

CLAIM CHART EXHIBIT 3

"VIOLA ALPHA"

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,838,906

- BASED ON VIOLA ALPHA, DATED OCTOBER 16, 1993 [PA-NAT-78\VIOLA\1993-10-16 ALPHA RELEASE- EX AF TO INV CONTENTIONS\VIOLA\TOGO.TAR.Z], (“VIOLA ALPHA”). THE BODY OF MY REPORT HAS A NARRATIVE DESCRIPTION THAT AUGMENTS AND SHOULD BE CONSIDERED PART OF THIS CHART, AND VISE-VERSA FOR THIS AND ALL MY CHARTS.**

Note: Reference is made in this chart to the file testplot.hmml. This is a file that I am appending along with this report and which I used in Viola Video 17 (*see* Appendix C) and which is analogous to the testplot.hmml file I reference in Claim Chart Exhibit 1 for [Viola-5/12/93]. Testplot.hmml can be found on the accompanying DVD of Video Exhibits, in the Related Materials subdirectory, under the folder for Viola Video 17.

Claim Text from '906 Patent	Viola Alpha
<p>906-1.a: A method for running an application program in a computer network environment, comprising:</p>	<p>Viola Alpha discloses an application program. <i>See, e.g.,</i> :</p> <p style="padding-left: 40px;">ViolaWWW is a web browser computer program that was implemented using the Viola toolkit/language system. (See, e.g., violaTOGO\www\objs\www.v.)</p> <p style="padding-left: 40px;">By way of example, Viola's documentation states that “ViolaWWW implements an XWindows browser to the World Wide Web system,” (See, e.g., violaTOGO\docs\viola\about.hmml.)</p> <p>Viola Alpha discloses a computer network environment. <i>See, e.g.,</i> :</p> <p style="padding-left: 40px;">ViolaWWW operates in a computer network environment, including local area networks or the World Wide Web. ViolaWWW operates in computer network environment consisting of clients and servers wherein a server — such as a file server or HTTP server — publishes hypermedia documents such as HMML and HTML files to client workstations. As Viola's documentation notes, “ViolaWWW implements an XWindows browser to the World Wide Web system” (See, e.g., violaTOGO\docs\viola\about.hmml.)</p>
<p>906-1.b: providing at least one client workstation and one</p>	<p>Viola Alpha discloses a client workstation. <i>See, e.g.,</i> :</p>

Claim Text from '906 Patent	Viola Alpha
<p>network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment;</p>	<p>ViolaWWW operated on a client workstation. By way of example, ViolaWWW executed on Sun SPARCstations running a SunOS operating system. This client workstation could interoperate with servers, such as file servers or HTTP servers. Viola's documentation notes that "ViolaWWW implements an XWindows browser to the World Wide Web system" (See, e.g., violaTOGO\docs\viola\about.hmml.)</p> <p>Viola Alpha discloses a network server. <i>See, e.g., :</i></p> <p>ViolaWWW operated with network servers. Viola's documentation notes that "ViolaWWW implements an XWindows browser to the World Wide Web system" (See, e.g., violaTOGO\docs\viola\about.hmml.) Servers that ViolaWWW operate with include file servers or HTTP servers. These servers transmit hypermedia documents, such as HMML or HTML files, and a client workstation running ViolaWWW receives them.</p> <p>Viola Alpha discloses a distributed hypermedia environment. <i>See, e.g., :</i></p> <p>ViolaWWW operates in a distributed hypermedia environment, including local area networks or the World Wide Web. ViolaWWW operates in computer network environment consisting of clients and servers wherein a server — such as a file server or HTTP server — publishes hypermedia documents such as HMML and HTML files to client workstations. As Viola's documentation notes, "ViolaWWW implements an XWindows browser to the World Wide Web system" (See, e.g., violaTOGO\docs\viola\about.hmml.)</p>
<p>906-1.c: executing, at said client workstation, a browser application, that parses a first distributed hypermedia document to identify text formats included in said distributed hypermedia document and for responding to predetermined text formats</p>	<p>Viola Alpha discloses a browser application. <i>See, e.g., :</i></p> <p>ViolaWWW is a web browser computer program that was implemented using the Viola toolkit/language system. (See, e.g., violaTOGO\www\objs\www.v.) By way of example, Viola's documentation states that "ViolaWWW</p>

Claim Text from '906 Patent	Viola Alpha
<p>to initiate processing specified by said text formats;</p>	<p>implements an XWindows browser to the World Wide Web system,” (See, e.g., violaTOGO\docs\viola\about.hmml.)</p> <p>Viola Alpha discloses that the browser application parses a hypermedia document. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia document files (i.e., HTML and HMML documents) that contain enabling information from a network server (e.g., a file server or HTTP server) over a distributed hypermedia network environment. Examples of such documents include those in violaTOGO\docs\viola.</p> <p>The hypermedia document files received from the network server contain predetermined text formats which enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window. For example, the hypermedia documents downloaded by ViolaWWW may contain HMML tags or HTML tags. In particular, the about.hmml hypermedia document contains the HMML tags (i.e., text formats) TITLE, H2 and P. (See, e.g., about.hmml.) As another example, the testAll.html file contains HTML tags, such as TITLE and H1.</p> <p>The hypermedia document downloaded from the remote network server is parsed by ViolaWWW to identify the tags. ViolaWWW then initiates processing specified by the tags. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., www\objs\VWHandler_hmml.v, src\viola\sgmlsA2B.c and src\viola\sgml.c, violaTOGO\src\libWWW.)</p> <p>Viola Alpha discloses a hypermedia document with text formats. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia document files (i.e., HTML and HMML documents) that contain text formats from a network server (e.g., a file server or HTTP server) over a distributed hypermedia network environment. Examples of such</p>

Claim Text from '906 Patent	Viola Alpha
	<p>documents include those in <code>violaTOGO\docs\viola</code>.</p> <p>The hypermedia document files received from the network server contain predetermined text formats which enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window. For example, the hypermedia documents downloaded by ViolaWWW may contain HMML tags or HTML tags. In particular, the <code>about.hmml</code> hypermedia document contains the HMML tags (i.e., text formats) TITLE, H2 and P. (See, e.g., <code>about.hmml</code>.) As another example, the <code>testAll.html</code> file contains HTML tags, such as TITLE and H1.</p> <p>The hypermedia document downloaded from the remote network server is parsed by ViolaWWW to identify the tags. ViolaWWW then initiates processing specified by the tags. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., <code>www\objs\VWHandler_hmml.v</code>, <code>src\viola\sgmlsA2B.c</code> and <code>src\viola\sgml.c</code>, <code>violaTOGO\src\libWWW</code>.)</p>
<p>906-1.d: utilizing said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server,</p>	<p>Viola Alpha discloses that a hypermedia document is received from the server. <i>See, e.g., :</i></p> <p>A client workstation running the ViolaWWW browser receives hypermedia documents from a server. Examples of servers from which ViolaWWW receives hypermedia documents include file servers or HTTP servers. Examples of documents that ViolaWWW receives can be found in <code>violaTOGO\docs\viola</code> and <code>violaTOGO\docs\test</code> folders, and include <code>about.hmml</code> and <code>testAll.html</code>. Code files evidencing ViolaWWW's ability to retrieve documents from servers include <code>www\objs\VWHandler_hmml.v</code> and <code>src\viola\cl_generic.c</code>.</p> <p>Viola Alpha discloses that the browser displays a hypermedia document. <i>See, e.g., :</i></p> <p>ViolaWWW displays hypermedia documents, including HMML and HTML hypermedia documents. Examples of hypermedia HTML and</p>

Claim Text from '906 Patent	Viola Alpha
	<p>HMML documents that ViolaWWW displayed — either by retrieving them from a local directory or retrieving them from a server location — are stored in violaTOGO\docs\viola, such as brief.hmml and in violaTOGO\docs\tests, such as testAll.html. ViolaWWW could display other hypermedia documents as well.</p> <p>ViolaWWW displays HTML and HMML documents by parsing them to identify HMML or HTML tags, and then initiates processing specified by the tags and displays the hypermedia document. For example, ViolaWWW will display the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. Other tags are discussed in more detail elsewhere in this chart.</p>
<p>906-1.e: wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client workstation,</p>	<p>Viola Alpha discloses that a hypermedia document is displayed in a browser window. <i>See, e.g., :</i></p> <p>ViolaWWW displays hypermedia documents, including HMML and HTML hypermedia documents. Examples of hypermedia HTML and HMML documents that ViolaWWW displayed — either by retrieving them from a local directory or retrieving them from a server location — are stored in violaTOGO\docs\viola, such as brief.hmml and in violaTOGO\docs\tests, such as testAll.html. ViolaWWW could display other hypermedia documents as well.</p> <p>ViolaWWW displays HTML and HMML documents by parsing them to identify HMML or HTML tags, and then initiates processing specified by the tags. The documents are displayed in the ViolaWWW browser window.</p>
<p>906-1.f: wherein said first distributed hypermedia document includes an embed text format, located at a first location in said first distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document,</p>	<p>Viola Alpha discloses an embed text format at a first location in a hypermedia document. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia documents (e.g., HTML and HMML documents) from a network server (e.g., a file server or HTTP server) over the distributed hypermedia network environment. The hypermedia document received from the</p>

Claim Text from '906 Patent	Viola Alpha
	<p>network server contains text formats which enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window. These text formats include embed text formats. For example, an HMML file can include an embed text format called VOBJF. (See, e.g., violaTOGO\docs\viola\brief.hmml; docs\viola\about.hmml; testplot.hmml). The VOBJF text format is located at a first location in its hypermedia document.</p> <p>Viola Alpha discloses that the embed text format specifies the location of an object. <i>See, e.g., :</i></p> <p>The VOBJF embed text format specifies the location of an object. This tag specifies the location of an object based on a filepath location in which the object can be found: <VOBJF>/usr/work/viola/apps/plot.v</VOBJF> (See testPlot.hmml.)</p> <p>As another example, brief.hmml includes a VOBJF tag specifying the location of the file chier.v. <VOBJF>../demo/objs/chier.v</VOBJF> (See docs/viola/brief.hmml.)</p> <p>Viola Alpha discloses an object that is external to a hypermedia document. <i>See, e.g., :</i></p> <p>In one example, when a Viola applet is embedded in a ViolaWWW web page, using the VOBJF tag, at least a portion of an object external to a hypermedia document, the default grid, appears in the ViolaWWW window upon browser launch (i.e., testPlot.hmml). The data for the default grid is specified in the file plot.v by the command: output("equation 0"); (See demo\objs\plot.v.)</p>
906-1.g:	Viola Alpha discloses that the object has associated type information. <i>See, e.g., :</i>

Claim Text from '906 Patent	Viola Alpha
<p>wherein said object has type information associated with it utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document, and</p>	<p>For example, the file plot.v contains type information associated with the object. /path{/usr/work/vplot/vplot} (See violaTOGO/demo/objs/plot.v.) The type information is used by the ViolaWWW to identify and locate the vplot executable application. switch (pid = vfork()) { ... case 0: * Child *\</p> <p>... execv(GET_path(self), args); (See src\viola\cl_TTY.c.)</p> <p>Viola Alpha discloses that the browser uses type information to identify and locate an executable application. <i>See, e.g.,</i> :</p> <p>For example, the file plot.v contains type information associated with the object. /path{/usr/work/vplot/vplot} (See violaTOGO/demo/objs/plot.v.) The type information is used by the ViolaWWW to identify and locate the vplot executable application. ViolaWWW then invokes the executable application. switch (pid = vfork()) { ... case 0: * Child *\</p> <p>... execv(GET_path(self), args); (See violaTOGO/src\viola\cl_TTY.c.)</p> <p>Viola Alpha discloses that the executable application is external to the</p>

Claim Text from '906 Patent	Viola Alpha
	<p>hypermedia document. <i>See, e.g., :</i></p> <p>For example, the vplot executable application is external to the testPlot.hmml hypermedia document. (<i>See, e.g., PA-NAT-00000078\vplot\</i>)</p>
<p>906-1.h: wherein said embed text format is parsed by said browser to automatically invoke said executable application to execute on said client workstation in order to display said object and enable an end-user to directly interact with said object within a display area created at said first location within the portion of said first distributed hypermedia document being displayed in said first browser-controlled window.</p>	<p>Viola Alpha discloses that the browser parses the embed text format. <i>See, e.g., :</i></p> <p>The file containing enabling information is downloaded from the remote network server and parsed by ViolaWWW to identify the text formats. ViolaWWW then initiates processing specified by the text formats. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (<i>See, e.g., www\objs\VWHandler_hmml.v, src\viola\sgmlsA2B.c and src\viola\sgml.c, violaTOGO\src\libWWW.</i>)</p> <p>For hypermedia documents containing embed text formats, these embed text formats are likewise identified by parsing the file. For example, when parsing about.hmml or brief.hmml or testPlot.hmml, violaWWW identifies the VOBJF tag while parsing. (<i>See, e.g., src\viola\sgml.c, sgml\objs\HMML_vobjf.v.</i>)</p> <p>Viola Alpha discloses automatic invocation of the executable application. <i>See, e.g., :</i></p> <p>When ViolaWWW parses the VOBJF tag, it automatically invokes the vplot executable application. The automatic invocation does not require action by the user.</p> <p>For example, when ViolaWWW parses the VOBJF tag in testPlot.hmml, the vplot application is automatically invoked as follows:</p> <pre> switch (pid = vfork()) { ... case 0: * Child *\ </pre>

Claim Text from '906 Patent	Viola Alpha
	<p>... execv(GET_path(self), args); (See src\viola\cl_TTY.c.)</p> <p>Viola Alpha discloses that the executable application displays the object. <i>See, e.g., :</i></p> <p>For example, the vplot application displays the object as a grid inside the ViolaWWW window that is displaying testPlot.hmml. The object is displayed at the location in the window corresponding to the VOBJF tag.</p> <p>Viola Alpha discloses that the executable application enables direct interaction with the object. <i>See, e.g., :</i></p> <p>For example, the vplot executable application enables the user to directly interact with the object using the slider bars to rotate the object around the X, Y and Z axes. (See, e.g., violaTOGO\demo\objs\plot.v, src\viola\cl_slider.c, src\viola\cl_client.c.)</p> <p>Viola Alpha discloses that interaction with the object is at a first location in the hypermedia document. <i>See, e.g., :</i></p> <p>For example, the vplot executable application enables the user to directly interact with the object using the slider bars to rotate the object around the X, Y and Z axes. (See, e.g., violaTOGO\demo\objs\plot.v, src\viola\cl_slider.c, src\viola\cl_client.c.) This interaction with the object occurs at the first location within the portion of the HMML document displayed in the ViolaWWW window. (See, e.g., testPlot.hmml, demo\objs\plot.v)</p>
<p>906-2.a: The method of claim 1, wherein said executable</p>	<p>Viola Alpha discloses interactive control via inter-process communications between a browser and an application. <i>See, e.g., :</i></p>

Claim Text from '906 Patent	Viola Alpha
<p>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.</p>	<p>ViolaWWW browser can communicate with an application using inter-process communication. As noted in Viola's documentation, "Viola is message driven, and messages may be generated by a number of sources. A message is typically caused by the user interacting with a graphical user interface object, but it could also be generated by other objects, or by a timer facility. Through a communication facility such as the socket, a message may also be generated from another process on the network." (See docs/viola/brief.hmml.)</p> <p>"A typical viola application consists of a collection of objects interacting - - generating, receiving, and delegating messages -- with each other, and with the user." (See docs/viola/brief.hmml.)</p> <p>"In sum, the Viola language/toolkit system provides an environment where applications are composed of groups of objects, where objects interact, by message passing, with the user and with each other." (See docs/viola/brief.hmml.)</p> <p>In one example, the ViolaWWW browser communicates with vplot using inter-process communication. (See, e.g., violaTOGO\apps\plot.v, violaTOGO\src\cl_TTY.c.) (See also main.c of vplot source code, available at, e.g., PA-NAT-00000078\vplot\MS_SUPP_1205_001\petra\vplot.)</p>
<p>906-3.a: The method of claim 2, 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.</p>	<p>Viola Alpha discloses ongoing inter-process communications. <i>See, e.g., :</i></p> <p>ViolaWWW browser can communicate with an application using inter-process communication. As noted in Viola's documentation, "Viola is message driven, and messages may be generated by a number of sources. A message is typically caused by the user interacting with a graphical user interface object, but it could also be generated by other objects, or by a timer facility. Through a communication facility such as the socket, a message may also be generated from another process on the network." (See docs/viola/brief.hmml.)</p>

Claim Text from '906 Patent	Viola Alpha
	<p>“A typical viola application consists of a collection of objects interacting - - generating, receiving, and delegating messages -- with each other, and with the user.” (See docs/viola/brief.hmml.)</p> <p>“In sum, the Viola language/toolkit system provides an environment where applications are composed of groups of objects, where objects interact, by message passing, with the user and with each other.” (See docs/viola/brief.hmml.)</p> <p>In one example, the ViolaWWW browser communicates with vplot using inter-process communication. (See, e.g., violaTOGO\apps\plot.v, violaTOGO\src\cl_TTY.c.) (See also main.c of vplot source code, available at, e.g., PA-NAT-00000078\vplot\MS_SUPP_1205_001\petra\vplot.) In this example, an end-user can continue interacting with an object, such as the default grid, in an ongoing basis using the slider bars to rotate the object around the X, Y and Z axes. (See, e.g., viola\apps\plot.v, src\cl_slider.c, src\cl_client.c.) These interactions are communicated between the browser and the vplot executable application on an ongoing basis so that vplot can process those interactions.</p>
<p>906-6.a: A computer program product for use in a system having at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment, the computer program product comprising:</p>	<p>Viola Alpha discloses an application program in a computer network environment. <i>See</i> evidence recited for 906-1.a.</p> <p>Viola Alpha also discloses a client workstation and a network server in a distributed hypermedia environment. <i>See</i> evidence recited for 906-1.b.</p>
<p>906-6.b: a computer usable medium having computer readable program code physically embodied therein, said computer program product further comprising:</p>	<p>Viola Alpha discloses computer code physically embodied on a medium. <i>See, e.g., :</i></p> <p>The computer on which ViolaWWW executes includes computer usable media having computer readable program code physically embodied</p>

Claim Text from '906 Patent	Viola Alpha
	<p>therein. By way of example, ViolaWWW executed on Sun SPARCstations running a SunOS operating system. ViolaWWW could be executed on the Sun SPARCstation with the command "viola -o www." Viola, including the www.v file that implements the ViolaWWW browser, contains computer readable code that is stored on the SPARCstation's computer usable media.</p>
<p>906-6.c: computer readable program code for causing said client workstation to execute a browser application to parse a first distributed hypermedia document to identify text formats included in said distributed hypermedia document and to respond to predetermined text formats to initiate processes specified by said text formats;</p>	<p>Viola Alpha discloses a browser application that parses a hypermedia document with text formats and responds to predetermined text formats to initiate processing specified by the text formats. <i>See</i> evidence recited for 906-1.c.</p>
<p>906-6.d: computer readable program code for causing said client workstation to utilize said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server,</p>	<p>Viola Alpha discloses a hypermedia document received from a server and a browser that displays the hypermedia document. <i>See</i> evidence recited for 906-1.d.</p>
<p>906-6.e: wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client workstation,</p>	<p>Viola Alpha discloses that the hypermedia document is displayed in a browser window. <i>See</i> evidence recited for 906-1.e.</p>
<p>906-6.f: wherein said first distributed hypermedia document includes an embed text format, located at a first location in said first distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document,</p>	<p>Viola Alpha discloses an embed text format at a first location in a hypermedia document; that the embed text format specifies the location of an object; and that the object is external to the hypermedia document. <i>See</i> evidence recited for 906-1.f.</p>
<p>906-6.g:</p>	<p>Viola Alpha discloses that the object has associated type information, that the</p>

Claim Text from '906 Patent	Viola Alpha
<p>wherein said object has type information associated with it utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document, and</p>	<p>browser uses the type information to identify and locate an executable application, and that the executable application is external to the hypermedia document. <i>See</i> evidence recited for 906-1.g.</p>
<p>906-6.h: wherein said embed text format is parsed by said browser to automatically invoke said executable application to execute on said client workstation in order to display said object and enable an end-user to directly interact with said object within a display area created at said first location within the portion of said first distributed hypermedia document being displayed in said first browser-controlled window.</p>	<p>Viola Alpha discloses that the browser parses the embed text format; that the browser automatically invokes the executable application; that the executable application displays the object and enables an end-user to directly interact with it; and that interaction with the object is at a first location in the hypermedia document. <i>See</i> evidence recited for 906-1.h.</p>
<p>906-7.a: The computer program product of claim 6, wherein said executable application is a controllable application and further comprising: computer readable program code for causing said client workstation to interactively control said controllable application on said client workstation via inter-process communications between said browser and said controllable application.</p>	<p>Viola Alpha discloses interactive control via inter-process communications between a browser and an application. <i>See</i> evidence recited for 906-2.a.</p>
<p>906-8.a: The computer program product of claim 7, 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</p>	<p>Viola Alpha discloses ongoing inter-process communications. <i>See</i> evidence recited for 906-3.a.</p>

Claim Text from '906 Patent	Viola Alpha
program has been launched.	
<p>906-11.a: The method of claim 3, 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:</p>	<p>Viola Alpha discloses additional instructions on the server <i>See, e.g.,</i> :</p> <p>Viola's documentation states that "Through a communication facility such as the socket, a message may also be generated from another process on the network." (See, e.g., violaTOGO\docs\viola\brief.hmml.) By way of example, and as discussed elsewhere in this chart, ViolaWWW provides for automatically invoking the vplot executable application using the <code>execv</code> system call. (See violaTOGO\src\viola\cl_TTY.c.) Vplot can be implemented as a distributed application wherein a portion of the computations — for example, graphics transformations — can be executed on a remote server by using the <code>rexec</code> socket-based system call or other Unix socket-based system calls in lieu of the <code>execv</code> system call. In this case, additional instructions would reside on the server.</p> <p>Furthermore, as discussed in my report and in Viola video 19.avi, vplot can be interchanged with distributed applications, including VIS. ViolaWWW works with a distributed application in the same way that it works with an executable application on the client workstation.</p>
<p>906-11.b: issuing, from the client workstation, one or more commands to the network server;</p>	<p>Viola Alpha discloses that the client issues commands to the server. <i>See, e.g.,</i> :</p> <p>Viola's documentation states that "Through a communication facility such as the socket, a message may also be generated from another process on the network." (See, e.g., violaTOGO\docs\viola\brief.hmml.) By way of example, and as discussed elsewhere in this chart, ViolaWWW provides for automatically invoking the vplot executable application using the <code>execv</code> system call. (See violaTOGO\src\viola\cl_TTY.c.) Vplot can be implemented as a distributed application wherein a portion of the computations — for example, graphics transformations — can be executed on a remote server by using the <code>rexec</code> socket-based system call or other Unix socket-based system calls in lieu of the <code>execv</code> system call. In this</p>

Claim Text from '906 Patent	Viola Alpha
	<p>case, the client would issue commands to the server. These facilities provide for a client issuing commands to a server.</p> <p>Furthermore, as discussed in my report and in Viola video 19.avi, vplot can be interchanged with distributed applications, including VIS. ViolaWWW works with a distributed application in the same way that it works with an executable application on the client workstation.</p>
<p>906-11.c: executing, on the network server, one or more instructions in response to said commands;</p>	<p>Viola Alpha discloses that the server executes instructions in response to client commands. <i>See, e.g., :</i></p> <p>Viola's documentation states that "Through a communication facility such as the socket, a message may also be generated from another process on the network." (See, e.g., violaTOGO\docs\viola\brief.hmml.) By way of example, and as discussed elsewhere in this chart, ViolaWWW provides for automatically invoking the vplot executable application using the execv system call. (See violaTOGO\src\viola\cl_TTY.c.) Vplot can be implemented as a distributed application wherein a portion of the computations — for example, graphics transformations — can be executed on a remote server by using the rexec socket-based system call or other Unix socket-based system calls in lieu of the execv system call. In this case, the server would execute in response to commands from the client. These facilities provide for a server executing in response to the comments from the client</p> <p>Furthermore, as discussed in my report and in Viola video 19.avi, vplot can be interchanged with distributed applications, including VIS. ViolaWWW works with a distributed application in the same way that it works with an executable application on the client workstation.</p>
<p>906-11.d: sending information from said network server to said client workstation in response to said executed instructions; and</p>	<p>Viola Alpha discloses that the server responds with information to the client. <i>See, e.g., :</i></p> <p>Viola's documentation states that "Through a communication facility such</p>

Claim Text from '906 Patent	Viola Alpha
	<p>as the socket, a message may also be generated from another process on the network." (See, e.g., violaTOGO\docs\viola\brief.hmml.) By way of example, and as discussed elsewhere in this chart, ViolaWWW provides for automatically invoking the vplot executable application using the execv system call. (See violaTOGO\src\viola\cl_TTY.c.) Vplot can be implemented as a distributed application wherein a portion of the computations — for example, graphics transformations — can be executed on a remote server by using the rexec socket-based system call or other Unix socket-based system calls in lieu of the execv system call. In this case, the server would respond with information to the client. These facilities provide for the server responding to the command issued by the client with information to the client.</p> <p>Furthermore, as discussed in my report and in Viola video 19.avi, vplot can be interchanged with distributed applications, including VIS. ViolaWWW works with a distributed application in the same way that it works with an executable application on the client workstation.</p>
<p>906-11.e: processing said information at the client workstation to interactively control said controllable application.</p>	<p>Viola Alpha discloses that the client uses information from the server to interactively control the application. <i>See, e.g., :</i></p> <p>Viola's documentation states that "Through a communication facility such as the socket, a message may also be generated from another process on the network." (See, e.g., violaTOGO\docs\viola\brief.hmml.) By way of example, and as discussed elsewhere in this chart, ViolaWWW provides for automatically invoking the vplot executable application using the execv system call. (See violaTOGO\src\viola\cl_TTY.c.) Vplot can be implemented as a distributed application wherein a portion of the computations — for example, graphics transformations — can be executed on a remote server by using the rexec socket-based system call or other Unix socket-based system calls in lieu of the execv system call. In this case, the client would use information from the server to control the application. These facilities provide for the server responding to the</p>

Claim Text from '906 Patent	Viola Alpha
	<p>command issued by the client with information to the client. The client can then use that information to control the executable application.</p> <p>Furthermore, as discussed in my report and in Viola video 19.avi, vplot can be interchanged with any distributed application, including VIS. ViolaWWW works with a distributed application in the same way that it works with an executable application on the client workstation.</p>
<p>906-13.a: The computer program product of claim 8, wherein additional instructions for controlling said controllable application reside on said network server, wherein said computer readable program code for causing said client workstation to interactively control said controllable application on said client workstation includes:</p>	<p>Viola Alpha discloses additional instructions on the server <i>See</i> evidence recited for 906-11.a.</p>
<p>906-13.b: computer readable program code for causing said client workstation to issue from the client workstation, one or more commands to the network server;</p>	<p>Viola Alpha discloses that the client issues commands to the server. <i>See</i> evidence recited for 906-11.b.</p>
<p>906-13.c: computer readable program code for causing said network server to execute one or more instructions in response to said commands;</p>	<p>Viola Alpha discloses that the server executes instructions in response to client commands. <i>See</i> evidence recited for 906-11.c.</p>
<p>906-13.d: computer readable program code for causing said network sever to send information to said client workstation in response to said executed instructions; and</p>	<p>Viola Alpha discloses that the server responds with information to the client. <i>See</i> evidence recited for 906-11.d.</p>
<p>906-13.e: computer readable program code for causing said</p>	<p>Viola Alpha discloses that the client uses information from the server to interactively control the application. <i>See</i> evidence recited for 906-11.e.</p>

Claim Text from '906 Patent	Viola Alpha
client workstation to process said information at the client workstation to interactively control said controllable application.	

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 7,599,985

- BASED ON VIOLA ALPHA, DATED OCTOBER 16, 1993 [PA-NAT-78\VIOLA\1993-10-16 ALPHA RELEASE- EX AF TO INV CONTENTIONS\VIOLA\TOGO.TAR.Z], (“VIOLA ALPHA”). THE BODY OF MY REPORT HAS A NARRATIVE DESCRIPTION THAT AUGMENTS AND SHOULD BE CONSIDERED PART OF THIS CHART, AND VISE-VERSA FOR THIS AND ALL MY CHARTS.**

Claim Text from '985 Patent	Viola Alpha
<p>985-1.a: 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:</p>	<p>Viola Alpha discloses an application program. <i>See, e.g., :</i></p> <p style="padding-left: 40px;">ViolaWWW is a web browser computer program that was implemented using the Viola toolkit/language system. (See, e.g., violaTOGO\www\objs\www.v.) By way of example, Viola's documentation states that “ViolaWWW implements an XWindows browser to the World Wide Web system,” (See, e.g., violaTOGO\docs\viola\about.hmml.)</p> <p>Viola Alpha discloses a distributed hypermedia environment. <i>See, e.g., :</i></p> <p style="padding-left: 40px;">ViolaWWW operates in a distributed hypermedia environment, including local area networks or the World Wide Web. ViolaWWW operates in computer network environment consisting of clients and servers wherein a server — such as a file server or HTTP server — publishes hypermedia documents such as HMML and HTML files to client workstations. As Viola's documentation notes, “ViolaWWW implements an XWindows browser to the World Wide Web system” (See, e.g., violaTOGO\docs\viola\about.hmml.)</p> <p>Viola Alpha discloses a client workstation. <i>See, e.g., :</i></p> <p style="padding-left: 40px;">ViolaWWW operated on a client workstation. By way of example, ViolaWWW executed on Sun SPARCstations running a SunOS operating</p>

Claim Text from '985 Patent	Viola Alpha
	<p>system. This client workstation could interoperate with servers, such as file servers or HTTP servers. Viola's documentation notes that "ViolaWWW implements an XWindows browser to the World Wide Web system" (See, e.g., violaTOGO\docs\viola\about.hmml.)</p> <p>Viola Alpha discloses a network server. <i>See, e.g., :</i></p> <p>ViolaWWW operated with network servers. Viola's documentation notes that "ViolaWWW implements an XWindows browser to the World Wide Web system" (See, e.g., violaTOGO\docs\viola\about.hmml.) Servers that ViolaWWW operate with include file servers or HTTP servers. These servers transmit hypermedia documents, such as HMML or HTML files, and a client workstation running ViolaWWW receives them.</p> <p>Viola Alpha discloses a distributed hypermedia environment. <i>See, e.g., :</i></p> <p>ViolaWWW operates in a distributed hypermedia environment, including local area networks or the World Wide Web. ViolaWWW operates in computer network environment consisting of clients and servers wherein a server — such as a file server or HTTP server — publishes hypermedia documents such as HMML and HTML files to client workstations. As Viola's documentation notes, "ViolaWWW implements an XWindows browser to the World Wide Web system" (See, e.g., violaTOGO\docs\viola\about.hmml.)</p>
<p>985-1.b: 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;</p>	<p>Viola Alpha discloses a browser application. <i>See, e.g., :</i></p> <p>ViolaWWW is a web browser computer program that was implemented using the Viola toolkit/language system. (See, e.g., violaTOGO\www\objs\www.v.) By way of example, Viola's documentation states that "ViolaWWW implements an XWindows browser to the World Wide Web system," (See, e.g., violaTOGO\docs\viola\about.hmml.)</p>

Claim Text from '985 Patent	Viola Alpha
	<p>Viola Alpha discloses a file containing enabling information. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia document files (i.e., HTML and HMML documents) that contain enabling information from a network server (e.g., a file server or HTTP server) over a distributed hypermedia network environment. Examples of such documents include those in <code>violaTOGO\docs\viola</code>.</p> <p>The hypermedia document files received from the network server contain predetermined text formats which enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window. For example, the hypermedia documents downloaded by ViolaWWW may contain HMML tags or HTML tags. In particular, the <code>about.hmml</code> hypermedia document contains the HMML tags (i.e., text formats) TITLE, H2 and P. (See, e.g., <code>about.hmml</code>.) As another example, the <code>testAll.html</code> file contains HTML tags, such as TITLE and H1.</p> <p>The hypermedia document downloaded from the remote network server is parsed by ViolaWWW to identify the tags. ViolaWWW then initiates processing specified by the tags. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., <code>www\objs\VWHandler_hmml.v</code>, <code>src\viola\sgmlsA2B.c</code> and <code>src\viola\sgml.c</code>, <code>violaTOGO\src\libWWW</code>.)</p> <p>Viola Alpha discloses that the file is received at the client workstation from the network server. <i>See, e.g., :</i></p> <p>A client workstation running the ViolaWWW browser receives hypermedia documents from a server. Examples of servers from which ViolaWWW receives hypermedia documents include file servers or HTTP servers. Examples of files that ViolaWWW receives can be found in <code>violaTOGO\docs\viola</code> and <code>violaTOGO\docs\tests</code> folders, including <code>about.hmml</code> and <code>testAll.html</code>, and <code>\violaTOGO\demo\objs</code>, such as <code>plot.v</code>.</p>

Claim Text from '985 Patent	Viola Alpha
	<p>Code files evidencing ViolaWWW's ability to retrieve documents from servers include <code>www\objs\VWHandler_hmml.v</code> and <code>src\viola\cl_generic.c</code>.</p> <p>Viola Alpha discloses that the browser displays at least a portion of a distributed hypermedia document. <i>See, e.g.,</i> :</p> <p>ViolaWWW displays hypermedia documents, including HMML and HTML hypermedia documents. Examples of hypermedia HTML and HMML documents that ViolaWWW displayed — either by retrieving them from a local directory or retrieving them from a server location — are stored in <code>violaTOGO\docs\viola</code>, such as <code>brief.hmml</code> and in <code>violaTOGO\docs\tests</code>, such as <code>testAll.html</code>. ViolaWWW could display other hypermedia documents as well.</p> <p>ViolaWWW displays HTML and HMML documents by parsing them to identify HMML or HTML tags, and then initiates processing specified by the tags and displays the hypermedia document. For example, ViolaWWW will display the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. Other tags are discussed in more detail elsewhere in this chart.</p> <p>Viola Alpha discloses that at least a portion of a hypermedia document is displayed in a browser-controlled window. <i>See, e.g.,</i> :</p> <p>ViolaWWW displays hypermedia documents, including HMML and HTML hypermedia documents. Examples of hypermedia HTML and HMML documents that ViolaWWW displayed — either by retrieving them from a local directory or retrieving them from a server location — are stored in <code>violaTOGO\docs\viola</code>, such as <code>brief.hmml</code> and in <code>violaTOGO\docs\tests</code>, such as <code>testAll.html</code>. ViolaWWW could display other hypermedia documents as well.</p> <p>ViolaWWW displays HTML and HMML documents by parsing them to identify HMML or HTML tags, and then initiates processing specified by</p>

Claim Text from '985 Patent	Viola Alpha
	<p>the tags. The documents are displayed in the ViolaWWW browser window.</p>
<p>985-1.c: executing the browser application on the client workstation, with the browser application:</p>	<p>Viola Alpha discloses a browser application executing on the client workstation. <i>See, e.g., :</i></p> <p>ViolaWWW is a web browser computer program that was implemented using the Viola toolkit/language system. (See, e.g., <code>violaTOGO\www\objs\www.v.</code>)</p> <p>By way of example, Viola's documentation states that "ViolaWWW implements an XWindows browser to the World Wide Web system," (See, e.g., <code>violaTOGO\docs\viola\about.hmml.</code>)</p>
<p>985-1.d: responding to text formats to initiate processing specified by the text formats;</p>	<p>Viola Alpha discloses responding to text formats to initiate processing specified by the text formats, i.e., parsing text formats. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia document files (i.e., HTML and HMML documents) that contain text formats from a network server (e.g., a file server or HTTP server) over a distributed hypermedia network environment. Examples of such documents include those in <code>violaTOGO\docs\viola.</code></p> <p>The hypermedia document files received from the network server contain predetermined text formats which enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window. For example, the hypermedia documents downloaded by ViolaWWW may contain HMML tags or HTML tags. In particular, the <code>about.hmml</code> hypermedia document contains the HMML tags (i.e., text formats) TITLE, H2 and P. (See, e.g., <code>about.hmml.</code>) As another example, the <code>testAll.html</code> file contains HTML tags, such as TITLE and H1.</p> <p>The hypermedia document downloaded from the remote network server is parsed by ViolaWWW to identify the tags. ViolaWWW then initiates processing specified by the tags. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., <code>www\objs\VWHandler_hmml.v,</code></p>

Claim Text from '985 Patent	Viola Alpha
	src\viola\sgmlsA2B.c and src\viola\sgml.c, violaTOGO\src\libWWW.)
<p>985-1.e: displaying at least a portion of the document within the browser-controlled window;</p>	<p>Viola Alpha discloses that the browser displays a hypermedia document. <i>See, e.g., :</i></p> <p>ViolaWWW displays hypermedia documents, including HMML and HTML hypermedia documents. Examples of hypermedia HTML and HMML documents that ViolaWWW displayed — either by retrieving them from a local directory or retrieving them from a server location — are stored in violaTOGO\docs\viola, such as brief.hmml and in violaTOGO\docs\tests, such as testAll.html. ViolaWWW could display other hypermedia documents as well.</p> <p>ViolaWWW displays HTML and HMML documents by parsing them to identify HMML or HTML tags, and then initiates processing specified by the tags and displays the hypermedia document. For example, ViolaWWW will display the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. Other tags are discussed in more detail elsewhere in this chart.</p> <p>Viola Alpha discloses that a hypermedia document is displayed in a browser window. <i>See, e.g., :</i></p> <p>ViolaWWW displays hypermedia documents, including HMML and HTML hypermedia documents. Examples of hypermedia HTML and HMML documents that ViolaWWW displayed — either by retrieving them from a local directory or retrieving them from a server location — are stored in violaTOGO\docs\viola, such as brief.hmml and in violaTOGO\docs\tests, such as testAll.html. ViolaWWW could display other hypermedia documents as well.</p> <p>ViolaWWW displays HTML and HMML documents by parsing them to identify HMML or HTML tags, and then initiates processing specified by the tags. The documents are displayed in the ViolaWWW browser</p>

Claim Text from '985 Patent	Viola Alpha
<p>985-1.f: 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;</p>	<p>window.</p> <p>Viola Alpha discloses identifying an embed text format. <i>See, e.g., :</i></p> <p>The file containing enabling information is downloaded from the remote network server and parsed by ViolaWWW to identify the text formats. ViolaWWW then initiates processing specified by the text formats. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., www\objs\VWHandler_hmml.v, src\viola\sgmlsA2B.c and src\viola\sgml.c, violaTOGO\src\libWWW.)</p> <p>For hypermedia documents containing embed text formats, these embed text formats are likewise identified by parsing the file. For example, when parsing about.hmml or brief.hmml or testPlot.hmml, violaWWW identifies the VOBJF tag while parsing. (See, e.g., src\viola\sgml.c, sgml\objs\HMML_vobjf.v)</p> <p>Viola Alpha discloses that the embed text format corresponds to a first location in the hypermedia document. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia documents (e.g., HTML and HMML documents) from a network server (e.g., a file server or HTTP server) over the distributed hypermedia network environment. The hypermedia document received from the network server contains text formats which enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window. These text formats include embed text formats. For example, an HMML file can include an embed text format called VOBJF. (See, e.g., violaTOGO\docs\viola\brief.hmml; docs\viola\about.hmml.)</p> <p>Another example is testplot.hmml. In all cases, the VOBJF text format corresponds to a first location in the hypermedia document.</p>

Claim Text from '985 Patent	Viola Alpha
	<p>Viola Alpha discloses that the embed text format specifies the location of an object. <i>See, e.g., :</i></p> <p>The VOBJF embed text format specifies the location of an object. This tag specifies the location of an object based on a filepath location in which the object can be found:</p> <p><VOBJF>/usr/work/viola/apps/plot.v</VOBJF> (See testPlot.hmml.)</p> <p>As another example, brief.hmml includes a VOBJF tag specifying the location of the file chier.v. <VOBJF>../demo/objs/chier.v</VOBJF> (See docs/viola/brief.hmml.)</p> <p>Viola Alpha discloses that the object is external to the file containing enabling information. <i>See, e.g., :</i></p> <p>In one example, when a Viola applet is embedded in a ViolaWWW web page, using the VOBJF tag, at least a portion of an object external to a hypermedia document, the default grid, appears in the ViolaWWW window upon browser launch (i.e., testPlot.hmml). The data for the default grid is specified in the file plot.v by the command: output("equation 0"); (See demo\objs\plot.v.)</p> <p>Viola Alpha discloses that the object has associated type information. <i>See, e.g., :</i></p> <p>For example, the file plot.v contains type information associated with the object. /path{/usr/work/vplot/vplot} (See violaTOGO\demo\objs\plot.v.) The type information is used by the ViolaWWW to identify and locate the</p>

Claim Text from '985 Patent	Viola Alpha
	<pre>vplot executable application. switch (pid = vfork()) { ... case 0: * Child *\ ... execv(GET_path(self), args); (See src\viola\cl_TTY.c.)</pre>
<p>985-1.g: utilizing the type information to identify and locate an executable application external to the file; and</p>	<p>Viola Alpha discloses that the browser uses type information to identify and locate an executable application. <i>See, e.g., :</i></p> <p>For example, the file plot.v contains type information associated with the object. /path{/usr/work/vplot/vplot} (See violaTOGO\demo\objs\plot.v.) The type information is used by the ViolaWWW to identify and locate the vplot executable application. ViolaWWW then invokes the executable application.</p> <pre>switch (pid = vfork()) { ... case 0: * Child *\ ... execv(GET_path(self), args); (See violaTOGO\src\viola\cl_TTY.c.)</pre> <p>Viola Alpha discloses that the executable application is external to the file containing enabling information. <i>See, e.g., :</i></p> <p>For example, the vplot executable application is external to the testPlot.hmm1 file containing enabling information. (<i>See, e.g., PA-NAT-00000078\vplot\</i>).</p>
<p>985-1.h: automatically invoking the executable application,</p>	<p>Viola Alpha discloses that the browser parses the embed text format. <i>See, e.g., :</i></p>

Claim Text from '985 Patent	Viola Alpha
<p>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 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.</p>	<p>The file containing enabling information is downloaded from the remote network server and parsed by ViolaWWW to identify the text formats. ViolaWWW then initiates processing specified by the text formats. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., <code>www\objs\VWHandler_hmml.v</code>, <code>src\viola\sgmlsA2B.c</code> and <code>src\viola\sgml.c</code>, <code>violaTOGO\src\libWWW</code>.)</p> <p>For hypermedia documents containing embed text formats, these embed text formats are likewise identified by parsing the file. For example, when parsing <code>about.hmml</code> or <code>brief.hmml</code> or <code>testPlot.hmml</code>, <code>violaWWW</code> identifies the VOBJF tag while parsing. (See, e.g., <code>src\viola\sgml.c</code>, <code>sgml\objs\HMML_vobjf.v</code>.)</p> <p>Viola Alpha discloses automatic invocation of the executable application. <i>See, e.g., :</i></p> <p>When ViolaWWW parses the VOBJF tag, it automatically invokes the <code>vplot</code> executable application. The automatic invocation does not require action by the user.</p> <p>For example, when ViolaWWW parses the VOBJF tag in <code>testPlot.hmml</code>, the <code>vplot</code> application is automatically invoked as follows:</p> <pre>switch (pid = vfork()) { ... case 0: * Child *\ ... execv(GET_path(self), args); (See <code>src\viola\cl_TTY.c</code>.)</pre> <p>Viola Alpha discloses that the executable application displays the object. <i>See, e.g., :</i></p>

Claim Text from '985 Patent	Viola Alpha
	<p>For example, the vplot application displays the object as a grid inside the ViolaWWW window that is displaying testPlot.hmml. The object is displayed at the location in the window corresponding to the VOBJF tag.</p> <p>Viola Alpha discloses that the executable application enables direct interaction with the object. <i>See, e.g., :</i></p> <p>For example, the vplot executable application enables the user to directly interact with the object using the slider bars to rotate the object around the X, Y and Z axes. (See, e.g., violaTOGO\demo\objs\plot.v, src\viola\cl_slider.c, src\viola\cl_client.c.)</p> <p>Viola Alpha discloses that interaction with the object is at a first location in the hypermedia document. <i>See, e.g., :</i></p> <p>For example, the vplot executable application enables the user to directly interact with the object using the slider bars to rotate the object around the X, Y and Z axes. (See, e.g., violaTOGO\demo\objs\plot.v, src\viola\cl_slider.c, src\viola\cl_client.c.) This interaction with the object occurs at the first location within the portion of the HMML document displayed in the ViolaWWW window. (See, e.g., testPlot.hmml, demo\objs\plot.v)</p>
<p>985-2.a: The method of claim 1 where: the information to enable comprises text formats.</p>	<p>Viola Alpha discloses that the enabling information in the file is text formats. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia document files (i.e., HTML and HMML documents) that contain text formats from a network server (e.g., a file server or HTTP server) over a distributed hypermedia network environment. Examples of such documents include those in violaTOGO\docs\viola.</p> <p>The hypermedia document files received from the network server contain</p>

Claim Text from '985 Patent	Viola Alpha
	<p>predetermined text formats which enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window. For example, the hypermedia documents downloaded by ViolaWWW may contain HMML tags or HTML tags. In particular, the about.hmml hypermedia document contains the HMML tags (i.e., text formats) TITLE, H2 and P. (See, e.g., about.hmml.) As another example, the testAll.html file contains HTML tags, such as TITLE and H1.</p> <p>The hypermedia document downloaded from the remote network server is parsed by ViolaWWW to identify the tags. ViolaWWW then initiates processing specified by the tags. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., www\objs\VWHandler_hmml.v, src\viola\sgmlsA2B.c and src\viola\sgml.c, violaTOGO\src\libWWW.)</p>
<p>985-3.a: The method of claim 2 where the text formats are HTML tags.</p>	<p>Viola Alpha discloses that the text formats are HTML tags. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia document files that contain text formats in the form of HTML tags from a network server (e.g., a file server or HTTP server) over a distributed hypermedia network environment. Examples of such documents include those in violaTOGO\docs\viola and violaTOGO\docs\tests.</p> <p>For example, the testAll.html file contains HTML tags, such as TITLE and H1</p>
<p>985-4.a: The method of claim 1 where the information contained in the file received comprises at least one embed text format.</p>	<p>Viola Alpha discloses that the enabling information in the file includes an embed text format. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia documents (e.g., HTML and HMML documents) from a network server (e.g., a file server or HTTP server) over the distributed hypermedia network environment. The hypermedia document received from the network server contains text formats which enable a browser application</p>

Claim Text from '985 Patent	Viola Alpha
	<p>to display at least a portion of a distributed hypermedia document within a browser-controlled window. These text formats include embed text formats.</p> <p>For example, an HMML file can include an embed text format called VOBJF. (See, e.g., violaTOGO\docs\viola\brief.hmml; docs\viola\about.hmml. Another example is testplot.hmml.</p>
<p>985-5.a: The method of claim 1 where the step of identifying an embed text format comprises: parsing the received file to identify text formats included in the received file.</p>	<p>Viola Alpha discloses that the embed text format is identified by parsing the file containing enabling information. <i>See, e.g., :</i></p> <p>The file containing enabling information is downloaded from the remote network server and parsed by ViolaWWW to identify the text formats. ViolaWWW then initiates processing specified by the text formats. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., www\objs\VWHandler_hmml.v, src\viola\sgmlsA2B.c and src\viola\sgml.c, violaTOGO\src\libWWW.)</p> <p>For hypermedia documents containing embed text formats, these embed text formats are likewise identified by parsing the file. For example, when parsing about.hmml, brief.hmml, or testplot.hmml, violaWWW identifies the VOBJF tag while parsing. (See, e.g., src\viola\sgml.c, sgml\objs\HMML_vobjf.v.)</p>
<p>985-6.a: The method of claim 5 where the parsing is by a parser in the browser.</p>	<p>Viola Alpha discloses that the parser is in the browser <i>See, e.g., :</i></p> <p>Viola includes files for parsing. (See, e.g., www\objs\VWHandler_hmml.v, src\viola\sgmlsA2B.c and src\viola\sgml.c, violaTOGO\src\libwww.)</p> <p>Furthermore, as discussed within my report, HTML and/or HMML parsers were readily available and inherent to the operation of world wide web browser applications.</p>

Claim Text from '985 Patent	Viola Alpha
<p>985-7.a: The method of claim 1 where the processing specified by the text formats is specified directly.</p>	<p>Viola Alpha discloses that the text formats directly specify the processing. <i>See, e.g., :</i></p> <p>ViolaWWW running on the client workstation can receive hypermedia document files (i.e., HTML and HMML documents) that contain text formats from a network server (e.g., a file server or HTTP server) over a distributed hypermedia network environment. Examples of such documents include those in <code>violaTOGO\docs\viola</code>.</p> <p>The hypermedia document files received from the network server contain predetermined text formats which enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window. For example, the hypermedia documents downloaded by ViolaWWW may contain HMML tags or HTML tags. In particular, the <code>about.hmml</code> hypermedia document contains the HMML tags (i.e., text formats) TITLE, H2 and P. (See, e.g., <code>about.hmml</code>.) As another example, the <code>testAll.html</code> file contains HTML tags, such as TITLE and H1.</p> <p>The hypermedia document downloaded from the remote network server is parsed by ViolaWWW to identify the tags. ViolaWWW then initiates processing specified by the tags. For example, ViolaWWW displays the text marked by the H1 tag in large, bold, header text and the text marked by the ITALIC tag in italics. (See, e.g., <code>www\objs\VWHandler_hmml.v</code>, <code>src\viola\sgmlsA2B.c</code> and <code>src\viola\sgml.c</code>, <code>violaTOGO\src\libWWW</code>.)</p>
<p>985-8.a: The method of claim 1 where the correspondence is implied by the order of the text format in a set of all of the text formats.</p>	<p>Viola Alpha discloses that the correspondence is implied by the order of text formats. <i>See, e.g., :</i></p> <p>In ViolaWWW, the correspondence between the location in the document and the text formats is implied by the order of the text formats. For example, <code>brief.hmml</code> has several HMML tags (text formats). In <code>brief.hmml</code>, after some other tags, the H2 tag containing the heading “Classes and Objects” is listed before a paragraph containing “The single</p>

Claim Text from '985 Patent	Viola Alpha
	<p>inheritance classing system...” This is followed by another paragraph with “This is the Viola class heirarchy...” and then the VOBJF tag for “../demo/objs/chier.v”. When ViolaWWW displays the document, the heading appears in a big font, followed by the first and second paragraphs of text, followed by the embedded object. (See, e.g., docs/viola/brief.hmml.)</p> <p>Similarly, for testAll.html, a title (associated with a TITLE tag) is displayed ahead of Header 1 (associated with a subsequent header tag), which is displayed ahead of Header 2 (associated with a still subsequent header tag). (See docs\tests\testAll.html.)</p>
<p>985-9.a: The method of claim 1 where the embed text format specifies the location of at least a portion of an object directly.</p>	<p>Viola Alpha discloses that the embed text format specifies the location of the object directly. <i>See, e.g., :</i></p> <p>The VOBJF embed text format specifies the location of an object directly For example, in testplot.hmml, the VOBJF tag is used to specify the location of the object directly: <VOBJF>/usr/work/viola/apps/plot.v</VOBJF> (See testPlot.hmml.)</p> <p>As another example, brief.hmml includes a VOBJF tag specifying the location of the file chier.v. <VOBJF>../demo/objs/chier.v</VOBJF> (See docs/viola/brief.hmml.)</p>
<p>985-10.a: The method of claim 1 where having type information associated is by including type information in the embed text format.</p>	<p>Viola Alpha discloses that the type information is in the embed text format. <i>See, e.g., :</i></p> <p>For example, the file plot.v (which contains type information as described above) is in the VOBJF embed text format. <VOBJF>/usr/work/viola/apps/plot.v</VOBJF> (See testPlot.hmml.)</p>

Claim Text from '985 Patent	Viola Alpha
<p>985-11.a: The method of claim 1 where automatically invoking does not require interactive action by the user.</p>	<p>Viola Alpha discloses that automatic invocation does not require interactive action by the user. <i>See, e.g., :</i></p> <p>When ViolaWWW parses the VOBJF tag, it automatically invokes the vplot executable application. The automatic invocation does not require action by the user.</p> <p>For example, when ViolaWWW parses the VOBJF tag in testPlot.html, the vplot application is automatically invoked without any interactive action by the user, as follows:</p> <pre>switch (pid = vfork()) { ... case 0: * Child *\ ... execv(GET_path(self), args); (See violaTOGO\src\viola\cl TTY.c.)</pre>
<p>985-16.a: 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:</p>	<p>Viola Alpha discloses computer code physically embodied on a medium. <i>See, e.g., :</i></p> <p>The computer on which ViolaWWW executes includes computer usable media having computer readable program code physically embodied therein. By way of example, ViolaWWW executed on Sun SPARCstations running a SunOS operating system. ViolaWWW could be executed on the Sun SPARCstation with the command “viola -o www.” Viola, including the www.v file that implements the ViolaWWW browser, contains computer readable code that is stored on the SPARCstation's computer usable media.</p> <p>Viola Alpha discloses a client workstation and a network server in a distributed hypermedia environment. <i>See evidence recited for 985-1.a.</i></p>
<p>985-16.b: receive, at the client workstation from the network</p>	<p>Viola Alpha discloses a browser application; a file containing enabling information received from a server; that the browser displays at least a portion of</p>

Claim Text from '985 Patent	Viola Alpha
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;	a distributed hypermedia document; and that the display is in a browser-controlled window. <i>See</i> evidence recited for 985-1.b.
985-16.c: cause the client workstation to utilize the browser to:	Viola Alpha discloses a browser application executing on the client workstation. <i>See</i> evidence recited for 985-1.c.
985-16.d: respond to text formats to initiate processing specified by the text formats;	Viola Alpha discloses parsing text formats. <i>See</i> evidence recited for 985-1.d.
985-16.e: display at least a portion of the document within the browser-controlled window;	Viola Alpha discloses displaying at least a portion of the document within the browser-controlled window. <i>See</i> evidence recited for 985-1.e.
985-16.f: 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;	Viola Alpha discloses identifying an embed text format; that the embed text format corresponds to a first location in a hypermedia document; that the embed text format specifies the location of at least a portion of an object external to the file containing enabling information; and that the object has associated type information. <i>See</i> evidence recited for 985-1.f.
985-16.g: utilize the type information to identify and locate an executable application external to the file; and	Viola Alpha discloses using type information to identify and locate an executable application external to the file. <i>See</i> evidence recited for 985-1.g.
985-16.h: 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 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	Viola Alpha discloses automatically invoking the executable application; that the executable application displays the object and enables an end-user to directly interact with it; and that the interaction with the object is at a first location in a hypermedia document. <i>See</i> evidence recited for 985-1.h.

Claim Text from '985 Patent	Viola Alpha
browser-controlled window.	
<p>985-17.a: The computer readable media of claim 16 where: the information to enable comprises text formats.</p>	<p>Viola Alpha discloses that the enabling information in the file is text formats. <i>See</i> evidence recited for 985-2.a.</p>
<p>985-18.a: The computer readable media of claim 17 where: the text formats are HTML tags.</p>	<p>Viola Alpha discloses that the text formats are HTML tags. <i>See</i> evidence recited for 985-3.a.</p>
<p>985-19.a: The computer readable media of claim 16 where: the information contained in the file received comprises at least one embed text format.</p>	<p>Viola Alpha discloses that the enabling information in the file includes an embed text format. <i>See</i> evidence recited for 985-4.a.</p>
<p>985-20.a: 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:</p>	<p>Viola Alpha discloses digital information. <i>See, e.g., :</i></p> <p>The information that is exchanged between a client workstation running ViolaWWW and a network server (such as web server or file server) is digital information.</p> <p>For example, ViolaWWW running on the client workstation can receive hypermedia documents (HTML and HMML documents) from a network server over the distributed hypermedia network environment. These documents are transmitted according to network protocols that transmit information in digital form. Examples of the types of documents that can be served in digital form can be found in <code>violaTOGO\docs\viola</code> and <code>violaTOGO\docs\tests</code>, including <code>brief.hmml</code>, <code>testAll.html</code>.</p> <p>ViolaWWW also could receive *.v files over networks in digital form. Examples of *.v files are in <code>violaTOGO\demo\objs</code>, and include <code>plot.v</code>.</p> <p>Viola Alpha discloses a network server in a distributed hypermedia environment. <i>See</i> evidence recited for 985-1.a.</p>

Claim Text from '985 Patent	Viola Alpha
<p>985-20.b: communicating via the network server with at least one client workstation over said network in order to cause said client workstation to:</p>	<p>Viola Alpha discloses a client workstation. <i>See</i> evidence recited for 985-1.a.</p> <p>Viola Alpha discloses communicating via network server in order to cause the client workstation to act. <i>See, e.g.,</i> :</p> <p>The ViolaWWW browser operating on a client workstation requests hypermedia documents from a server, such as a file server or an HTTP server. The server responds by communicating to the client workstation. By way of example, the server transmits hypermedia documents, examples of which can be found in <code>violaTOGO\docs\viola</code>. The server can transmit *.v files, examples of which are found in <code>violaTOGO\demo\objs</code>. In one example, the server transmits <code>about.htmml</code> and <code>plot.v</code> to the client workstation.</p> <p>The client workstation, after receiving the hypermedia document and/or *.v file from the network server, parses the hypermedia document and responds to the text formats contained in the hypermedia document to initiate processing specified by the text formats and/or by initiating processing according to a *.v file. (See, e.g., <code>src\viola\sgmlsA2B.c</code> and <code>src\viola\sgml.c</code>.)</p>
<p>985-20.c: 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;</p>	<p>Viola Alpha discloses a browser application; a file containing enabling information received from a server; that the browser displays at least a portion of a distributed hypermedia document; and that the display is in a browser-controlled window. <i>See</i> evidence recited for 985-1.b.</p>
<p>985-20.d: execute, at said client workstation, a browser application, with the browser application:</p>	<p>Viola Alpha discloses a browser application executing on the client workstation. <i>See</i> evidence recited for 985-1.c.</p>
<p>985-20.e: responding to text formats to initiate processing specified by the text formats;</p>	<p>Viola Alpha discloses parsing text formats. <i>See</i> evidence recited for 985-1.d.</p>
<p>985-20.f:</p>	<p>Viola Alpha discloses displaying at least a portion of the document within the</p>

Claim Text from '985 Patent	Viola Alpha
displaying, on said client workstation, at least a portion of the document within the browser-controlled window;	browser-controlled window. <i>See</i> evidence recited for 985-1.e.
985-20.g: 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;	Viola Alpha discloses identifying an embed text format; that the embed text format corresponds to a first location in a hypermedia document; that the embed text format specifies the location of at least a portion of an object external to the file containing enabling information; and that the object has associated type information. <i>See</i> evidence recited for 985-1.f.
985-20.h: utilizing the type information to identify and locate an executable application external to the file; and	Viola Alpha discloses using type information to identify and locate an executable application external to the file. <i>See</i> evidence recited for 985-1.g.
985-20.i: 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 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.	Viola Alpha discloses automatically invoking the executable application; that the executable application displays the object and enables an end-user to directly interact with it; and that the interaction with the object is at a first location in a hypermedia document. <i>See</i> evidence recited for 985-1.h.
985-21.a: The method of claim 20 where: the information to enable comprises text formats.	Viola Alpha discloses that the enabling information in the file is text formats. <i>See</i> evidence recited for 985-2.a.
985-22.a: The method of claim 21 where: the text formats are HTML tags.	Viola Alpha discloses that the text formats are HTML tags. <i>See</i> evidence recited for 985-3.a.

Claim Text from '985 Patent	Viola Alpha
<p>985-23.a: The method of claim 20 where: the information contained in the file received comprises at least one embed text format.</p>	<p>Viola Alpha discloses that the enabling information in the file includes an embed text format. <i>See</i> evidence recited for 985-4.a.</p>
<p>985-24.a: A method for running an executable application in a computer network environment, wherein said network environment has at least one client workstation and one network server coupled to a network environment, the method comprising:</p>	<p>Viola Alpha discloses a client workstation and a network server in a network environment. <i>See</i> evidence recited for 985-1.a.</p> <p>Viola Alpha discloses an executable application. <i>See</i> evidence recited for 985-1.g.</p>
<p>985-24.b: enabling an end-user to directly interact with an object by utilizing said executable application to interactively process said object while the object is being displayed within a display area created at a first location within a portion of a hypermedia document being displayed in a browser-controlled window,</p>	<p>Viola Alpha discloses displaying at least a portion of the document within the browser-controlled window. <i>See</i> evidence recited for 985-1.e.</p> <p>Viola Alpha discloses an object external to a file containing enabling information. <i>See</i> evidence recited for 985-1.f.</p> <p>Viola Alpha discloses that there is enabling of an end-user to directly interact with the object. <i>See, e.g., :</i></p> <p style="padding-left: 40px;">For example, the vplot executable application enables the user to directly interact with the object using the slider bars to rotate the object around the X, Y and Z axes. (See, e.g., violaTOGO\demo\objs\plot.v, src\viola\cl_slider.c, src\viola\cl_client.c.)</p> <p>Viola Alpha discloses that the interaction with the object is at a first location in a hypermedia document. <i>See</i> evidence recited for 985-1.h.</p> <p>Viola Alpha discloses that the object is displayed at a first location within a portion of the hypermedia document being displayed. <i>See, e.g., :</i></p> <p style="padding-left: 40px;">In one example, the object is displayed and interactive processing of the</p>

Claim Text from '985 Patent	Viola Alpha
	<p>object occurs at the first location within the portion of the hypermedia document displayed in the ViolaWWW window.</p> <p>For example, the vplot application displays the object as a grid (the default grid) inside the ViolaWWW window. The object is displayed at the first location in the portion of the testPlot.hmml hypermedia document being displayed in the ViolaWWW window. (See, e.g., testPlot.hmml.)</p>
<p>985-24.c: wherein said network environment is a distributed hypermedia environment,</p>	<p>Viola Alpha discloses a client workstation and a network server in a distributed hypermedia environment. <i>See</i> evidence recited for 985-1.a.</p>
<p>985-24.d: 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,</p>	<p>Viola Alpha discloses a browser application; a file containing enabling information received from a server; that the browser displays at least a portion of a distributed hypermedia document; and that the display is in a browser-controlled window. <i>See</i> evidence recited for 985-1.b.</p>
<p>985-24.e: wherein said executable application is external to said file,</p>	<p>Viola Alpha discloses an executable application external to the file. <i>See</i> evidence recited for 985-1.g.</p>
<p>985-24.f: wherein said client workstation executes the browser application, with the browser application responding to text formats to initiate processing specified by the text formats,</p>	<p>Viola Alpha discloses a browser application executing on the client workstation. <i>See</i> evidence recited for 985-1.c.</p> <p>Viola Alpha discloses parsing text formats. <i>See</i> evidence recited for 985-1.d.</p>
<p>985-24.g: wherein at least said portion of the document is displayed within the browser-controlled window,</p>	<p>Viola Alpha discloses displaying at least a portion of the document within the browser-controlled window. <i>See</i> evidence recited for 985-1.e.</p>
<p>985-24.h: wherein an embed text format which corresponds to said first location in the document is identified by the browser,</p>	<p>Viola Alpha discloses identifying an embed text format and that the embed text format corresponds to a first location in a hypermedia document. <i>See</i> evidence recited for 985-1.f.</p>

Claim Text from '985 Patent	Viola Alpha
<p>985-24.i: wherein the embed text format specifies the location of at least a portion of said object external to the file,</p>	<p>Viola Alpha discloses that the embed text format specifies the location of at least a portion of an object external to the file containing enabling information. <i>See</i> evidence recited for 985-1.f.</p>
<p>985-24.j: wherein the object has type information associated with it,</p>	<p>Viola Alpha discloses that the object has associated type information. <i>See</i> evidence recited for 985-1.f.</p>
<p>985-24.k: wherein the type information is utilized by the browser to identify and locate said executable application, and</p>	<p>Viola Alpha discloses using type information to identify and locate an executable application external to the file. <i>See</i> evidence recited for 985-1.g.</p>
<p>985-24.l: wherein the executable application is automatically invoked by the browser, in response to the identifying of the embed text format.</p>	<p>Viola Alpha discloses automatically invoking the executable application. <i>See</i> evidence recited for 985-1.h.</p>
<p>985-25.a: The method of claim 24 where: the information to enable comprises text formats.</p>	<p>Viola Alpha discloses that the enabling information in the file is text formats. <i>See</i> evidence recited for 985-2.a.</p>
<p>985-26.a: The method of claim 25 where: the text formats are HTML tags.</p>	<p>Viola Alpha discloses that the text formats are HTML tags. <i>See</i> evidence recited for 985-3.a.</p>
<p>985-27.a: The method of claim 24 where: the information contained in the file received comprises at least one embed text format.</p>	<p>Viola Alpha discloses that the enabling information in the file includes an embed text format. <i>See</i> evidence recited for 985-4.a.</p>
<p>985-28.a: One or more computer readable media encoded with software comprising an executable</p>	<p>Viola Alpha discloses computer code physically embodied on a medium. <i>See</i> evidence recited for 985-16.a.</p>

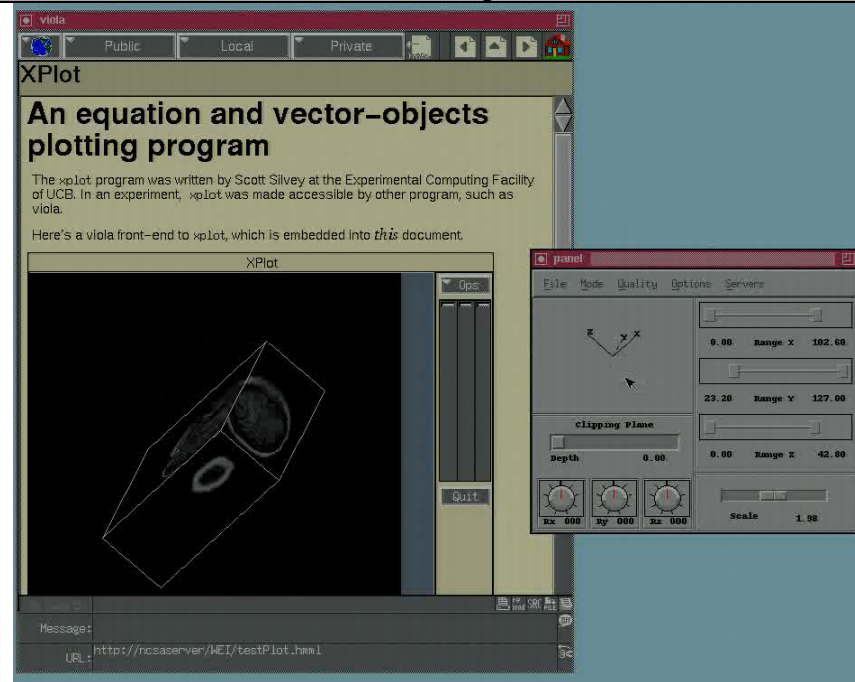
Claim Text from '985 Patent	Viola Alpha
<p>application for use in a system having at least one client workstation and one network server coupled to a network environment, operable to:</p>	<p>Viola Alpha discloses a client workstation and a network server in a network environment. <i>See</i> evidence recited for 985-1.a.</p> <p>Viola Alpha discloses an executable application. <i>See</i> evidence recited for 985-1.g.</p>
<p>985-28.b: cause the client workstation to display an object and enable an end-user to directly interact with said object while the object is being displayed within a display area created at a first location within a portion of a hypermedia document being displayed in a browser-controlled window,</p>	<p>Viola Alpha discloses displaying at least a portion of the document within the browser-controlled window. <i>See</i> evidence recited for 985-1.e.</p> <p>Viola Alpha discloses an object external to a file containing enabling information. <i>See</i> evidence recited for 985-1.f.</p> <p>Viola Alpha discloses that there is enabling of an end-user to directly interact with the object. <i>See</i> evidence recited for 985-24.b.</p> <p>Viola Alpha discloses that the interaction with the object is at a first location in a hypermedia document. <i>See</i> evidence recited for 985-1.h.</p> <p>Viola Alpha discloses that the object is displayed within a display area created at the first location.. <i>See, e.g., :</i></p> <p style="padding-left: 40px;">In one example, the object is displayed and interactive processing of the object occurs at the first location within the portion of the hypermedia document displayed in the ViolaWWW window.</p> <p style="padding-left: 40px;">For example, the vplot application displays the object as a grid (the default grid) inside the ViolaWWW window. The object is displayed at the first location in the portion of the testPlot.hmml hypermedia document being displayed in the ViolaWWW window. (See, e.g., testPlot.hmml.)</p>
<p>985-28.c: wherein said network environment is a distributed hypermedia environment,</p>	<p>Viola Alpha discloses a client workstation and a network server in a distributed hypermedia environment. <i>See</i> evidence recited for 985-1.a.</p>
<p>985-28.d:</p>	<p>Viola Alpha discloses a browser application; a file containing enabling</p>

Claim Text from '985 Patent	Viola Alpha
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,	information received from a server; that the browser displays at least a portion of a distributed hypermedia document; and that the display is in a browser-controlled window. <i>See</i> evidence recited for 985-1.b.
985-28.e: wherein said executable application is external to said file,	Viola Alpha discloses an executable application external to the file. <i>See</i> evidence recited for 985-1.g.
985-28.f: wherein said client workstation executes said browser application, with the browser application responding to text formats to initiate processing specified by the text formats,	Viola Alpha discloses a browser application executing on the client workstation. <i>See</i> evidence recited for 985-1.c. Viola Alpha discloses parsing text formats. <i>See</i> evidence recited for 985-1.d.
985-28.g: wherein at least said portion of the document is displayed within the browser-controlled window,	Viola Alpha discloses displaying at least a portion of the document within the browser-controlled window. <i>See</i> evidence recited for 985-1.e.
985-28.h: wherein an embed text format which corresponds to said first location in the document is identified by the browser,	Viola Alpha discloses identifying an embed text format and that the embed text format corresponds to a first location in a hypermedia document. <i>See</i> evidence recited for 985-1.f.
985-28.i: wherein the embed text format specifies the location of at least a portion of said object external to the file,	Viola Alpha discloses that the embed text format specifies the location of at least a portion of an object external to the file containing enabling information. <i>See</i> evidence recited for 985-1.f.
985-28.j: wherein the object has type information associated with it,	Viola Alpha discloses that the object has associated type information. <i>See</i> evidence recited for 985-1.f.
985-28.k: wherein the type information is utilized by the browser to identify and locate said executable application, and	Viola Alpha discloses using type information to identify and locate an executable application external to the file. <i>See</i> evidence recited for 985-1.g.

Claim Text from '985 Patent	Viola Alpha
<p>985-28.i: wherein the executable application is automatically invoked by the browser, in response to the identifying of the embed text format.</p>	<p>Viola Alpha discloses automatically invoking the executable application. <i>See</i> evidence recited for 985-1.h.</p>
<p>985-36.a: 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:</p>	<p>Viola Alpha discloses an application program in a distributed hypermedia environment comprising at least client workstation and network server. <i>See</i> evidence recited for 985-1.a.</p>
<p>985-36.b: 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;</p>	<p>Viola Alpha discloses a browser application; a file containing enabling information; that the file is received at the client workstation from the network server; that the browser displays at least a portion of a distributed hypermedia document; and that at least a portion of a hypermedia document is displayed in a browser-controlled window. <i>See</i> evidence recited for 985-1.b.</p>
<p>985-36.c: executing the browser application on the client workstation, with the browser application:</p>	<p>Viola Alpha discloses a browser application executing on the client workstation. <i>See</i> evidence recited for 985-1.c.</p>
<p>985-36.d: responding to text formats to initiate processing specified by the text formats;</p>	<p>Viola Alpha discloses parsing text formats. <i>See</i> evidence recited for 985-1.d.</p>
<p>985-36.e: displaying at least a portion of the document within the browser-controlled window;</p>	<p>Viola Alpha discloses displaying at least a portion of the document within the browser-controlled window. <i>See</i> evidence recited for 985-1.e.</p>
<p>985-36.f: identifying an embed text format which</p>	<p>Viola Alpha discloses an object. <i>See, e.g., :</i></p>

Claim Text from '985 Patent	Viola Alpha
<p>corresponds to a first location in the document, where the embed text format specifies the location of at least a portion of an object;</p>	<p>In one example, when a Viola applet is embedded in a ViolaWWW web page, using the VOBJF tag, at least a portion of an object, the default grid, appears in the ViolaWWW window upon browser launch. The data for the default grid is specified in the file plot.v by the command: <code>output("equation 0");</code> (See demo\objs\plot.v.)</p> <p>Viola Alpha discloses identifying an embed text format; that the embed text format corresponds to a first location in the hypermedia document; and that the embed text format specifies the location of an object. <i>See</i> evidence recited for 985-1.f.</p>
<p>985-36.g: identifying and locating an executable application associated with the object; and</p>	<p>Viola Alpha discloses that the browser identifies and locates an executable application associated with the object. <i>See, e.g.,</i></p> <p>For example, the file plot.v contains type information associated with the object.</p> <pre>/path {/usr/work/vplot/vplot} (See violaTOGO\demo\objs\plot.v.)</pre> <p>The type information is used by the ViolaWWW to identify and locate the vplot executable application. ViolaWWW then invokes the executable application.</p> <pre>switch (pid = vfork()) { ... case 0: * Child *\ ... execv(GET_path(self), args); (See violaTOGO\src\viola\cl_TTY.c.)</pre>
<p>985-36.h: automatically invoking the executable application, in response to the identifying of the embed text format, in order to enable an end-user to directly interact with the object, while the object is being</p>	<p>Viola Alpha discloses identifying an embed text format. <i>See</i> evidence recited in 985-1.f.</p> <p>Viola Alpha discloses automatic invocation of the executable application; that the executable application displays the object; that the executable application</p>

Claim Text from '985 Patent	Viola Alpha
<p>displayed within a display area created at the first location within the portion of the hypermedia document being displayed in the browser-controlled window,</p>	<p>enables direct interaction with the object; and that interaction with the object is at a first location in the hypermedia document. <i>See</i> evidence recited in 985-1.h.</p> <p>Viola Alpha discloses that the object is displayed at a first location within a portion of the hypermedia document being displayed. <i>See</i> evidence recited at 985-24.b.</p> <p>Viola Alpha discloses that a hypermedia document is displayed in a browser window. <i>See, e.g.</i>, evidence recited for 985-1.e.</p>
<p>985-36.i: wherein the executable application is part of a distributed application, and</p>	<p>Viola Alpha discloses a distributed application. <i>See, e.g.</i>, :</p> <p>Viola's documentation states that "Through a communication facility such as the socket, a message may also be generated from another process on the network." (See, e.g., <code>violaTOGO\docs\viola\brief.hmml</code>.) By way of example, and as discussed elsewhere in this chart, ViolaWWW provides for automatically invoking the <code>vplot</code> executable application using the <code>execv</code> system call. (See <code>violaTOGO\src\viola\cl_TTY.c</code>.) <code>Vplot</code> can be implemented as a distributed application wherein a portion of the computations — for example, graphics transformations — can be executed on a remote server by using the <code>rexec</code> socket-based system call or other Unix socket-based system calls in lieu of the <code>execv</code> system call. In this case, a portion of the computations would be performed on a process on a remote server.</p> <p>Furthermore, as discussed in my report and in the video Viola video 19.avi, <code>vplot</code> can be interchanged with distributed applications such as <code>VIS</code>. See examples below. ViolaWWW works with a distributed application in the same way that it works with an executable application on the client workstation.</p>



VIS

Viola Alpha discloses that the executable application is part of a distributed application. *See, e.g.,* :

Viola's documentation states that "Through a communication facility such as the socket, a message may also be generated from another process on the network." (See, e.g., *violaTOGO\docs\viola\brief.html*.) By way of example, and as discussed elsewhere in this chart, ViolaWWW provides for automatically invoking the *vplot* executable application using the *execv* system call. (See *violaTOGO\src\viola\cl_TTY.c*.) *Vplot* can be implemented as a distributed application wherein a portion of the computations — for example, graphics transformations — can be executed

Claim Text from '985 Patent	Viola Alpha
	<p>on a remote server by using the rexec socket-based system call or other Unix socket-based system calls in lieu of the execv system call. In this case, the executable application is part of a distributed application. Furthermore, as discussed in my report and in Viola video 19.avi, vplot can be interchanged with distributed applications, including VIS. ViolaWWW works with a distributed application in the same way that it works with an executable application on the client workstation.</p>
<p>985-36.j: wherein at least a portion of the distributed application is for execution on a remote network server coupled to the distributed hypermedia network environment.</p>	<p>Viola Alpha discloses that the distributed application executes at least partially on a network server. <i>See, e.g., :</i></p> <p>Viola's documentation states that "Through a communication facility such as the socket, a message may also be generated from another process on the network." (See, e.g., violaTOGO\docs\viola\brief.hmml.) By way of example, and as discussed elsewhere in this chart, ViolaWWW provides for automatically invoking the vplot executable application using the execv system call. (See violaTOGO\src\viola\cl_TTY.c.) Vplot can be implemented as a distributed application wherein a portion of the computations — for example, graphics transformations — can be executed on a remote server by using the rexec socket-based system call or other Unix socket-based system calls in lieu of the execv system call. In this case, a portion of the computations would be performed on a process on a remote server.</p> <p>Furthermore, as discussed in my report and in Viola video 19.avi, vplot can be interchanged with distributed applications such as VIS. ViolaWWW works with a distributed application in the same way that it works with an executable application on the client workstation.</p>
<p>985-37.a: The method of claim 36 where: the information to enable comprises text formats.</p>	<p>Viola Alpha discloses that the enabling information in the file is text formats. <i>See evidence recited for 985-2.a.</i></p>
<p>985-38.a:</p>	<p>Viola Alpha discloses that the text formats are HTML tags. <i>See evidence recited</i></p>

Claim Text from '985 Patent	Viola Alpha
The method of claim 37 where: the text formats are HTML tags.	for 985-3.a.
985-39.a: The method of claim 36 where: the information contained in the file received comprises at least one embed text format.	Viola Alpha discloses that the enabling information in the file includes an embed text format. <i>See</i> evidence recited for 985-4.a.
985-40.a: 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:	Viola Alpha discloses digital information. <i>See</i> evidence recited for 985-20.a. Viola Alpha discloses a network server in a distributed hypermedia environment. <i>See</i> evidence recited for 985-1.a.
985-40.b: 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:	Viola Alpha discloses a client workstation. <i>See</i> evidence recited for 985-1.a. Viola Alpha discloses communicating via network server in order to cause the client workstation to act. <i>See</i> evidence recited for 985-20.b.
985-40.c: 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;	Viola Alpha discloses a browser application; a file containing enabling information received from a server; that the browser displays at least a portion of a distributed hypermedia document; and that the display is in a browser-controlled window. <i>See</i> evidence recited for 985-1.b.
985-40.d: execute, at said client workstation, a browser application, with the browser application:	Viola Alpha discloses a browser application executing on the client workstation. <i>See</i> evidence recited for 985-1.c.
985-40.e: responding to text formats to initiate processing	Viola Alpha discloses parsing text formats. <i>See</i> evidence recited for 985-1.d.

Claim Text from '985 Patent	Viola Alpha
specified by the text formats;	
985-40.f: displaying, on said client workstation, at least a portion of the document within the browser-controlled window;	Viola Alpha discloses displaying at least a portion of the document within the browser-controlled window. <i>See</i> evidence recited for 985-1.e.
985-40.g: 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;	Viola Alpha discloses an object. <i>See</i> evidence recited for 985-36.f. Viola Alpha discloses identifying an embed text format; that the embed text format corresponds to a first location in the hypermedia document; and that the embed text format specifies the location of an object. <i>See</i> evidence recited for 985-1.f.
985-40.h: identifying and locating an executable application associated with the object; and	Viola Alpha discloses that the browser identifies and locates an executable application associated with the object. <i>See</i> evidence recited for 985-36.g.
985-40.i: automatically invoking the executable application, in response to the identifying of the embed text format, in order 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,	Viola Alpha discloses identifying an embed text format. <i>See</i> evidence recited in 985-1.f. Viola Alpha discloses automatic invocation of the executable application; that the executable application displays the object; that the executable application enables direct interaction with the object; and that interaction with the object is at a first location in the hypermedia document. <i>See</i> evidence recited in 985-1.h. Viola Alpha discloses that the object is displayed at a first location within a portion of the hypermedia document being displayed. <i>See</i> evidence recited for 985-24.b. Viola Alpha discloses that a hypermedia document is displayed in a browser window. <i>See, e.g.,</i> evidence recited for 985-1.e.
985-40.j:	Viola Alpha discloses that the executable application is part of a distributed

Claim Text from '985 Patent	Viola Alpha
wherein the executable application is part of a distributed application, and	application. <i>See evidence recited in 985-36.i.</i>
985-40.k: wherein at least a portion of the distributed application is for execution on the network server.	Viola Alpha discloses that the distributed application executes at least partially on a network server. <i>See evidence recited for 985-36.j.</i>
985-41.a: The method of claim 40 where: the information to enable comprises text formats.	Viola Alpha discloses that the enabling information in the file is text formats. <i>See evidence recited for 985-2.a.</i>
985-42.a: The method of claim 41 where: the text formats are HTML tags.	Viola Alpha discloses that the text formats are HTML tags. <i>See evidence recited for 985-3.a.</i>
985-43.a: The method of claim 40 where: the information contained in the file received comprises at least one embed text format.	Viola Alpha discloses that the enabling information in the file includes an embed text format. <i>See evidence recited for 985-4.a.</i>