., and Washington Univ. of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

NCSA:

- Cooperation with CKM in developing an enhanced version of Mosaic to allow easier integration of a client modeule for CKM's interactive remote visualization server.
- Cooperation with CKM and Springer-Verlag in the modification of NCSA's group annotation server to facilitate the peer-review process.

Personnel:

Co-Investigators:

UCSF:

Library & CKM: Richard Lucier, David Martin, Zoe Stavri, Ph.D., Cheong Ang, Marc Salomon
Radiology: Tom Budinger, Ph.D.
Molecular & developmental Biology: Tom Ferrin, Ph.D., Charles Ordahl, PhD.

Washington University (molecular biology): Toni Kazic, PhD
Bell Laboratories: Ed Szurkowski, Guy Story
Springer Verlag: Bob Badger, PhD
NCSA: Joseph Hardin, PhD, & Mosaic development group
SFSU: Computer Science Dept. MS students

Timetable: 4 years

Budget: 1.2 $M/year

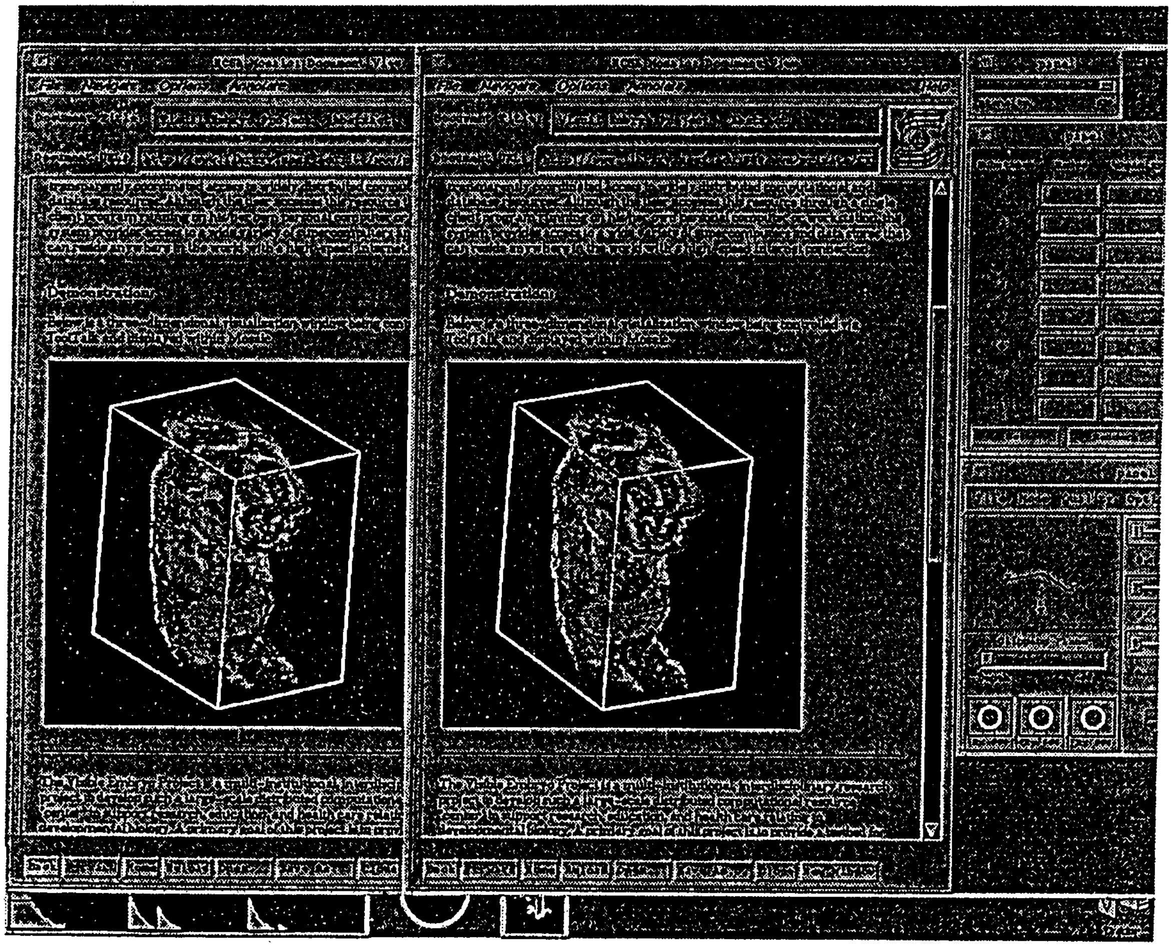


Figure 1: A stereo-pair illustration of interactive real-time 3-dimensional human embryonic volume reconstructions embedded within an NCSA Mosaic document. This technology was developed by the Center for Knowledge Management at the University of California, San Francisco, and was demonstrated there in November, 1993.

JOSEPH HARDIN

National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign
Champaign, IL 61820
217/244-6095
email: hardin@ncsa.uiuc.edu

Major work experience

University of Illinois,
Associate Director, Software Development Group, NCSA, 1992-present

University of Illinois,
Manager, Software Development Group, NCSA, 1988-1992
Coordinator, Academic Affiliates Program, NCSA, 1987 -1988
Visiting Research Associate, NCSA, 1986-1987

University of Georgia at Athens,
Visiting Instructor, Department of Speech Communication, 1985-1986

University of Illinois,
Teaching Assistant, Department of Speech Communication, 1979-1985
Teaching Assistant, Department of English-Rhetoric, 1978
Research Assistant, Department of Sociology, 1977-1978

Other work

Consultant -Business computer communications, database management, and office organization, 1982-1985

Consulting Editor/Contributor-The Champaign-Urbana Weekly news magazine, 1981-1982

Editor-The Champaign-Urbana Weekly news magazine, 1979-1980

Recent Grants and Awards

Exploratory Research and Initial Development of Software for the Analysis of Multiple Hybridization Images, 1990-1992
IBM Second Generation RISL Workstation Graphics Capabilities: Jointly Defined Effort, 1990-1992
NCSA Hierarchical Data File Software Capitalization: National Distribution and Support, 1991-1993
Research Education for Undergraduate Students Supplement, 1990-1993
Supercomputers for Biologists: Macromolecular Sequence Analysis through Distributed Computing, 1991-1993
Visualization in the MS DOS Environment: NCSA/Jackson State Collaborative Project, 1990-1993
x3d Program Development, 1992-1993

Education

Study toward Ph.D., Speech Communication, University of Illinois
Study Abroad, University of Cologne, Cologne, West Germany, 1978
B.A., History, University of Illinois, 1972

Selected Recent Conference Papers/Participation

Sigchi, Session Paper, "Scientific Visualization in a Collaborative Environment," Amsterdam, Holland, April, 1993 (in submission)

Sigchi, Session Paper, "Collaborative Hypermedia for Computational Science," Amsterdam, Holland, April, 1993 (in submission)

Oceanographic Institute, Old Dominion University, Invited Seminar, "Recent Developments in Collaboration Technologies," Norfolk, Va., Feb. 1993

Scientific Computing & Automation Conference, Invited Speaker, "Cross platform Digital Conferencing Software Development," Washington, DC, 1992

New York University, Academic Computing Facility, "Scientific Visualization in Collaborative Technologies," New York City, NY, 1992

International Institute of Ecological Economics Conference, Invited Workshop Participant, Beijer Institute, Stockholm, Sweden, 1992

American Educational Research Association Conference, "Collaborative Tools for Scientific Communication and Understanding," San Francisco, CA, 1992

Mac SciTech National Conference'92, "Interpersonal Computing for Computational Scientists: The NCSA Collage Series," San Francisco, CA, 1992

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in ~~an envelope addressed to:~~ Assistant Commissioner for Patents, By FAX 703-308-5359 Washington, D.C. 20231, on

1-8-97

TOWNSEND and TOWNSEND and CREW LLP

By

PATENT

Attorney Docket No. 023071-553

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:)

MICHAEL D. DOYLE et al.) Examiner: D. Dinh

Application No.: 08/324,443) Art Unit: 2317

Filed: 10/17/94) **DECLARATION OF MICHAEL D. DOYLE UNDER RULE 131**

For: EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS)

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

I, MICHAEL D. DOYLE, hereby declare that:

1. I am a co-inventor of the subject matter disclosed and claimed in U.S. Patent Application No. 08/324,443.

2. The subject matter claimed in the above patent application was reduced to practice in this country prior to October, 1994, the publication date of the cited reference entitled "Mosaic and the World-Wide Web" by Vetter.

3. The reduction to practice of the claimed invention is evidenced by ATTACHMENT A, which is a transcript of the audio portion and still photographs of a video tape presented to an audience of scientists prior to October, 1994.

4. As stated in ATTACHMENT A, starting at the bottom of page 2, interface and control software had been developed that allows the embedding of a visualization application within a Mosaic document. As is apparent from the photographs, the object is displayed and processed within the browser-controlled window. The visualization application is external to the hypermedia document displayed by the browser.

MICHAEL D. DOYLE et al. **PATENT**
Application No.: 08/324,443
Page 2

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Dated: 1-2-97, 1997.

MICHAEL D. DOYLE

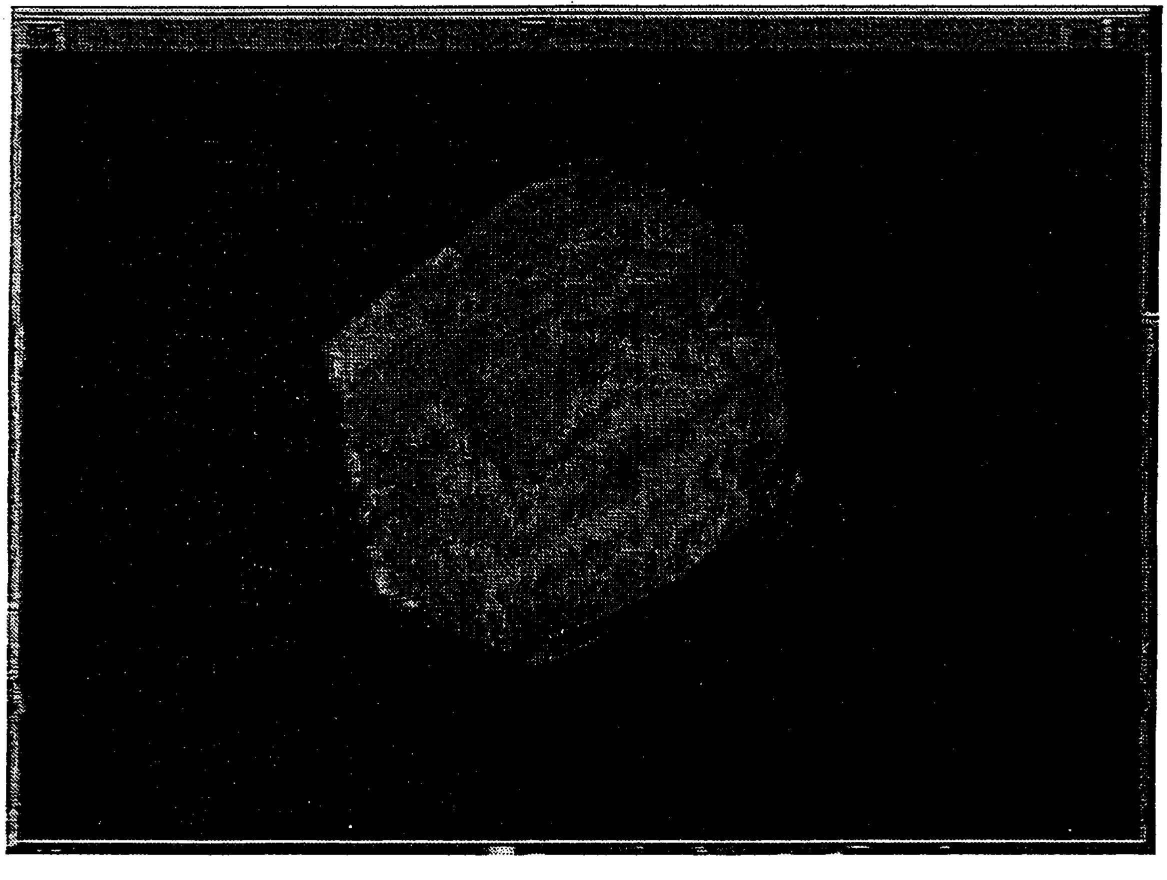
CEK:db
s:\02307I\553\DECL.01

M.D. Doyle, et. al., *The Virtual Embryo: VR Applications in Human Developmental Anatomy,* presented at "Medicine Meets Virtual Reality II, Interactive Technology and Healthcare: Visionary Applications for Simulation, Visualization & Robotics," sponsored by the UCSD School of Medicine and the Advanced Projects Research Agency, San Diego, CA, **January 27, 1994**.

Video Presentation Transcript:

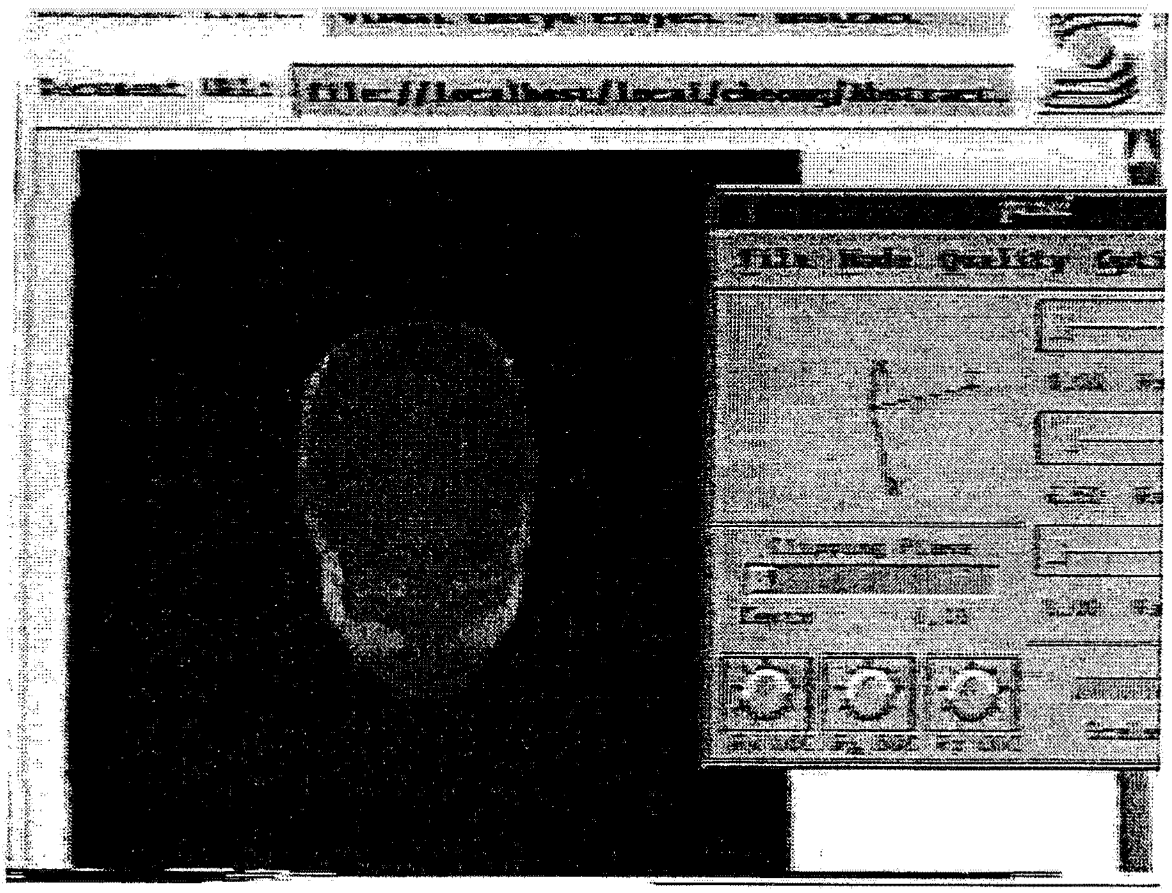
This is a status report of some of the work that's been accomplished during the first years of the Visible Embryo Project.

One of our first tasks was to develop some volume visualization software that we could use for imaging and analysis of the embryo reconstructions that we planned to create during the full term of the embryo project. One thing that was an absolute requirement was that this software be able to distribute its computational load across a network of graphics computers that weren't necessarily all in the same place. Basically we wanted to be able to have computers that could be all over the country connected by high -speed networking, able to contribute to a computation of three-dimensional datasets.



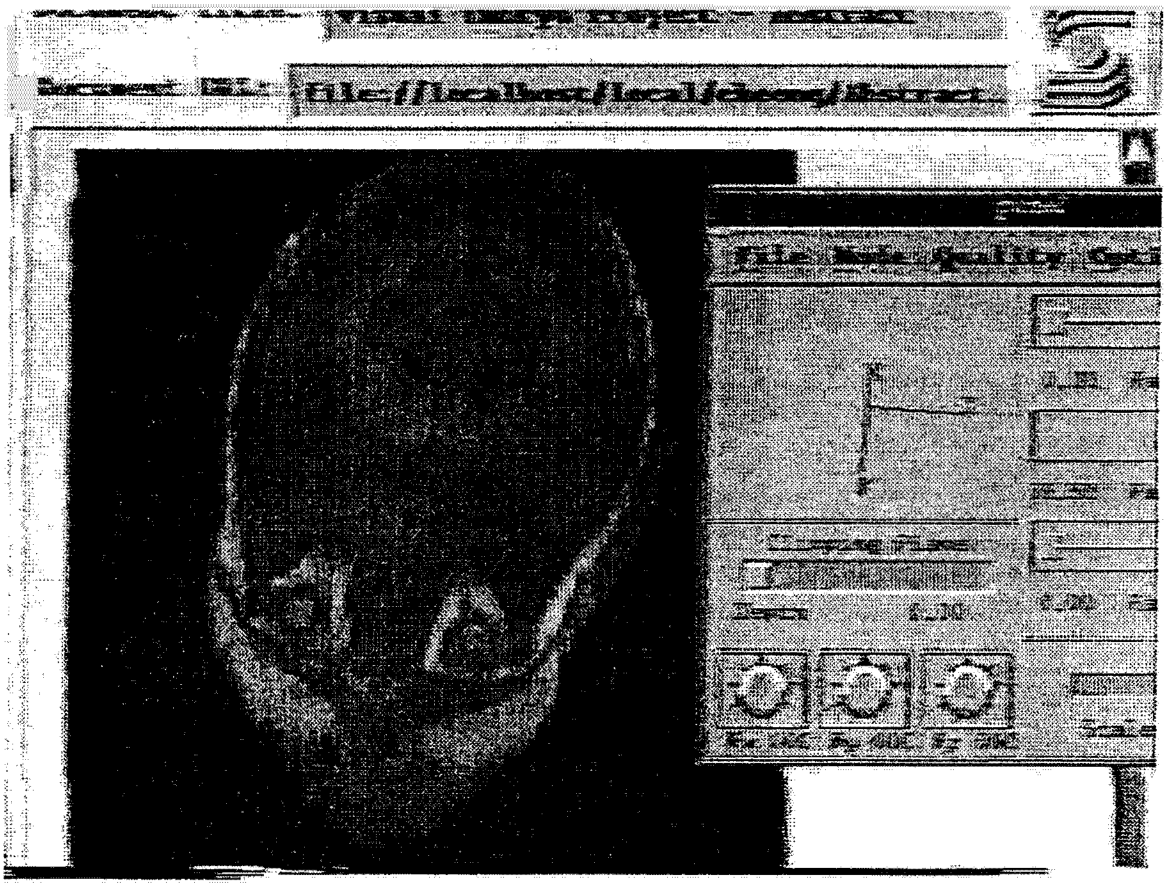
What you see here is a package called "VIS," which was developed in our group, for three-dimensional volume visualization. This is a very portable package that has been generated using as generic code as possible, although this particular image that you see here is running on a Silicon Graphics Reality Engine II, which is optimized for volume visualization. We need to use that kind of high-speed optimization to accomplish the real-time interactivity that we need to accomplish for this project. And as you see, as this rotates, and it starts rotating faster and faster, only a very powerful graphics - basically a graphics supercomputer - can accomplish this much computation on a three dimensional dataset. One of our goals is to allow anybody on the Internet with a very low-level access workstation to accomplish this kind of interactivity through their network connection, and the way that we do that is through a client-server architecture where we have a very powerful server computer accessed by a very low-end client machine.

We decided early on to use NCSA's Mosaic program and the World Wide Web to integrate access to this system. One problem with Mosaic is, as it exists today, is that the images within Mosaic are typically static or passive-playback images.



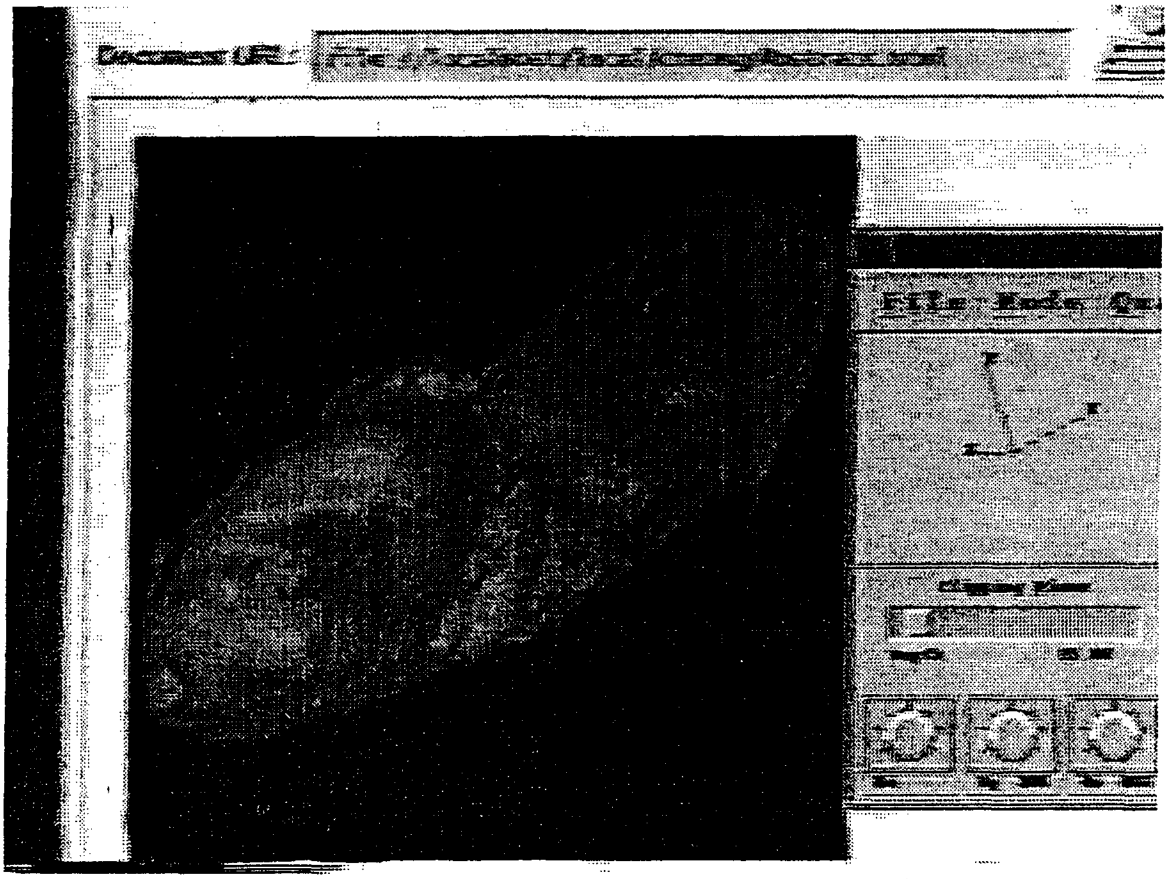
What you see here is an enhancement that we've created to the Mosaic interface and control software that allows the embedding of a dynamic real-time visualization application within a Mosaic document. You see a head, a volume MRI model of a human head, that's being rotated in real-time interactively by the viewer. You see a little panel to the right which is, it's a control panel that's popped up - it's really external to Mosaic itself but it can talk to the internal control programs that drive the Mosaic client - and allow you to interactively control the display from within Mosaic. This is actually controlling the volume visualization software that you just saw a few minutes ago.

By moving around the controls in the control panel we can do things like rotate, we can control slices through the dataset, and so on. As you can see, there's rotation in x, y, and z planes. We can also compute arbitrary oblique sections through the data and look at the internal anatomy. Here we can see the brain of this individual; we can rotate and view that section of the dataset from a variety of vantage points. There are zooming capabilities that allow us to zoom up on the data or zoom back to look at things in more or less detail, and you can see here that once zoomed, we're moving our cutting plane down through the dataset and looking at more and more inferior levels.



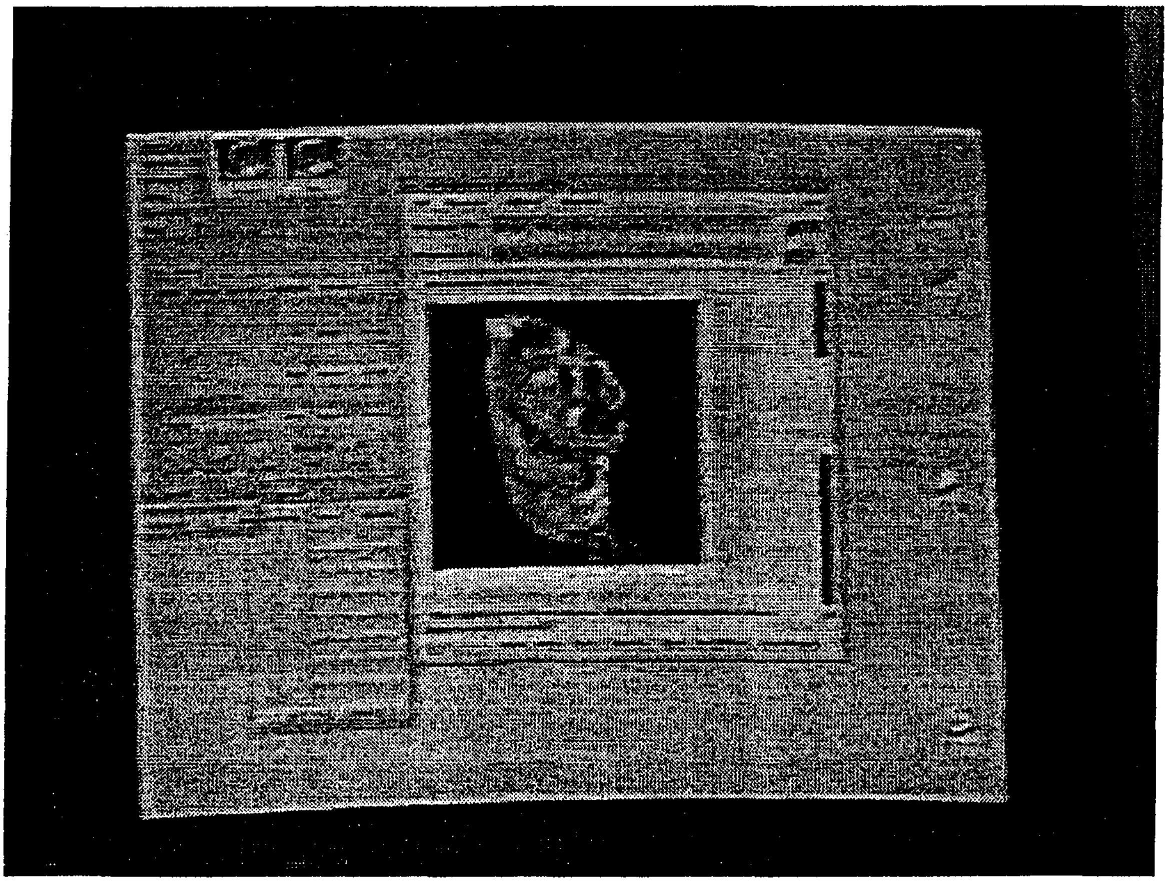
Normally graphics in Mosaic are static, as I said, but this embedding of graphic applications within Mosaic is really going to form the basis of how we integrate information access through the Visible Embryo Project. What you are about to see is our prototype system of a Visible Embryo Mosaic document that has embedded realtime visualizations within it.

We just loaded the Visible Embryo Mosaic page, the system is about to scroll down this page - it is an abstract about the Visible Embryo Project - and then we see a window. It looks like a static image just like is normally found within the World Wide Web databases that you can access today through Mosaic, but you can see that suddenly, by moving controls on the control panel, we can zoom in and see that this is a reconstruction of a seven week old human embryo. This is a reconstruction from approximately 2900 serial cross sections of an embryo sectioned about in the 1930s. It's part of the Carnegie Collection of Human Embryology.



We're looking at this in volume visualization mode, we can rotate the embryo around, we can see internal structures, neurological structures; just in the lower abdomen area, we can see the liver, the arms are very evident - so we are actually looking through the dataset. We can also slice through this dataset obliquely, and look at the internal anatomy that way as well. We can load a volume visualization table that allows us to interactively enter tissue characteristic numbers that control the translucency, transparency, or opacity of various ranges of voxal intensity. And what we've done now is we've made the exterior of the embryo a little more opaque so that as we rotate around with an oblique cutting plane we can see the difference between the cutting plane and the exterior of the embryo a little better. So now we're looking, slicing - we've rotated the embryo to an inferior view and we're looking up at the embryo from below. And we can see, we've just gone through the heart region, we're going through the liver, we're moving inferiorly and we start to get to an area where we can see the herniated gut.

Now the real key to all of this is that these are embedded visualizations. We're actually creating documents that are - I guess you'd call them currently compound documents where you have the traditional type of information, but you've also got, within that document, links to the raw data rather than just pictures generated from data. This allows you to tie together representations of data with the actual data themselves as well as with notes and different kinds of descriptive textual information based on that data.



Our basic objective here is really to create what we're calling a national meta-center which is going to be a computational resource for the entire nation that allows people interested in many different areas, including developmental biology, also multi-dimensional imaging and high-speed networking, and parallel computing. All these people can access this database. And the parallel nature of the computation that's taking place is invisible to the user. They log in through a Mosaic window, and that window is giving them very high-performance control of interactive visualizations of datasets. By scrolling the window, we can see that this is actually embedded within the Mosaic document.

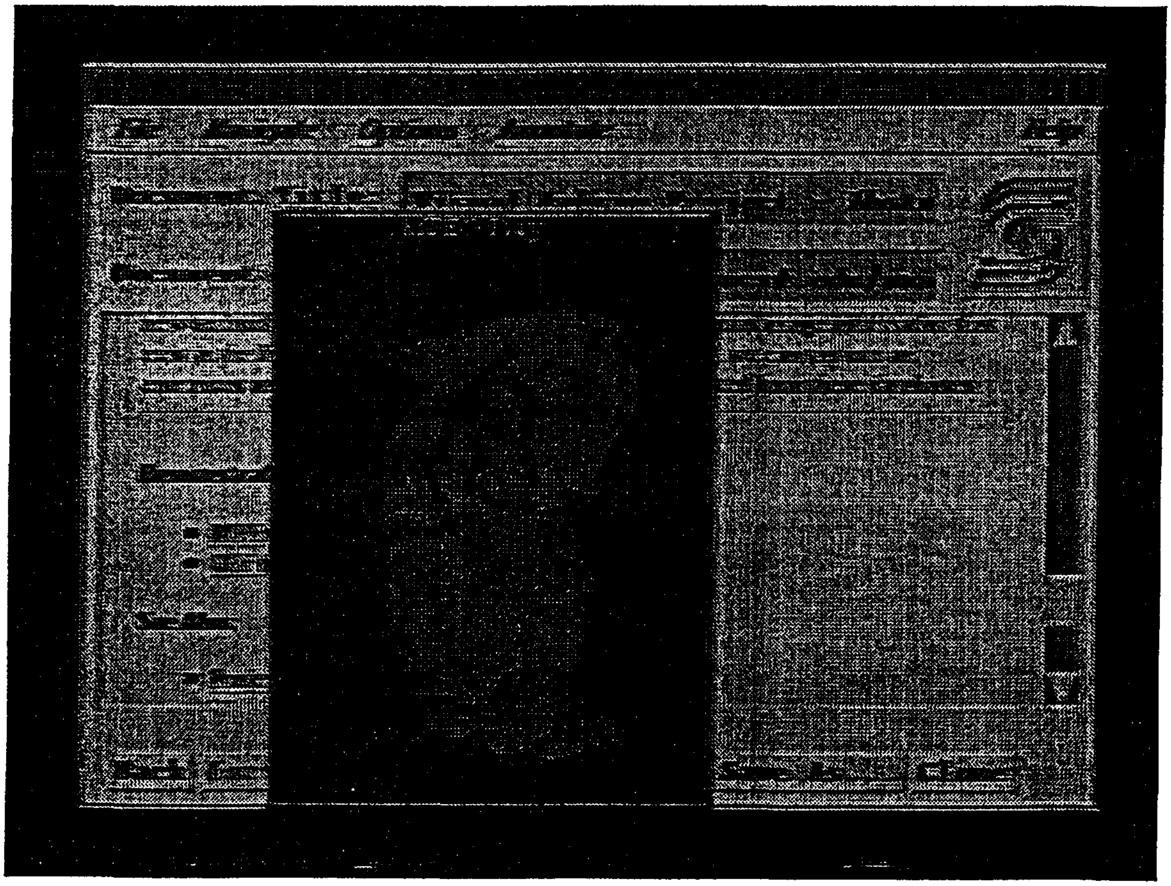
During a recent demonstration of this technology at the corporate briefing center at Silicon Graphics Corporation in Mountain View, California, I discussed some of the implications of this technology for researchers of human genetics.

"We're also looking at using these models as a basis for creating three-dimensional maps of gene expression, which is a way to correlate the findings of the Human Genome project within a context that everyone uses. It sort of sets up a standard space within which everybody can report their findings, so that you can finally have some way of comparing studies that happen in different laboratories.

If you're studying the three-dimensional distribution of gene expression of a gene relating to heart development, what you do now is you have a little fluorescent marker that glows under an ultraviolet light, and you use confocal microscopy to develop a three-dimensional model of it, and then you say, well, it's on here and it's on there and it's on there and you try to describe it in anatomical terms but it is a qualitative description, right?.

But here there would be a standard anatomy space that people could use to describe their findings so that they could, rather than say, yeah, we saw five different studies that said that this was expressed at the bifurcation of the aorta with the Common Carotid artery - you don't have to do in terms of verbal descriptions, you can do it in terms of a true measurable Cartesian coordinate system.

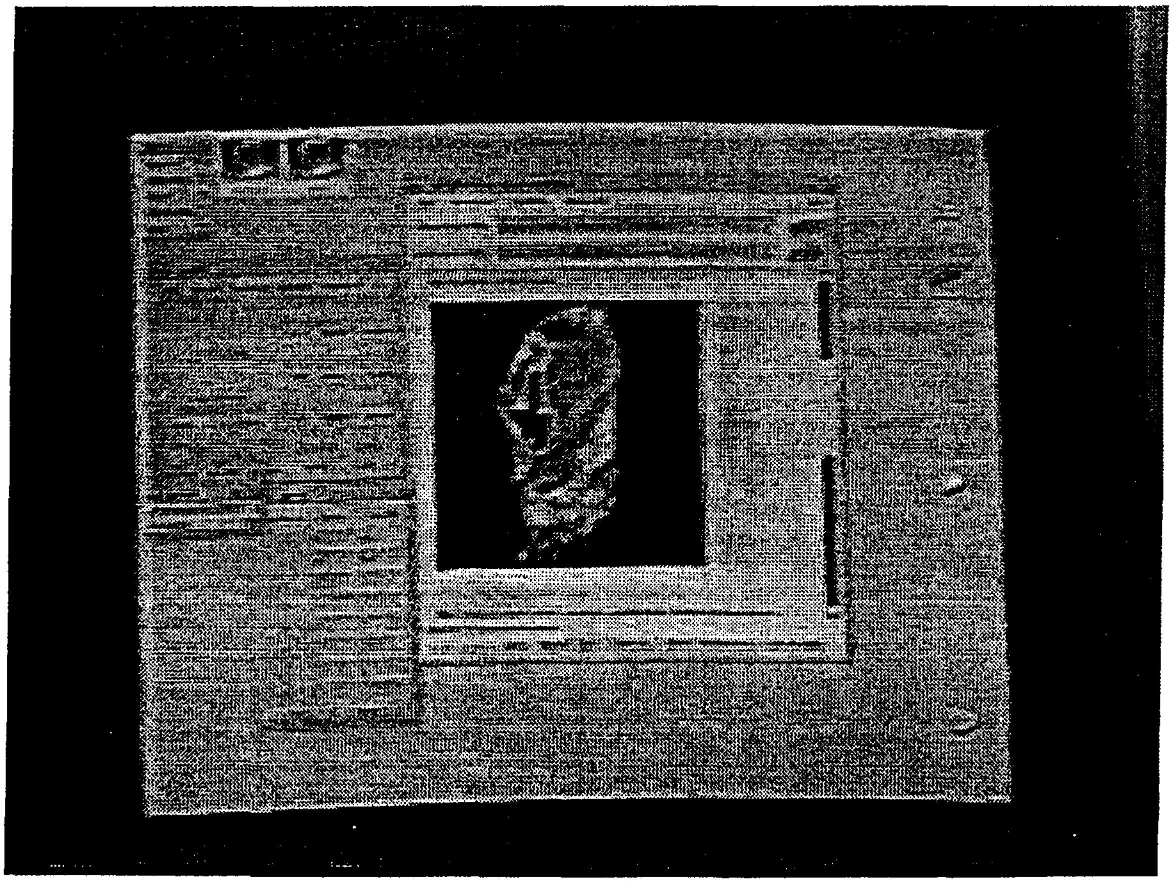
If you take your current version of Mosaic, the kind that is accessible for free through the Internet today, and log onto our home page of the Visible Embryo Project, what you'll see is a series of multi-media documents that basically give you information about the status of the Visible Embryo Project and the status of our current proposal development efforts. You'll be able to load MPEG movies of visualizations of human embryos, as you can see here.



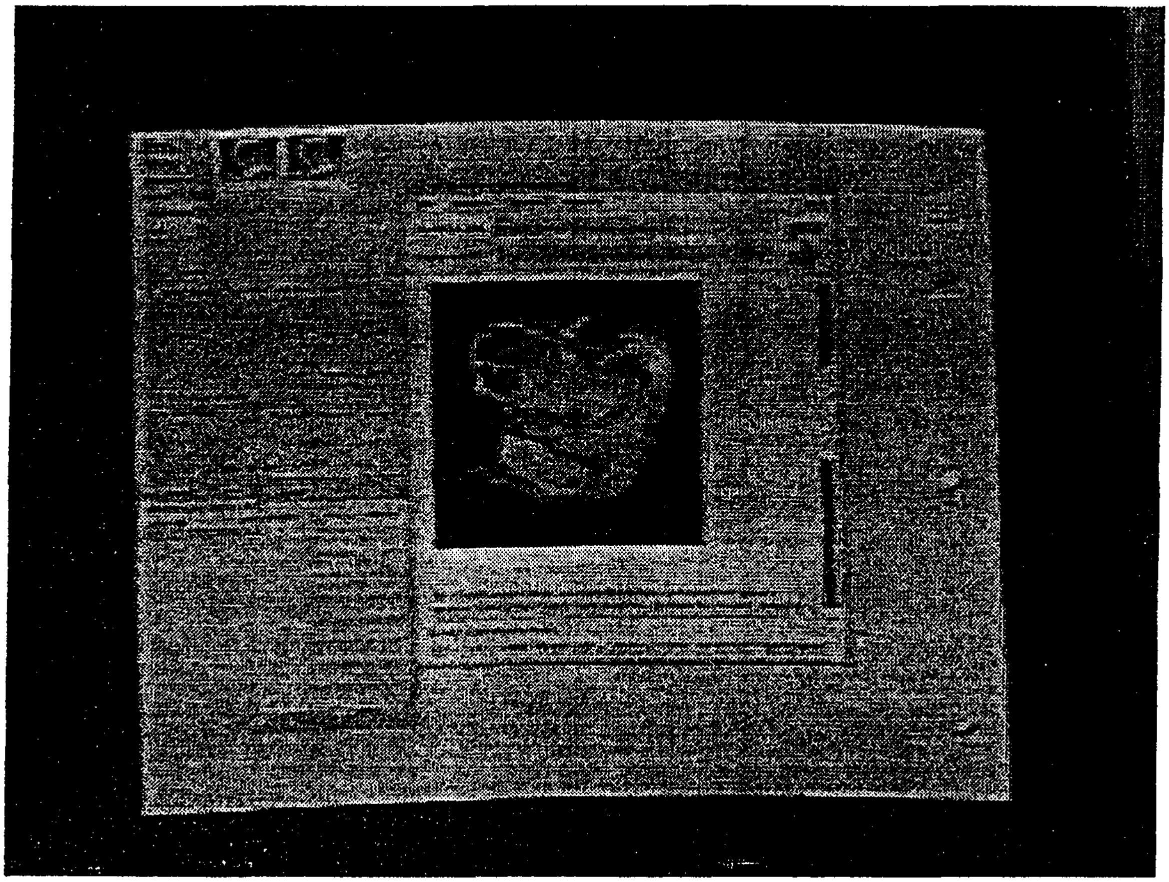
One thing you should keep in mind is that this little MPEG movie is just a canned movie, it's not interactive. Once you hit Play it just goes and it plays and then it goes away, but you can't stop it and interact with it, and rotate that embryo, for instance, to different vantage points.

Also available is an image that shows some of the early work, some screen shots representing the volume visualization tool current as of about last summer. We've come much farther and in fact a lot of the video that you saw earlier in this presentation shows you the current status of this volume visualization package. Last summer, that imaging package was separate from Mosaic, as you see it's off to the right, and Mosaic could just call it but couldn't actually embed visualizations. Now, everything is tied together into a single multi-media document.

You'll also see articles that relate to the Visible Embryo Project. This project has been going on for several years now, mostly on the coattails of other research projects, collaborators funding it wherever they could find the money. But already a significant body of literature is starting to be built up around this project.



Of course, in the very near future, you'll be able to log on through an enhanced version of Mosaic. **We're working together with the National Center for Supercomputing Applications on enhancing the standard release of Mosaic to allow these capabilities.** And you'll be able to access interactive dynamic visualizations that are being served by a network of high-end supercomputers across the country. Even if you're only accessing the system with a machine like a Macintosh or PC, you'll still be able to access the power of these supercomputers from your own location.



What I've attempted to demonstrate in the last several minutes in this presentation is that there's been a considerable amount of work already done in this project that we call the Visible Embryo Project. Many collaborators across the country have worked together to create a set of enabling technologies to allow this project to accomplish its goals. The Visible Embryo Project represents an effort to serve the needs of both the biology community as well as the information science community, in that we are attempting through current applications in information technology to break through barriers that have prevented biological researchers from asking and answering questions about the most fundamental mechanisms of human growth and development. We're also creating a technology development testbed for the information sciences that will allow researchers to push the envelope, so to speak, of information technologies to their very limits.

SPEAKER CONSENT FOR AUDIO/VIDEO RECORDINGS

We would like to audio/video tape your presentation at **Medicine Meets Virtual Reality II: Interactive Technology and Healthcare,**January 27-30, 1994.

Audio and video tapes will be available for immediate distribution to attendees and will be marketed and sold after the conference. Your colleagues will be able to benefit from your remarks by listening to cassettes and viewing tapes whether or not they were in attendance. The tapes will be copyrighted and marketed under Title 17 of the U.S. Code or other law as may be enacted, and all rights to royalties, if any, in conjunction with cassette sales shall hereby be assigned to Medicine Meets Virtual Reality. This agreement does not preclude the publication by you of your paper, speech or comments at any time and in any format.

Please cooperate with the audio/visual taping staff to obtain the best possible recordings by using the microphone at all times and repeating the questions that are asked by persons not near a microphone.

Sign this form below and return it to us at your earliest convenience.

Thank You.

I hereby give permission to ***Medicine Meets Virtual Reality*** to record and distribute audio and video tapes containing my presentation as outlined above.

Signature Michael D Doyle Date 11-30-93

Please print name Michael D. Doyle

Thank you very much for your cooperation. Remember to pick up the complimentary audiotape of your presentation before leaving the meeting.

Please return this completed form to:
MEDICINE MEETS VIRTUAL REALITY
P.O. Box 23220, San Diego, CA 92193
For further information call 619/751-8841,
or Fax 619/751-8842.

UNIVERSITY OF CALIFORNIA, SAN DIEGO
OFFICE OF CONTINUING MEDICAL EDUCATION
DISCLOSURE STATEMENT

It is the policy of the UCSD Office of Continuing Medical Education to insure balance, independence, objectivity, and scientific rigor in all its individually sponsored or jointly sponsored educational programs. All faculty participating in any UCSD sponsored programs are expected to disclose to the program audience any real or apparent conflict(s) of interest that may have a direct bearing on the subject matter of the continuing education program. This pertains to relationships with pharmaceutical companies, biomedical device manufacturers, or other corporations whose products or services are realated to the subject matter of the presenatation topic. The intent of this policy is not to prevent a speaker with a potential conflict of interest from making a presentation. It is merely intended that any potential conflict should be identified openly so that the listeners may form their own judgments about the presentation with the full disclosure of the facts. It remains for the audience to determine whether the speaker's outside interests may reflect a possible bias in either the exposition or the conclusions presented.

MEDICINE MEETS VIRTUAL REALITY II

INTERACTIVE TECHNOLOGY & HEALTHCARE

January 27-30, 1994

PRESENTER'S NAME: Michael D. Doyle
(please print or type)

I have no actual or potential conflict of interest in relation to this program.

11-30-93
Signature (date)

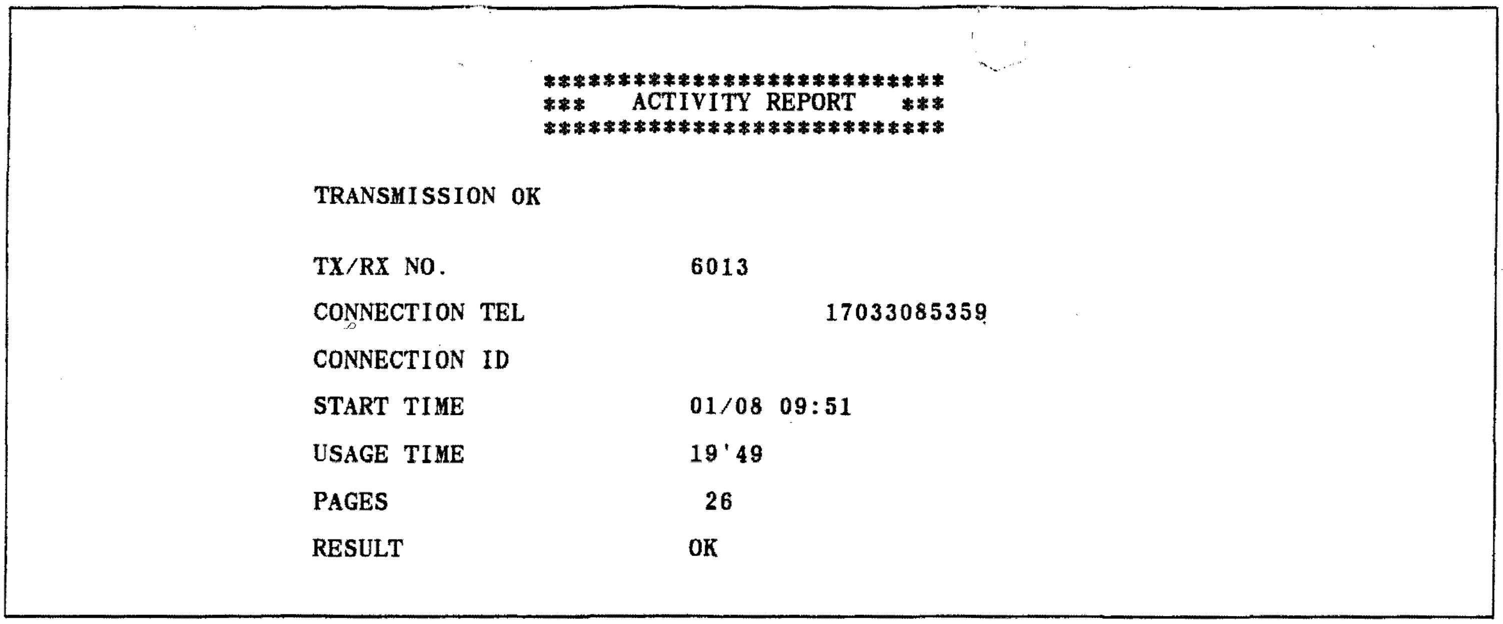
I have a financial interest/arrangement or affiliation with one or more organizations that could be perceiveed as a real or apparent conflict of interest in the context of the subject of this presentation.

Affiliation/Financial Interest	Name of Organization(s)
Grant/Research Support	
Consultant	
Speakers' Bureau	
Major Stock Shareholder	
Other Financial or Material Support	

Signature (date)

Your cooperation in complying with these guidelines is appreciated.

Please return this form as soon as possible to: Medicine Meets Virtual Reality II, P.O. Box 23220, San Diego, CA 92193
Fax: 619-751-8842; Phone: 619-751-8841



****
***   ACTIVITY REPORT   ***
****

TRANSMISSION OK

TX/RX NO.	6013
CONNECTION TEL	17033085359
CONNECTION ID	
START TIME	01/08 09:51
USAGE TIME	19'49
PAGES	26
RESULT	OK

Atty Docket No. 02307I-553

PTO FAX NO.: 1-703-308-5359

ATTENTION: Examiner D. Dinh
Group Art Unit 2317

OFFICIAL COMMUNICATION

FOR THE PERSONAL ATTENTION OF

EXAMINER D. DINH

CERTIFICATION OF FACSIMILE TRANSMISSION

I hereby certify that the following Response After Final and Declaration of Michael D. Doyle Under Rule 131, in re Application of Doyle et al., Serial No. 08/324,443, filed October 17, 1994, for EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS is being facsimile transmitted to the Patent and Trademark Office on the date shown below.

Number of pages being transmitted, including this page: 26