Eolas Technologies Incorporated v. Adobe Systems Incorporated et al Doc. 875 Att. 6

EXHIBIT E

Dockets.Justia.com

985 PH Ex. 14

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Doyle et al. Application No.: 10/217,955 Filed: 08/09/2002 For: DISTRIBUTED HYPERMEDIA METHOD AND SYSTEM FOR AUTOMATICALLY INVOKING EXTERNAL APPLICATION PROVIDING INTERACTION AND DISPLAY OF EMBEDDED OBJECTS WITHIN A HYPERMEDIA DOCUMENT	Examiner: DONAGHUE, LARRY D Art Unit: 2154 SUPPLEMENTAL AMENDMENT

Commissioner for Patents
Alexandria, VA 22313-1450

Sir:

In response to the Office Action mailed 09/09/2004, please amend the application as follows:

Amendments to the Claims begin on page 2 of this paper.

Remarks/Conclusion begins on page 17 of this paper.

AMENDMENTS TO THE CLAIMS:

This listing of the claims replaces all prior versions, and listings, of claims in the application.

LISTING OF CLAIMS:

1-3 (canceled)

4. (currently amended) A method for running an application program in a distributed hypermedia network environment, wherein the network environment comprises at least one client workstation and one network server coupled to the network environment, the method comprising:

receiving, at the client workstation from the network server over the network environment, at least one file containing information to enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window;

executing the browser application on the client workstation, with the browser application:

responding to text formats to initiate processing specified by the text formats;

displaying at least a portion of the document within the browser-controlled window;

identifying an embed text format which corresponds to a first location in the document, where the embed text format specifies the location of at least a portion of an object external to the file, where the object has type information associated with it;

utilizing the type information to identify and locate an executable application external to the file; and

automatically invoking the executable application, in response to the identifying of the embed text format, to execute on the client workstation in order to display the object and enable an end-user to directly interact with ~~interactive processing of~~ the object while the object is being displayed within a display area created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window.

5. (previously presented) The method of claim 4 where:

the information to enable comprises text formats.

6. (previously presented) The method of claim 5 where the text formats are HTML tags.

7. (previously presented) The method of claim 4 where the information contained in the file received comprises at least one embed text format.

8. (previously presented) The method of claim 4 where the step of identifying an embed text format comprises:

parsing the received file to identify text formats included in the received file.

9. (previously presented) The method of claim 8 where the parsing is by a parser in the browser.

10. (previously presented) The method of claim 4 where the processing specified by the text formats is specified directly.

11. (previously presented) The method of claim 4 where the correspondence is implied by the order of the text format in a set of all of the text formats.

12. (previously presented) The method of claim 4 where the embed text format specifies the location of at least a portion of an object directly.

13. (previously presented) The method of claim 4 where having type information associated is by including type information in the embed text format.

14. (previously presented) The method of claim 4 where automatically invoking does not require interactive action by the user.

15. (previously presented) The method of claim 4, wherein said executable application is a controllable application and further comprising the step of:

interactively controlling said controllable application on said client workstation via inter-process communications between said browser and said controllable application

16. (previously presented) The method of claim 15, wherein the communications to interactively control said controllable application continue to be exchanged between the controllable application and the browser even after the controllable application program has been launched.

17. (previously presented) The method of claim 16, wherein additional instructions for controlling said controllable application reside on said network server, wherein said step of interactively controlling said 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 controllable application.

18. (previously presented) The method of claim 17, wherein said additional instructions for controlling said controllable application reside on said client workstation.

19. (currently amended) One or more computer readable media encoded with software comprising computer executable instructions, for use in a distributed hypermedia network environment, wherein the network environment comprises at least one client workstation and one network server coupled to the network environment, and when the software is executed operable to:

receive, at the client workstation from the network server over the network environment, at least one file containing information to enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window;

cause the client workstation to utilize the browser to:

respond to text formats to initiate processing specified by the text formats;

display at least a portion of the document within the browser-controlled window;

identify an embed text format corresponding to a first location in the document, the embed text format specifying the location of at least a portion of an object external to the file, with the object having type information associated with it;

utilize the type information to identify and locate an executable application external to the file; and

automatically invoke the executable application, in response to the identifying of the embed text format, to execute on the client workstation in order to display the object and enable an end-user to directly interact with ~~interactive processing of~~ the object while the object is being displayed within a display area created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window.

20. (previously presented) The computer readable media of claim 19 where: the information to enable comprises text formats.

21. (previously presented) The computer readable media of claim 20 where: the text formats are HTML tags.

22. (previously presented) The computer readable media of claim 19 where: the information contained in the file received comprises at least one embed text format.

23. (currently amended) A method of serving digital information in a computer network environment having a network server coupled the network environment, and where the network environment is a distributed hypermedia environment, the method comprising:

communicating via the network server with at least one client workstation over said network in order to cause said client workstation to:

receive, over said network environment from said server, at least one file containing information to enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window;

execute ~~invoke~~, at said client workstation, a browser application, with the browser application:

responding to text formats to initiate processing specified by the text formats;

displaying, on said client workstation, at least a portion of the document within the browser-controlled window;

identifying an embed text format which corresponds to a first location in the document, where the embed text format specifies the location of at least a portion of an object external to the file, where the object has type information associated with it;

utilizing the type information to identify and locate an executable application external to the file; and

automatically invoking the executable application, in response to the identifying of the embed text format, to execute on the client workstation in order to display the object and enable an end-user to directly interact with ~~interactive processing of~~ the object while the object is being displayed within a display area created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window.

24. (previously presented) The method of claim 23 where:
the information to enable comprises text formats.

25. (previously presented) The method of claim 24 where:
the text formats are HTML tags.

26. (currently amended) The method of claim 23 ~~24~~ where:
the information contained in the file received comprises at least one embed text format.

27. (currently amended) A method for running an executable application ~~program~~ in a computer network environment, wherein said network environment has at least one client workstation and one network server coupled to a network environment, ~~wherein said network environment is a distributed hypermedia environment, wherein said client workstation receives, over said network environment from said server, at least one file containing information to enable a browser application to display, on said client workstation, at least a portion of a distributed hypermedia document within a browser-controlled window, wherein said client workstation executes a browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least a portion of the document is displayed within the browser-controlled window, wherein an embed text format corresponds to a first location in the document is identified, wherein the embed text format specifies the location of at least a portion of an object external to the file, wherein the object has type information associated with it,; wherein the type information is utilized to identify and locate an executable application external to the file, and wherein the executable application is automatically invoked, in response to the identifying of the embed text format,~~ the method comprising:

enabling an end-user to directly interact with an object by utilizing said executable application ~~external to said file~~ to interactively process said object while the object is being displayed within a display area created at a ~~the~~ first location within a ~~the~~ portion of a ~~the~~ hypermedia document being displayed in a ~~the~~ browser-controlled window, wherein said network environment is a distributed hypermedia environment, wherein said client workstation receives, over said network environment from said server, at least one file containing information to enable said browser application to display, on said client workstation, at least said portion of said distributed hypermedia document within said browser-controlled window, wherein said executable application is external to said file, wherein said client workstation executes the browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least said portion of the document is displayed within the browser-controlled window, wherein an embed text format which corresponds to said first location in the document is identified by the browser, wherein the embed text format specifies the location of at least a portion of said object external to the file, wherein the object has type information associated with it, wherein the type information is utilized by the browser to identify and

locate said executable application, and wherein the executable application is automatically invoked by the browser, in response to the identifying of the embed text format.

28. (previously presented) The method of claim 27 where:
the information to enable comprises text formats.

29. (previously presented) The method of claim 28 where:
the text formats are HTML tags.

30. (previously presented) The method of claim 27 where:
the information contained in the file received comprises at least one embed text format.

31. (currently amended) One or more computer readable media encoded with software comprising an executable application ~~computer executable instructions~~ for use in a system having at least one client workstation and one network server coupled to a network environment, ~~wherein said network environment is a distributed hypermedia environment, wherein said client workstation receives, over said network environment from said server, at least one file containing information to enable a browser application to display, on said client workstation, at least a portion of a distributed hypermedia document within a browser-controlled window, wherein said client workstation executes a browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least a portion of the document is displayed within the browser-controlled window, wherein an embed text format corresponds to a first location in the document is identified, wherein the embed text format specifies the location of at least a portion of an object external to the file, wherein the object has type information associated with it, wherein the type information is utilized to identify and locate an executable application external to the file, and wherein the executable application is automatically invoked, in response to the identifying of the embed text format, with software encoded on said computer readable media, identified by said type information and when automatically invoked,~~ operable to:
cause the client workstation to display an ~~said~~ object and enable an end-user to directly interact with ~~interactive processing of~~ said object while the object is being displayed within a display area created at a ~~the~~ first location within a ~~the~~ portion of a ~~the~~ hypermedia

document being displayed in a ~~the~~ browser-controlled window, wherein said network environment is a distributed hypermedia environment, wherein said client workstation receives, over said network environment from said server, at least one file containing information to enable said browser application to display, on said client workstation, at least said portion of said distributed hypermedia document within said browser-controlled window, wherein said executable application is external to said file, wherein said client workstation executes said browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least said portion of the document is displayed within the browser-controlled window, wherein an embed text format which corresponds to said first location in the document is identified by the browser, wherein the embed text format specifies the location of at least a portion of said object external to the file, wherein the object has type information associated with it, wherein the type information is utilized by the browser to identify and locate said executable application, and wherein the executable application is automatically invoked by the browser, in response to the identifying of the embed text format.

32. (previously presented) The method of claim 31 where:
the information to enable comprises text formats.

33. (previously presented) The method of claim 32 where:
the text formats are HTML tags.

34. (previously presented) The method of claim 31 where:
the information contained in the file received comprises at least one embed text format.

35. (currently amended) A method for serving digital information in a computer network environment, ~~with a network server coupled to said network environment, wherein said network environment has at least one client workstation and one network server coupled to a network environment, wherein said network environment is a distributed hypermedia environment, wherein said client workstation receives, over said network environment from said server, at least one file containing information to enable a browser application to display, on said client workstation, at least a portion of a distributed~~

~~hypermedia document within a browser-controlled window, wherein said client workstation executes a browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least a portion of the document is displayed within the browser-controlled window, wherein an embed text format corresponds to a first location in the document is identified, wherein the embed text format specifies the location of at least a portion of an object external to the file, wherein the object has type information associated with it, wherein the type information is utilized to identify and locate an executable application external to the file, and wherein the executable application is automatically invoked, in response to the identifying of the embed text format;~~ said method comprising:

communicating via a ~~said~~ network server with at least one client workstation over said computer network environment in order to cause said client workstation to:

receive at said client workstation, over said computer network environment from said server, at least one file containing information to enable a browser application to display, on said client workstation, at least a portion of a distributed hypermedia document within a browser-controlled window;

utilize an ~~said~~ executable application external to said file to enable an end-user to directly interact with ~~interactive processing of the~~ an object while the object is being displayed within a display area created at a ~~the~~ first location within the portion of the distributed hypermedia document being displayed in the browser-controlled window, with said network server coupled to said computer network environment, wherein said computer network environment has at least said client workstation and said network server coupled to the computer network environment, wherein said computer network environment is a distributed hypermedia environment, wherein said client workstation executes the browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least said portion of the document is displayed within the browser-controlled window, wherein an embed text format which corresponds to said first location in the document is identified by the browser, wherein the embed text format specifies the location of at least a portion of said object external to the file, wherein the object has type information associated with it, wherein the type information is utilized by the browser to identify and locate said executable application, and wherein the executable application is automatically invoked by the browser, in response to the identifying of the embed text format.

36. (previously presented) The method of claim 35 where:
the information to enable comprises text formats.

37. (previously presented) The method of claim 36 where:
the text formats are HTML tags.

38. (previously presented) The method of claim 35 where:
the information contained in the file received comprises at least one embed text format.

39. (currently amended) A method for running an application program in a distributed hypermedia network environment, wherein the distributed hypermedia network environment comprises at least one client workstation and one remote network server coupled to the distributed hypermedia network environment, the method comprising:
receiving, at the client workstation from the network server over the distributed hypermedia network environment, at least one file containing information to enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window;
executing the browser application on the client workstation, with the browser application:
responding to text formats to initiate processing specified by the text formats;
displaying at least a portion of the document within the browser-controlled window;
identifying an embed text format which corresponds to a first location in the document, where the embed text format specifies the location of [[an]] at least a portion of an object;
identifying and locating an executable application ~~program code~~ associated with the object; and
automatically invoking the executable application ~~program code~~, in response to the identifying of the embed text format, ~~to execute on the client workstation~~ in order to ~~display the object and~~ enable an end-user to directly interact with ~~interactive processing of~~ the object, while the object is being displayed within a display area created at the first location within the portion of the hypermedia document being displayed in the browser-

controlled window, wherein the executable application ~~program code~~ is part of a distributed application, and wherein at least a portion of the distributed application is for execution on a remote network server coupled to the distributed hypermedia network environment.

40. (previously presented) The method of claim 39 where:
the information to enable comprises text formats.

41. (previously presented) The method of claim 40 where:
the text formats are HTML tags.

42. (previously presented) The method of claim 39 where:
the information contained in the file received comprises at least one embed text format.

43. (currently amended) A method of serving digital information in a computer network environment having a network server coupled to said computer network environment, and where the network environment is a distributed hypermedia network environment, the method comprising:
communicating via the network server with at least one remote client workstation over said computer network environment in order to cause said client workstation to:
receive, over said computer network environment from the network server, at least one file containing information to enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window;
execute ~~invoke~~, at said client workstation, a browser application, with the browser application~~, with the browser application~~:
responding to text formats to initiate processing specified by the text formats;
displaying, on said client workstation, at least a portion of the document within the browser-controlled window;
identifying an embed text format which corresponds to a first location in the document, where the embed text format specifies the location of [[an]] at least a portion of an object ;

identifying and locating an executable application ~~program code~~ associated with the object; and

automatically invoking the executable application ~~program code~~, in response to the identifying of the embed text format, ~~to execute on the client workstation~~ in order to ~~display the object and~~ enable an end-user to directly interact with ~~interactive processing of~~ the object while the object is being displayed within a display area created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window, wherein the executable application ~~program code~~ is part of a distributed application, and wherein at least a portion of the distributed application is for execution on the network server.

44. (previously presented) The method of claim 43 where:
the information to enable comprises text formats.

45. (previously presented) The method of claim 44 where:
the text formats are HTML tags.

46. (previously presented) The method of claim 43 where:
the information contained in the file received comprises at least one embed text format.

47. (currently amended) A method for serving digital information in a computer network environment, ~~with a network server coupled to said network environment, wherein said network environment has at least one client workstation and one network server coupled to a network environment, wherein the network environment is a distributed hypermedia environment, wherein the client workstation receives, over the network environment from the server, at least one file containing information to enable a browser application to display, on the client workstation, at least a portion of a distributed hypermedia document within a browser-controlled window, wherein the client workstation executes a browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least a portion of the document is displayed within the browser-controlled window, wherein an embed text format corresponds~~

~~to a first location in the document is identified, wherein the embed text format specifies the location of an object, wherein program code associated with the object is identified and located, wherein the executable application is automatically invoked, in response to the identifying of the embed text format, to enable the object while the object is being displayed within a display area created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window, wherein the program code is part of a distributed application, and wherein at least a portion of the distributed application is for execution on the network server;~~ said method comprising:

communicating via a ~~the~~ network server with at least a ~~the~~ remote client workstation over the computer network environment in order to receive commands from the client workstation, with the network server coupled to said computer network environment, wherein said computer network environment has at least said client workstation and said network server coupled to the computer network environment, wherein the computer network environment is a distributed hypermedia environment, wherein the client workstation receives, over the computer network environment from the server, at least one file containing information to enable a browser application to display, on the client workstation, at least a portion of a distributed hypermedia document within a browser-controlled window, wherein the client workstation executes the browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least said portion of the document is displayed within the browser-controlled window, wherein an embed text format which corresponds to a first location in the document is identified by the browser, wherein the embed text format specifies the location of at least a portion of an object, wherein an executable application associated with the object is identified and located by the browser, wherein the executable application is automatically invoked by the browser, in response to the identifying of the embed text format, to enable an end-user to directly interact with the object while the object is being displayed within a display area created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window, wherein the executable application is part of a distributed application, and wherein at least a portion of the distributed application is for execution on the network server;

executing one or more instructions in response to the commands;

sending information to the client workstation in response to the executed instructions, to allow processing of the information at the client workstation to enable said

end-user to directly interact with said object ~~for interactively controlling the controllable application~~.

48. (previously presented) The method of claim 47 where:
the information to enable comprises text formats.

49. (previously presented) The method of claim 48 where:
the text formats are HTML tags.

50. (previously presented) The method of claim 47 where:

the information contained in the file received comprises at least one embed text format.

REMARKS

Claims 4-50 are pending. Claims 1, 19, 23, 26, 27, 31, 35, 39, 43 and 47 are amended herein. Reexamination and reconsideration of all outstanding rejections and objections is requested.

PROCEDURAL HISTORY

This application is a continuation of and claims the benefit of U.S. Application No. 09/075,359, filed May 8, 1998, which is a continuation of U.S. Application No. 08/324,443, filed October 17, 1994, which issued as U.S. Patent No. 5,838,906 ("the '906 patent).

There have been two reexaminations of the '906 patent. The first reexamination was a Director Ordered Reexamination, Control No. 90/006,831 ("the first reexamination"), which resulted in issuance of a Reexamination Certificate on 5/17/2006 without amending the claims. Shortly after the NIRC for the first reexamination was posted on the PAIR page, the second reexamination, Control No. 90/007,838 ("the second reexamination"), was requested.

The office action relating to the currently-pending application was mailed on 07/20/2004, and non-finally rejected claims 1-3. This rejection is identical to the rejection then pending in the first reexamination. A response to the first office action was filed 03/11/2005 and canceled claim 2. A first supplemental amendment was filed 4/11/2008, which presented new claims 4-50. This paper is a second supplemental amendment which presents new amendments to certain of the claims listed above and provides representative citations to support in the specification for the elements and limitations recited in the claims, as requested by the examiner.

Subsequent to the filing of the response, prosecution of the application was suspended by the patent office. Letters of suspension were mailed 05/05/2005, 01/18/2006, 10/18/2006 and 08/13/2006.

The first letter of suspension stated that the outcome of the first reexamination had a material bearing on the patentability of the claims in the present application. The first reexamination resolved all issues of patentablilty in favor of the patentee.

The subsequent letters of suspension stated that the outcome of the second reexamination had a material bearing on the patentability of the claims in the present application. The second reexamination resolved all issues of patentablilty in favor of the patentee.

INTERVIEW SUMMARY

A personal interview was conducted on January 8, 2009. Present at the interview were Examiner Donaghue, inventor Michael Doyle, and Charles E. Krueger, the attorney of record.

The prior art discussed was: 1) the five-way combination of Mosaic, Berners-Lee, Raggett I and II and Toye; and 2) Viola.

The examiner requested that citations to support in the specification for the elements and limitations of the pending claims be provided in the remarks section of a newly presented supplemental amendment.

CITATIONS TO THE SPECIFICATION OF REPRESENTATIVE EXAMPLES OF SUPPORT OF ALL ELEMENTS AND LIMITATIONS RECITED IN THE PENDING CLAIMS

As requested by the examiner, this section cites representative examples of support in the specification for the elements and limitations in the pending claims.

For ease of reference, citations in bold are to column and line numbers of U.S. Patent 5,838,906, which has a specification identical to the pending application and which is the grand-parent of the pending application. In the following, claim language is in italics.

The following citations are representative examples of support in the specification for each element and limitation recited in the claims. Many other parts of the specification, not specifically cited, further support the recitations of the claims and there are other examples that could be cited.

The representative citations are taken from the description of several example embodiments and are not intended to limit the invention, which is defined by the claims.

CLAIM 4. *A method for running an application program in a* ***distributed hypermedia***

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

network environment,

EXAMPLE SUPPORT:

9:48 "Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200.

wherein the network environment comprises at least one ***client workstation*** *and one* ***network server*** *coupled to the network environment,*

EXAMPLE SUPPORT:

8:58 "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."

the method comprising:

receiving, at the client workstation from the network server over the network environment, ***at least one file***

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

containing ***information to enable*** *a browser application to* ***display*** *at least a portion of a* ***distributed hypermedia*** *document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

within a ***browser-controlled window;***

EXAMPLE SUPPORT:

16:8 "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 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."

executing the browser application *on the client workstation,*

EXAMPLE SUPPORT:

9:15 "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. "

with the browser application:

responding to text formats *to initiate processing specified by the text formats;*

EXAMPLE SUPPORT:

9:24 "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."

displaying at least a portion *of the document within the browser-controlled window;*

EXAMPLE SUPPORT:

14:12 "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"

identifying an embed text format *which corresponds to a first location in the document,*

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

where the embed text format ***specifies the location of at least a portion of an object external*** *to the file,*

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."

14:67 "the data object specified by the URL in the EMBED tag. "

where the object has ***type information associated*** *with it;*

EXAMPLE SUPPORT:

12:67 "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."

15:9 "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."

utilizing the type information to ***identify and locate an executable application*** *external to the file; and*

EXAMPLE SUPPORT:

15:11 "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."

automatically invoking the executable application, in response to the identifying *of the embed text format,*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"

15:11 "step 290 is executed to launch a predetermined application.

to ***execute*** *on the client workstation*

EXAMPLE SUPPORT:

9:43 "and application client 210 executes instructions to perform processing in accordance with the present invention. "

in order to display the object and ***enable an end-user to directly interact*** *with the object*

EXAMPLE SUPPORT:

10:2 "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."

while the object is being displayed within a display area *created* ***at the first location*** *within the portion of the hypermedia document being displayed in the browser-controlled window.*

EXAMPLE SUPPORT:

16:8 "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 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."

CLAIM 5. (previously presented) *The method of claim 4 where: the information to enable comprises* ***text formats.***

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 6. *The method of claim 5 where the text formats are* ***HTML tags.***

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 7. *The method of claim 4 where the information contained in the file received comprises* ***at least one embed text format.***

EXAMPLE SUPPORT:

14:29 "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."

CLAIM 8. *The method of claim 4 where the step of identifying an embed text format comprises:*

parsing *the received file to identify text formats included in the received file.*

EXAMPLE SUPPORT:

14:15 "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."

CLAIM 9. *The method of claim 8 where the parsing is by a parser* ***in the browser.***

EXAMPLE SUPPORT:

14:16 "browser program executing on the client computer displays the document and calls a first routine in the HTMLparse.c file called "HTMLparse""

CLAIM 10. *The method of claim 4 where the processing specified by the text formats is* ***specified directly.***

EXAMPLE SUPPORT:

9:30 "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."

CLAIM 11. *The method of claim 4 where the correspondence is implied by the order of the text format in a set of all of the text formats.*

EXAMPLE SUPPORT:

14:53 "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."

CLAIM 12. *The method of claim 4 where the embed text format specifies the location of at least a portion of an object* ***directly***.

EXAMPLE SUPPORT:

14:66 "the data object specified by the URL in the EMBED tag. "

CLAIM 13. *The method of claim 4 where having type information associated is by including type information* ***in the embed text format***.

EXAMPLE SUPPORT:

12:66 "As shown in Table II, the EMBED tag includes TYPE, HREF, WIDTH and HEIGHT elements."

CLAIM 14. *The method of claim 4 where automatically invoking does not require interactive action by the user.*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210

CLAIM 15. *The method of claim 4, wherein said executable application is a controllable application and further comprising the step of:*

interactively controlling said controllable application on said client workstation ***via inter-process communications*** *between said browser and said controllable application*

EXAMPLE SUPPORT:

10:10 "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. "

CLAIM 16. *The method of claim 15, wherein the communications to interactively control said controllable application* ***continue to be exchanged*** *between the controllable application and the browser even after the controllable application program has been launched.*

EXAMPLE SUPPORT:

10:12 "both the browser client and the application client must be in communication with each other"

CLAIM 17. *The method of claim 16, wherein additional instructions for controlling said controllable application* ***reside on said network server,***

EXAMPLE SUPPORT:

10:33 "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."

wherein said step of interactively controlling said controllable application includes the following substeps:

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

EXAMPLE SUPPORT:

10:56 "This information is received by application client 210 and processed to generate a command sent over network 206 to application server 220."

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

EXAMPLE SUPPORT:

10:59 "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."

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

EXAMPLE SUPPORT:

10:63 "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."

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

EXAMPLE SUPPORT:

10:2 "The user is then able to interactively operate controls to recompute different views for the image data."

10:52 "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."

CLAIM 18. *The method of claim 17, wherein said additional instructions for controlling said controllable application reside on said client workstation.*

EXAMPLE SUPPORT:

'906 Claim 5 "The method of claim 4, wherein said additional instructions for controlling said controllable application reside on said client workstation."

CLAIM 19. *One or more computer readable media encoded with software comprising computer executable instructions, for use in a* ***distributed hypermedia***

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

network environment, or

EXAMPLE SUPPORT:

9:48 "Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200."

wherein the network environment comprises at least one ***client workstation*** *and one* ***network server*** *coupled to the network environment,*

EXAMPLE SUPPORT:

8:58 "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."

and when the software is executed operable to:

receive, at the client workstation from the network server over the network environment, ***at least one file***

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

containing ***information to enable*** *a browser application to* ***display*** *at least a portion of a* ***distributed hypermedia*** *document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

within a ***browser-controlled window****;*

EXAMPLE SUPPORT:

16:8 "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 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."

cause the client workstation to utilize the browser *to:*

EXAMPLE SUPPORT:

9:15 "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. "

respond to text formats *to initiate processing specified by the text formats;*

EXAMPLE SUPPORT:

9:24 "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."

display at least a portion *of the document within the browser-controlled window;*

EXAMPLE SUPPORT:

14:12 "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"

identify an embed text format *corresponding to a first location in the document,*

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

the embed text format ***specifying the location of at least a portion of an object external*** *to the file,*

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."
14:67 "the data object specified by the URL in the EMBED tag. "

with the object having ***type information associated*** *with it;*

EXAMPLE SUPPORT:

12:67 "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."
15:9 "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."

utilize the type information to ***identify and locate an executable application*** *external to the file; and*

EXAMPLE SUPPORT:

15:11 "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."

automatically invoke the executable application, in response to the identifying *of the embed text format,*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"

15:11 "step 290 is executed to launch a predetermined application.

*to **execute** on the client workstation*

EXAMPLE SUPPORT:

9:43 "and application client 210 executes instructions to perform processing in accordance with the present invention. "

*in order to display the object and **enable an end-user to directly interact** with the object*

EXAMPLE SUPPORT:

10:2 "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."

***while the object is being displayed within a display area** created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window.*

EXAMPLE SUPPORT:

16:8 "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 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."

CLAIM 20. *The computer readable media of claim 19 where: the information to enable comprises* ***text formats***.

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 21. *The computer readable media of claim 20 where: the text formats are* ***HTML tags***.

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 22. *The computer readable media of claim 19 where: the information contained in the file received comprises* ***at least one embed text format***.

EXAMPLE SUPPORT:

14:29 "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."

CLAIM 23. *A method of serving digital information in a computer network environment having a network server coupled the* ***network environment***,

EXAMPLE SUPPORT:

9:48 "Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200

and where the network environment is a ***distributed hypermedia environment****, the method comprising*:

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

communicating via the network server with at least one client workstation *over said network*

EXAMPLE SUPPORT:

4:44 "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."

5:6 "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. "

8:58 "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."

9:15 "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 order to cause said client workstation to:

receive, over said network environment from said server,

EXAMPLE SUPPORT:

5:10 "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."

9:20 "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. "

at least one file

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

*containing **information to enable** a browser application to **display** at least a portion of a **distributed hypermedia** document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

*within **a browser-controlled window**;*

EXAMPLE SUPPORT:

16:8 "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 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."

***execute**, at said client workstation, a browser application, with the browser application:*

EXAMPLE SUPPORT:

9:15 "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. "

***responding to text formats** to initiate processing specified by the text formats;*

EXAMPLE SUPPORT:

9:24 "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."

displaying, on said client workstation, at least a portion *of the document within the browser-controlled window;*

EXAMPLE SUPPORT:

14:12 "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"

identifying an embed text format *which corresponds to a first location in the document,*

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

where the embed text format ***specifies the location of at least a portion of an object external*** *to the file,*

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."

14:67 "the data object specified by the URL in the EMBED tag. "

where the object has ***type information associated*** *with it;*

EXAMPLE SUPPORT:

12:67 "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."

15:9 "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."

utilizing the type information to ***identify and locate an executable application*** *external to the file; and*

EXAMPLE SUPPORT:

15:11 "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."

automatically invoking the executable application, in response to the identifying *of the embed text format,*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"

15:11 "step 290 is executed to launch a predetermined application.

to ***execute*** *on the client workstation*

EXAMPLE SUPPORT:

9:43 "and application client 210 executes instructions to perform processing in accordance with the present invention. "

in order to display the object and ***enable an end-user to directly interact with*** *the object*

EXAMPLE SUPPORT:

10:2 "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."

while the object is being displayed within a display area *created at* ***the first location*** *within the portion of the hypermedia document being displayed in the browser-controlled window.*

EXAMPLE SUPPORT:

16:8 "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 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."

CLAIM 24. *The method of claim 23 where:*

the information to enable comprises ***text formats.***

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 25. *The method of claim 24 where the text formats are* ***HTML tags***.

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 26. *The method of claim 23 where the information contained in the file received comprises* ***at least one embed text format***.

EXAMPLE SUPPORT:

14:29 "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."

CLAIM 27. *A method for running an executable application in a computer* ***network environment***,

EXAMPLE SUPPORT:

9:48 "Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200.

wherein said network environment has at least ***one client workstation and one network server*** *coupled to a network environment, the method comprising:*

EXAMPLE SUPPORT:

8:58 "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."

enabling an end-user to directly interact with an object *by utilizing said executable application to interactively process said object*

EXAMPLE SUPPORT:

10:2 "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."

while the object is being displayed within a display area *created at a* ***first location*** *within a portion of a hypermedia document*

EXAMPLE SUPPORT:

16:8 "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 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."

being displayed in a ***browser-controlled window,***

EXAMPLE SUPPORT:

16:8 "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 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."

wherein said network environment is a ***distributed hypermedia environment,***

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

wherein said client workstation receives, over said network environment from said server, ***at least one file***

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

*containing **information to enable** said browser application to **display**, on said client workstation, at least said portion of said **distributed hypermedia** document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

*within said browser-controlled window, wherein said executable application is external to said file, wherein said **client workstation executes the browser application**,*

EXAMPLE SUPPORT:

9:15 "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. "

with the browser application ***responding to text formats*** *to initiate processing specified by the text formats,*

EXAMPLE SUPPORT:

9:24 "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."

wherein at least ***said portion of the document is displayed*** *within the browser-controlled window,*

EXAMPLE SUPPORT:

14:12 "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"

wherein ***an embed text format which corresponds to said first location in the document is identified by the browser,***

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

wherein the embed text format ***specifies the location of at least a portion of said object external*** *to the file,*

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."
14:67 "the data object specified by the URL in the EMBED tag. "

wherein the object has ***type information associated*** *with it,*

EXAMPLE SUPPORT:

12:67 "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."
15:9 "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."

wherein the type information is utilized by the browser to ***identify and locate said executable application****,*

EXAMPLE SUPPORT:

15:11 "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."

and wherein the executable application is ***automatically invoked by the browser, in response to the identifying*** *of the embed text format.*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"
15:11 "step 290 is executed to launch a predetermined application.

CLAIM 28. *The method of claim 27 where:*

the information to enable comprises ***text formats***.

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 29. *The method of claim 28 where the text formats are* ***HTML tags***.

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 30. *The method of claim 27 where the information contained in the file received comprises* ***at least one embed text format***.

EXAMPLE SUPPORT:

14:29 "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."

CLAIM 31. *One or more computer readable media encoded with software comprising an executable application for use in a system having at least* ***one client workstation and one network server***

EXAMPLE SUPPORT:

8:58 "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."

coupled to a ***network environment*** *operable to:*

EXAMPLE SUPPORT:

9:48 "Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200.

cause the client workstation to display an object and ***enable an end-user to directly interact with said object***

EXAMPLE SUPPORT:

10:2 "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."

while the object is being displayed within a display area *created at a* ***first location*** *within a portion of a hypermedia document*

EXAMPLE SUPPORT:

16:8 "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 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."

being displayed in a ***browser-controlled window,***

EXAMPLE SUPPORT:

16:8 "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 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."

*wherein said network environment is a **distributed hypermedia environment**,*

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

*wherein said client workstation receives, over said network environment from said server, **at least one file***

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

*containing **information to enable** said browser application to **display**, on said client workstation, at least said portion of said **distributed hypermedia** document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

within said browser-controlled window, wherein said executable application is external to said file, wherein said ***client workstation executes the browser application,***

EXAMPLE SUPPORT:

9:15 "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. "

with the browser application ***responding to text formats*** *to initiate processing specified by the text formats,*

EXAMPLE SUPPORT:

9:24 "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."

wherein at least ***said portion of the document is displayed*** *within the browser-controlled window,*

EXAMPLE SUPPORT:

14:12 "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"

wherein ***an embed text format which corresponds to said first location in the document is identified by the browser,***

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

wherein the embed text format ***specifies the location of at least a portion of said object external*** *to the file,*

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."

14:67 "the data object specified by the URL in the EMBED tag. "

wherein the object has ***type information associated*** *with it,*

EXAMPLE SUPPORT:

12:67 "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."

15:9 "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."

wherein the type information is utilized by the browser to ***identify and locate said executable application****,*

EXAMPLE SUPPORT:

15:11 "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."

and wherein the executable application is ***automatically invoked by the browser, in response*** *to the identifying of the embed text format.*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"

15:11 "step 290 is executed to launch a predetermined application.

CLAIM 32. *The method of claim 31 where:*

the information to enable comprises ***text formats.***

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 33. *The method of claim 32 where the text formats are* ***HTML tags.***

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 34. *The method of claim 31 where the information contained in the file received comprises* ***at least one embed text format.***

EXAMPLE SUPPORT:

14:29 "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."

CLAIM 35. *A method for serving digital information in a computer network environment, said method comprising:*

communicating via a network server with at least one client workstation over said computer network environment *in order to cause said client workstation to:*

EXAMPLE SUPPORT:

4:44 "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."

5:6 "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. "

8:58 "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."

9:15 "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."

receive at said client workstations, over said computer network environment from said server, ***at least one file***

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

*containing **information to enable** said browser application to **display**, on said client workstation, at least said portion a said **distributed hypermedia** document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

*within a **browser-controlled window**,*

EXAMPLE SUPPORT:

16:8 "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 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."

*utilize an executable application external to a file to **enable an end-user to directly interact with an object***

EXAMPLE SUPPORT:

10:2 "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."

***while the object is being displayed within a display area** created at a **first location** within a portion of the distributed hypermedia document*

EXAMPLE SUPPORT:

16:8 "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 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."

being displayed in the ***browser-controlled window****, with said network server coupled to said computer network environment,*

EXAMPLE SUPPORT:

16:8 "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 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."

wherein said computer network environment has ***at least said client workstation and said network server*** *coupled to the computer network environment,*

EXAMPLE SUPPORT:

8:58 "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."

wherein said network environment is a ***distributed hypermedia environment****,*

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

wherein said executable application is external to said file, wherein said ***client workstation executes the browser application,***

EXAMPLE SUPPORT:

9:15 "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. "

with the browser application ***responding to text formats*** *to initiate processing specified by the text formats,*

EXAMPLE SUPPORT:

9:24 "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."

wherein at least ***said portion of the document is displayed*** *within the browser-controlled window,*

EXAMPLE SUPPORT:

14:12 "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"

wherein ***an embed text format which corresponds to said first location in the document is identified by the browser,***

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

wherein the embed text format ***specifies the location of at least a portion of said object external*** *to the file,*

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."

14:67 "the data object specified by the URL in the EMBED tag. "

wherein the object has ***type information associated*** *with it,*

EXAMPLE SUPPORT:

12:67 "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."

15:9 "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."

wherein the type information is utilized by the browser to ***identify and locate said executable application,***

EXAMPLE SUPPORT:

15:11 "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."

and wherein the executable application is ***automatically invoked by the browser, in response*** *to the identifying of the embed text format.*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"

15:11 "step 290 is executed to launch a predetermined application.

CLAIM 36. *The method of claim 35 where:*
the information to enable comprises ***text formats.***

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 37. *The method of claim 36 where:*
the text formats are ***HTML tags.***

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 38. *The method of claim 35 where:*
the information contained in the file received comprises ***at least one embed text format.***

EXAMPLE SUPPORT:

14:29 "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."

CLAIM 39. *A method for running an application program in a distributed hypermedia network environment, wherein the* ***distributed hypermedia***

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

network environment

EXAMPLE SUPPORT:

9:48 "Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200.

comprises at least one client workstation and one ***remote network server***

EXAMPLE SUPPORT:

7:7 "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."

8:58 "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. "

10:61 "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."

coupled to the distributed hypermedia network environment, the method comprising:

receiving, at the client workstation from the network server over the distributed hypermedia network environment, ***at least one file***

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

containing ***information to enable*** *a browser application to* ***display*** *at least a portion of a distributed hypermedia document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

*within a **browser-controlled window**;*

EXAMPLE SUPPORT:

16:8 "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 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."

executing the browser application *on the client workstation, with the browser application:*

EXAMPLE SUPPORT:

9:15 "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. "

responding to text formats *to initiate processing specified by the text formats;*

EXAMPLE SUPPORT:

9:24 "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."

displaying at least a portion *of the document within the browser-controlled window;*

EXAMPLE SUPPORT:

14:12 "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"

identifying an embed text format *which corresponds to a first location in the document,*

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

where the embed text format ***specifies the location of at least a portion*** *of an object;*

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."

14:67 "the data object specified by the URL in the EMBED tag.

identifying and locating an executable application *associated with the object; and*

EXAMPLE SUPPORT:

12:67 "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."

15:9 "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."

15:11 "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."

automatically invoking the executable application, in response to the identifying *of the embed text format,*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"

15:11 "step 290 is executed to launch a predetermined application.

in order to ***enable an end-user to directly interact*** *with the object,*

EXAMPLE SUPPORT:

10:2 "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."

while the object is being displayed within a display area *created at the* ***first location*** *within the portion of the hypermedia document being displayed in the browser-controlled window,*

EXAMPLE SUPPORT:

16:8 "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 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."

wherein the executable application is part of a ***distributed application,***

EXAMPLE SUPPORT:

10:33 "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."

11:18 "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"

and wherein at least a portion of the distributed application is for execution on a ***remote network server coupled to the distributed hypermedia network*** *environment.*

EXAMPLE SUPPORT:

11:24 "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."

CLAIM 40.. *The method of claim 39 where:*
the information to enable comprises ***text formats.***

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 41. *The method of claim 40 where the text formats are* ***HTML tags.***

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 42. *The method of claim 39 where the information contained in the file received comprises* ***at least one embed text format.***

EXAMPLE SUPPORT:

14:29 "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."

CLAIM 43. *A method of serving digital information in a* ***computer network environment***

EXAMPLE SUPPORT:

9:48 "Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200.

having ***a network server*** *coupled to said computer network environment,*

EXAMPLE SUPPORT:

8:58 "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."

and where the network environment is a ***distributed hypermedia*** *network environment, the method comprising:*

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

communicating via the network server with at least one ***remote client workstation*** *over said computer network environment in order to cause said client workstation to:*

EXAMPLE SUPPORT:

8:58 "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. "

receive, over said computer network environment from the network server, ***at least one file***

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

containing ***information to enable*** *a browser application to* ***display*** *at least a portion of a distributed hypermedia document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

within a ***browser-controlled window****;*

EXAMPLE SUPPORT:

16:8 "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 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."

execute, at said client workstation, a browser application*, with the browser application:*

EXAMPLE SUPPORT:

9:15 "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.

responding to text formats *to initiate processing specified by the text formats;*

EXAMPLE SUPPORT:

9:24 "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."

displaying, on said client workstation, at least a portion of the document *within the browser-controlled window;*

EXAMPLE SUPPORT:

14:12 "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"

identifying an embed text format *which corresponds to a first location in the document,*

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

where the embed text format ***specifies the location of at least a portion of an object****;*

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."

14:67 "the data object specified by the URL in the EMBED tag. "

identifying and locating an executable application *associated with the object; and*

EXAMPLE SUPPORT:

15:11 "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."

automatically invoking the executable application, in response to the identifying *of the embed text format,*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"

15:11 "step 290 is executed to launch a predetermined application.

in order to ***enable an end-user to directly interact*** *with the object*

EXAMPLE SUPPORT:

10:2 "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."

while the object is being displayed within a display area *created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window,*

EXAMPLE SUPPORT:

16:8 "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 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."

wherein the executable application is part of a ***distributed application,***

EXAMPLE SUPPORT:

10:33 "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."

11:18 "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"

and wherein at least a portion of the distributed application is for execution on ***the network server.***

EXAMPLE SUPPORT:

11:24 "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."

CLAIM 44. *The method of claim 43 where:*

the information to enable comprises ***text formats.***

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 45. *The method of claim 44 where:*

the text formats are ***HTML tags.***

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 46. *The method of claim 43 where the information contained in the file received comprises* ***at least one embed text format.***

EXAMPLE SUPPORT:

14:29 "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."

CLAIM 47. *A method for serving digital information in a* ***computer network environment****, said method comprising:*

EXAMPLE SUPPORT:

9:48 "Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200.

communicating via ***a network server*** *with at least* ***a remote client workstation*** *over the computer network environment*

EXAMPLE SUPPORT:

11:24 "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."

in order to ***receive commands*** *from the client workstation,*

EXAMPLE SUPPORT:

10:52 "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.

with the network server coupled to said computer network environment, wherein said computer network environment has ***at least said client workstation*** *and said* ***network server*** *coupled to the computer network environment,*

EXAMPLE SUPPORT:

8:58 "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."

wherein the computer network environment is a ***distributed hypermedia*** *environment,*

EXAMPLE SUPPORT:

5:31 "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."

9:48 "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."

wherein the client workstation receives, over the computer network environment from the server, ***at least one file***

EXAMPLE SUPPORT:

2:14 "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."

3:34 "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."

9:20 "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. "

containing ***information to enable*** *a browser application to* ***display****, on the client workstation, at least a portion of a* ***distributed hypermedia*** *document*

EXAMPLE SUPPORT:

1:61 "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."

2:14 "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. "

9:24 "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."

*within a **browser-controlled window**,*

EXAMPLE SUPPORT:

16:8 "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 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."

*wherein the **client workstation executes the browser application**,*

EXAMPLE SUPPORT:

9:15 "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. "

*with the browser application **responding to text formats** to initiate processing specified by the text formats,*

EXAMPLE SUPPORT:

9:24 "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."

wherein at least ***said portion of the document is displayed*** *within the browser-controlled window,*

EXAMPLE SUPPORT:

14:12 "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"

wherein ***an embed text format which corresponds a said first location in the document is identified by the browser,***

EXAMPLE SUPPORT:

14:27 "a check is made as to whether the current tag is the EMBED tag."

wherein the embed text format ***specifies the location of at least a portion of an object,***

EXAMPLE SUPPORT:

6:63 "The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way."

14:32 "Each occurrence of a valid EMBED tag specifies an embedded object."

14:67 "the data object specified by the URL in the EMBED tag. "

wherein an executable application associated with the object is ***identified and located by the browser,***

EXAMPLE SUPPORT:

15:11 "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."

wherein the executable application is ***automatically invoked by the browser,*** *in response to the identifying of the embed text format,*

EXAMPLE SUPPORT:

9:41 "When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information)"

15:11 "step 290 is executed to launch a predetermined application.

*to **enable an end-user to directly interact with the object***

EXAMPLE SUPPORT:

10:2 "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."

***while the object is being displayed within a display area** created at the **first location** within the portion of the hypermedia document being displayed in the browser-controlled window,*

EXAMPLE SUPPORT:

16:8 "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 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."

*wherein the executable application is part of a **distributed application**, and wherein at least a portion of the distributed application is for **execution on the network server**;*

EXAMPLE SUPPORT:

10:33 "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."

11:18 "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"

executing one or more instructions ***in response to the commands;***

sending information to the client workstation in response to the executed instructions, to allow processing of the information at the client workstation ***to enable said end-user to directly interact with said object.***

EXAMPLE SUPPORT:

10:52 "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."

CLAIM 48. *The method of claim 47 where:*
the information to enable comprises ***text formats***.

EXAMPLE SUPPORT:

14:24 "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."

CLAIM 49. *The method of claim 48 where:*
the text formats are ***HTML tags***.

EXAMPLE SUPPORT:

14:19 "the document is parsed or scanned for HTML tags or other symbols. "

CLAIM 50. *The method of claim 47 where:*
the information contained in the file received comprises ***at least one embed text format***.

EXAMPLE SUPPORT:

14:29 "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."

CONCLUSION

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 (925) 944-3320.

Respectfully submitted,

/Charles E. Krueger/

Charles E. Krueger
Reg. No. 30,077

LAW OFFICE OF CHARLES E. KRUEGER
P.O.Box 5607
Walnut Creek, CA 94596
Tel: (925) 944-3320 / Fax: (925) 944-3363