Exhibit G

Defendants' proposed corresponding structure(s)/act(s) for § 112, ¶ 6

In the Joint Claim Construction Statement (Oct. 29, 2010) [Docket No. 479], Defendants contended that § 112, ¶ 6 applies to certain elements in claims 6–10 and 13–14 of the '906 patent, and certain elements in claims 16–35 and 40–43 of the '985 patent. Exhibit B to the Joint Claim Construction Statement provided Defendants' proposed corresponding structure(s)/act(s). *See* Docket No. 479-2, at 223–41. On January 25, 2011, Eolas informed Defendants that "it will no longer assert the following claims against any Defendant in the above-captioned matter: U.S. Patent No. 5,838,906: Claims 4, 5, 9, and 10; U.S. Patent No. 7,599,985: Claims 12, 13, 14, 15, 32, 33, 34, 35, 44, 45, 46, and 47." Accordingly, reprinted below from Exhibit B of the Joint Claim Construction Statement are the claim elements for the <u>remaining</u> asserted claims that the Defendants contend are governed by § 112, ¶ 6, along with the structure(s) or act(s) that the Defendants contend correspond to those claim elements.

'906 Claim 6	**Corresponding structure(s) or act(s)**
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;	The recited function includes the entire phrase that appears after "computer readable program code for causing said client workstation to". The corresponding structure includes at least the following: • NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A. Some of the modifications to the source code in Appendix A are also described in Figure 7A (flowchart for "HTMLparse" routine in the modified version of HTMLparse.c), Figure 7B (flowchart for routines in the modified version of HTMLformat.c), and Figure 8A (flowchart for "HTMLwidget" routine in the modified version of HTMLwidget.c).
computer readable program code for causing said client workstation to utilize said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client workstation, 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, wherein said object has type information	The recited function includes the entire phrase that appears after "computer readable program code for causing said client workstation to". The corresponding structure includes at least the following: • NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B. Some of the modifications to the source code in Appendix A are also described in Figure 7A (flowchart for "HTMLparse" routine in the modified version of HTMLparse.c), Figure 7B (flowchart for routines in the modified version of HTMLformat.c), and Figure 8A (flowchart for "HTMLwidget" routine in the modified version of HTMLwidget.c). • hypermedia document (212) with the following HTML tag at a "first location" in the

associated with it utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document, and 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 <u>enable an end-user to directly interact with said object</u> 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.	document: <EMBED TYPE = "application/x-vis" HREF = [URL address for data object (216)] WIDTH = [width of window to display the object] HEIGHT = [height of window to display the object]> • data object (216) There is no corresponding structure for at least the following: • "executable application . . . 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"
'906 Claim 7	**Corresponding structure(s) or act(s)**
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.	The recited function includes the entire phrase that appears after "computer readable program code for causing said client workstation to". The corresponding structure includes at least the following: • NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B There is no corresponding structure for at least the following: • "interactively control said controllable application"
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.	The recited function includes the entire phrase that appears after "computer readable program code for causing said client workstation to". The corresponding structure includes at least the following: • NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B There is no corresponding structure for at least the following: • "interactively control said controllable application"
'906 Claim 8	**Corresponding structure(s) or act(s)**
wherein the communications to interactively	The recited function includes the entire phrase

control said controllable application continue to be exchanged between the controllable application and the browser even after the controllable application program has been launched.	that appears after "wherein".
	
	    The corresponding structure includes at least the following:
	    • NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B
	
	    There is no corresponding structure for at least the following:
	    • "interactively control said controllable application"
	
	    The corresponding acts include at least the following:
	    • calling each of the following functions that appear in Appendix B one or more times after the "controllable application program" has been launched: send_client_msg and handle_client_msg

'906 Claim 13	**Corresponding structure(s) or act(s)**

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:
    computer readable program code for causing said client workstation to issue from the client workstation, one or more commands to the network server;
    computer readable program code for causing said network server to execute one or more instructions in response to said commands;
    computer readable program code for causing said network server to send information to said client workstation in response to said executed instructions; and
    computer readable program code for causing said client workstation to process said information at the client workstation to interactively control said controllable application.
	The recited function includes "controlling said controllable application" and each phrase that appears after the clauses "computer readable program code for causing said client workstation to" and "computer readable program code for causing said network server to".
	
	    There is no corresponding structure.

computer readable program code for causing said client workstation to issue from the client workstation, one or more commands to the network server;	The recited function includes the entire phrase that appears after "computer readable program code for causing said client workstation to".

	There is no corresponding structure.
computer readable program code for causing said network server to execute one or more instructions in response to said commands;	The recited function includes the entire phrase that appears after "computer readable program code for causing said network server to". There is no corresponding structure.
computer readable program code for causing said network server to send information to said client workstation in response to said executed instructions; and	The recited function includes the entire phrase that appears after "computer readable program code for causing said network server to". There is no corresponding structure.
computer readable program code for causing said client workstation to process said information at the client workstation to interactively control said controllable application.	The recited function includes the entire phrase that appears after "computer readable program code for causing said client workstation to". There is no corresponding structure.
'906 Claim 14	**Corresponding structure(s) or act(s)**
wherein said additional instructions for controlling said controllable application reside on said client workstation.	The recited function includes "controlling said controllable application". There is no corresponding structure.
'985 Claim 16	**Corresponding structure(s) or act(s)**

software comprising computer executable instructions . . . and when the software is executed operable to:
  receive, at the client workstation from the network server over the network environment, at least one file containing information to enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window;
  cause the client workstation to utilize the browser to:
    respond to text formats to initiate processing specified by the text formats;
    display at least a portion of the document within the browser-controlled window;
    identify an embed text format corresponding to a first location in the document, the embed text format specifying the location of at least a portion of an object external to the file, with the object having type information associated with it;
    utilize the type information to identify and locate an executable application external to the file; and
	The recited function includes the entire phrase that appears after "software comprising computer executable instructions . . . and when the software is executed operable to".
	
	The corresponding structure includes at least the following:
	• NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B. Some of the modifications to the source code in Appendix A are also described in Figure 7A (flowchart for "HTMLparse" routine in the modified version of HTMLparse.c), Figure 7B (flowchart for routines in the modified version of HTMLformat.c), and Figure 8A (flowchart for "HTMLwidget" routine in the modified version of HTMLwidget.c).
	• hypermedia document (212) with the following HTML tag at a "first location" in the document: <EMBED TYPE = "application/x-vis" HREF = [URL address for data object (216)] WIDTH = [width of window to display the object] HEIGHT = [height of window to display the object]>
	• data object (216)

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 browser-controlled window.	There is no corresponding structure for at least the following: • "executable application . . . 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"
'985 Claim 17	**Corresponding structure(s) or act(s)**
claim 16 where: the information to enable comprises text formats.	Same as for claim 16.
'985 Claim 18	**Corresponding structure(s) or act(s)**
claim 17 where: the text formats are HTML tags.	Same as for claim 17.
'985 Claim 19	**Corresponding structure(s) or act(s)**
claim 16 where: the information contained in the file received comprises at least one embed text format.	Same as for claim 16.
'985 Claim 20	**Corresponding structure(s) or act(s)**
communicating via the network server with at least one client workstation over said network in order to cause said client workstation to: receive, over said network environment from said server, at least one file containing information to enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window; execute, at said client workstation, a browser application, with the browser application: responding to text formats to initiate processing specified by the text formats; displaying, on said client workstation, at least a portion of the document within the browser-controlled window; identifying an embed text format which corresponds to a first location in the document, where the embed text format specifies the location of at least a portion of an object external to the file, where the object has type information associated with it;	The recited function includes the entire phrase that appears after "in order to cause said client workstation to". The corresponding acts includes at least the following: • the client workstation launches NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B (hereinafter the "browser application"). Some of the modifications to the source code in Appendix A are also described in Figure 7A (flowchart for "HTMLparse" routine in the modified version of HTMLparse.c), Figure 7B (flowchart for routines in the modified version of HTMLformat.c), and Figure 8A (flowchart for "HTMLwidget" routine in the modified version of HTMLwidget.c). • the browser application retrieves over the network from the network server the hypermedia document (212) with the following HTML tag at a "first location" in the document: <EMBED TYPE = "application/x-vis" HREF = [URL address for data object (216)] WIDTH = [width of window to display the object] HEIGHT =

utilizing the type information to identify and locate an executable application external to the file; and automatically invoking the executable application, in response to the identifying of the embed text format, to execute on the client workstation in order to display the object and enable an end-user to directly interact with 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.	[height of window to display the object]> • the browser application performs the steps in Figure 7A (e.g., parsing the hypermedia document to identify the <EMBED> tag • the browser application performs the steps in Figure 7B (e.g., initialize the drawing area) • the browser application performs the steps in Figure 8A to identify and locate an executable application using the information TYPE = "application/x-vis" found in the <EMBED> tag There is no corresponding act for at least the following: • "the browser application . . . automatically invoking the executable application . . . 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"
'985 Claim 21	**Corresponding structure(s) or act(s)**
The method of claim 20 where: the information to enable comprises text formats.	Same as for claim 20.
'985 Claim 22	**Corresponding structure(s) or act(s)**
The method of claim 21 where: the text formats are HTML tags.	Same as for claim 21.
'985 Claim 23	**Corresponding structure(s) or act(s)**
The method of claim 20 where: the information contained in the file received comprises at least one embed text format.	Same as for claim 20.
'985 Claim 24	**Corresponding structure(s) or act(s)**
A method for running an executable application in a computer network environment . . . the method comprising: 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, wherein said network environment is a distributed hypermedia environment, wherein said client workstation receives, over said network environment from said	The recited function includes the entire phrase that appears after "the method comprising:". The corresponding acts includes at least the following: • the client workstation launches NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B (hereinafter the "browser application"). Some of the modifications to the source code in Appendix A are also described in Figure 7A (flowchart for "HTMLparse" routine in the modified version of HTMLparse.c), Figure 7B (flowchart for routines in the modified version of HTMLformat.c), and

6

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,	Figure 8A (flowchart for "HTMLwidget" routine in the modified version of HTMLwidget.c).
wherein said executable application is external to said file,	• the browser application retrieves over the network from the network server the hypermedia document (212) with the following HTML tag at a "first location" in the document: <EMBED TYPE = "application/x-vis" HREF = [URL address for data object (216)] WIDTH = [width of window to display the object] HEIGHT = [height of window to display the object]>
wherein said client workstation executes the browser application, with the browser application responding to text formats to initiate processing specified by the text formats,	• the browser application performs the steps in Figure 7A (e.g., parsing the hypermedia document to identify the <EMBED> tag
wherein at least said portion of the document is displayed within the browser-controlled window,	• the browser application performs the steps in Figure 7B (e.g., initialize the drawing area)
wherein an embed text format which corresponds to said first location in the document is identified by the browser,	• the browser application performs the steps in Figure 8A to identify and locate an executable application using the information TYPE = "application/x-vis" found in the <EMBED> tag
wherein the embed text format specifies the location of at least a portion of said object external to the file,	
wherein the object has type information associated with it,	There is no corresponding act for at least the following:
wherein the type information is utilized by the browser to identify and locate said executable application, and	• "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"
wherein the executable application is automatically invoked by the browser, in response to the identifying of the embed text format.	
'985 Claim 25	**Corresponding structure(s) or act(s)**
25. The method of claim 24 where: the information to enable comprises text formats.	Same as for claim 24.
'985 Claim 26	**Corresponding structure(s) or act(s)**
26. The method of claim 25 where: the text formats are HTML tags.	Same as for claim 25.
'985 Claim 27	**Corresponding structure(s) or act(s)**
27. The method of claim 24 where: the information contained in the file received comprises at least one embed text format.	Same as for claim 24.
'985 Claim 28	**Corresponding structure(s) or act(s)**
software comprising an executable application . . . operable to: cause the client workstation to display an object and enable an end-user to directly interact with said object while the object is	The recited function includes the entire phrase that appears after "software comprising an executable application . . . operable to". The corresponding structure includes at least

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, wherein said network environment is a distributed hypermedia environment, wherein said client workstation receives, over said network environment from said server, at least one file containing information to enable said browser application to display, on said client workstation, at least said portion of said distributed hypermedia document within said browser-controlled window, wherein said executable application is external to said file, wherein said client workstation executes said browser application, with the browser application responding to text formats to initiate processing specified by the text formats, wherein at least said portion of the document is displayed within the browser-controlled window, wherein an embed text format which corresponds to said first location in the document is identified by the browser, wherein the embed text format specifies the location of at least a portion of said object external to the file, wherein the object has type information associated with it, wherein the type information is utilized by the browser to identify and locate said executable application, and wherein the executable application is automatically invoked by the browser, in response to the identifying of the embed text format.	the following: • NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B. Some of the modifications to the source code in Appendix A are also described in Figure 7A (flowchart for "HTMLparse" routine in the modified version of HTMLparse.c), Figure 7B (flowchart for routines in the modified version of HTMLformat.c), and Figure 8A (flowchart for "HTMLwidget" routine in the modified version of HTMLwidget.c). • hypermedia document (212) with the following HTML tag at a "first location" in the document: <EMBED TYPE = "application/x-vis" HREF = [URL address for data object (216)] WIDTH = [width of window to display the object] HEIGHT = [height of window to display the object]> • data object (216) There is no corresponding structure for at least the following: • "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"
'985 Claim 29	**Corresponding structure(s) or act(s)**
29. The method of claim 28 where: the information to enable comprises text formats.	The recited function includes the entire phrase that appears after "software comprising an executable application . . . operable to". The corresponding structure includes at least the following: • NCSA Mosaic version 2.4 for X-Windows

8

with the modifications to the source code shown in Appendix A and Appendix B (hereinafter the "browser application"). Some of the modifications to the source code in Appendix A are also described in Figure 7A (flowchart for "HTMLparse" routine in the modified version of HTMLparse.c), Figure 7B (flowchart for routines in the modified version of HTMLformat.c), and Figure 8A (flowchart for "HTMLwidget" routine in the modified version of HTMLwidget.c).

• hypermedia document (212) with the following HTML tag at a "first location" in the document: <EMBED TYPE = "application/x-vis" HREF = [URL address for data object (216)] WIDTH = [width of window to display the object] HEIGHT = [height of window to display the object]>

• data object (216)

The corresponding acts includes at least the following:

• the client workstation launches the browser application

• the browser application retrieves over the network from the network server the hypermedia document (212) with the following HTML tag at a "first location" in the document: <EMBED TYPE = "application/x-vis" HREF = [URL address for data object (216)] WIDTH = [width of window to display the object] HEIGHT = [height of window to display the object]>

• the browser application performs the steps in Figure 7A (e.g., parsing the hypermedia document to identify the <EMBED> tag

• the browser application performs the steps in Figure 7B (e.g., initialize the drawing area)

• the browser application performs the steps in Figure 8A to identify and locate an executable application using the information TYPE = "application/x-vis" found in the <EMBED> tag

There is no corresponding act for at least the following:

• "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

9

	location within a portion of a hypermedia document being displayed in a browser-controlled window"
'985 Claim 30	**Corresponding structure(s) or act(s)**
30. The method of claim 29 where: the text formats are HTML tags.	Same as for claim 29.
'985 Claim 31	**Corresponding structure(s) or act(s)**
31. The method of claim 28 where: the information contained in the file received comprises at least one embed text format.	Same as for claim 29.
'985 Claim 40	**Corresponding structure(s) or act(s)**
communicating via the network server with at least one remote client workstation over said computer network environment in order to cause said client workstation to: receive, over said computer network environment from the network server, at least one file containing information to enable a browser application to display at least a portion of a distributed hypermedia document within a browser-controlled window; execute, at said client workstation, a browser application, with the browser application: responding to text formats to initiate processing specified by the text formats; displaying, on said client workstation, at least a portion of the document within the browser-controlled window; identifying an embed text format which corresponds to a first location in the document, where the embed text format specifies the location of at least a portion of an object; identifying and locating an executable application associated with the object; and 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,	The recited function includes the entire phrase that appears after "in order to cause said client workstation to:" The corresponding acts includes at least the following: • the client workstation launches NCSA Mosaic version 2.4 for X-Windows with the modifications to the source code shown in Appendix A and Appendix B (hereinafter the "browser application"). Some of the modifications to the source code in Appendix A are also described in Figure 7A (flowchart for "HTMLparse" routine in the modified version of HTMLparse.c), Figure 7B (flowchart for routines in the modified version of HTMLformat.c), and Figure 8A (flowchart for "HTMLwidget" routine in the modified version of HTMLwidget.c). • the browser application retrieves over the network from the network server the hypermedia document (212) with the following HTML tag at a "first location" in the document: <EMBED TYPE = "application/x-vis" HREF = [URL address for data object (216)] WIDTH = [width of window to display the object] HEIGHT = [height of window to display the object]> • the browser application performs the steps in Figure 7A (e.g., parsing the hypermedia document to identify the <EMBED> tag • the browser application performs the steps in Figure 7B (e.g., initialize the drawing area) • the browser application performs the steps in Figure 8A to identify and locate an executable application using the information TYPE = "application/x-vis" found in the <EMBED> tag

wherein the executable application is part of a distributed application, and wherein at least a portion of the distributed application is for execution on the network server.	There is no corresponding act for at least the following: • "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, wherein the executable application is part of a distributed application, and wherein at least a portion of the distributed application is for execution on the network server."
'985 Claim 41	**Corresponding structure(s) or act(s)**
41. The method of claim 40 where: the information to enable comprises text formats.	Same as for claim 40.
'985 Claim 42	**Corresponding structure(s) or act(s)**
42. The method of claim 41 where: the text formats are HTML tags.	Same as for claim 41.
'985 Claim 43	**Corresponding structure(s) or act(s)**
43. The method of claim 40 where: the information contained in the file received comprises at least one embed text format.	Same as for claim 40.