

EXHIBIT K

Expert Report on Invalidity

ADOBE et al v. EOLAS

Richard L. Phillips

July 20, 2011

Table of contents

| | |
|--|----|
| Table of contents | 1 |
| Appendices | 7 |
| I. Appendix A: CV | 7 |
| II. Appendix B: Claim chart exhibits | 7 |
| III. Appendix C: Video Exhibits | 11 |
| IV. Appendix D: Declaration of Dan Sadowski | 11 |
| Expert report by Richard L. Phillips on Invalidity | 12 |
| I. Introduction | 12 |
| 1. EXPERT QUALIFICATIONS | 12 |
| 2. GENERAL TECHNICAL EXPERIENCE | 12 |
| 3. SOFTWARE DEVELOPMENT EXPERIENCE | 14 |
| II. Information Considered In Forming My Opinions | 16 |
| 1. PATENTS IN SUIT AND DECLARATIONS | 16 |
| 2. USPTO PROSECUTION HISTORIES | 16 |
| 3. PUBLICATIONS AND OTHER REFERENCES | 17 |
| 4. SOFTWARE | 21 |
| 5. COMPUTERS | 23 |
| 6. WWW-TALK PUBLICATIONS AND OTHER EMAILS | 25 |
| 7. VIDEOS | 29 |
| 8. LITIGATION MATERIALS | 29 |
| 9. OTHER | 29 |
| III. Tools Available To A Person of Ordinary Skill By October'93 | 30 |

certain functionality. My review of the documents above does not show documents that provide the adequate disclosure missing from the patent specification. To the extent the claim limitations are interpreted to cover the functionality described above in connection with Section 112, it is my opinion that the documents above do not support conception, reduction to practice, or diligence of those claim limitations before the filing date of the '906 patent.

X. Section 112

796. Under the Patent Law, 35 U.S.C. § 112, the specification of a patent must contain, "a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor of carrying out his invention."

797. I am informed that the written description requirement must be satisfied by the specification. To satisfy the written description requirement, which is an objective test, the specification as filed with the USPTO must clearly allow persons of ordinary skill in the art to recognize that the inventor invented what is claimed.

798. I have reviewed the specification of the '906 and '985 patents (the specification for both patents is the same) and considered it in view of the asserted claims.

799. It is my opinion that the specification does not contain a written description of the inventions as claimed in the asserted claims and as interpreted by Eolas. A summary of the reasons for my opinion is provided below.

a. The Specification Lacks An Adequate Written Description Of An "Embed Text Format" Other Than One Implemented Using Special Tags

800. Having reviewed the specification, I find that it lacks support for an "embed text format." The specification refers to special tags or symbols (see '906 at

col. 14), and during prosecution, the inventors characterized the embed text format as the EMBED tag and attempted to distinguish over the prior art by arguing that it is a "special tag" not found in the prior art. *See e.g.*, '906 patent at 12:54–13:36; '985 patent at 14:27; Applicants' Response, at 22 (Feb. 5, 2009) (identifying the EMBED tag disclosed in the specification as support for the "embed text format"); Declaration of Edward W. Felten, at ¶ 21–25 (Sept. 27, 2007) (accompanying Applicants' Response (Sept. 27, 2007)) ("In the claimed '906 system, the browser instead used a special tag, the "embed text format", to specify that an embedded object should be included. Mosaic lacked the embed text format."); *see also* Notice of Intent to Issue a Reexam Certificate, at 8–9 (Sept. 27, 2005) ("'interactive processing' is invoked . . . in response to the browser application parsing an 'embed text format' (i.e., an 'EMBED' tag, see col. 12, line 60, '906 patent) . . ."). Regardless of what exactly the embed text format is, the specification does not provide adequate disclosure of an "embed text format" implemented using scripts, such as code written in JavaScript, as opposed to a special tag or symbol that is simply parsed by the browser application.

801. The term "embed text format" is not found in the original specification, nor is it a term of art generally known to a person skilled in the art at the time of the invention. Rather, it was coined by Eolas added by amendment long after the first patent filing. *See* Applicants' Response, at 1–2 (Aug. 6, 1996); Doyle, Tr: 319:15–17 ("Q: No, nothing specific. I'm just asking if -- did embed text format, was that a term of art or is that a term that you coined? A: That's a term that we coined in our -- in our -- or that we used in our patent specification.").

802. As I noted above, the example of an "embed text format" linked by Eolas to the disclosure in the specification is that of an "embed tag" in HTML. (*See* Figure 7B). This example is shown in Table II, ('906 Col. 12, ln. 58–65; '985 Col. 12, ln. 60–67):

```
<EMBED  
    TYPE = "type"
```

```
HREF="href"  
WIDTH=width  
HEIGHT=height  
>
```

803. The "embed tag" shown above, and envisioned by the specification, is a short pre-defined tag in standard HTML format with angle brackets and attributes and contained within an HTML document. The embed tag has a TYPE attribute that specifies the same MIME type information that web browsers at the time, such as Mosaic, used to identify helper applications, a HREF attribute, which is a standard HTML attribute commonly used at the time to specify the location of a file, and WIDTH and HEIGHT attributes, which were standard HTML attributes commonly used at the time to specify the width and height of an image.

804. I understand that Eolas now asserts that an "embed text format" is "text that initiates processing" and has accused code contained in a HTML document and written in JavaScript, a client-side scripting language that is interpreted by the browser, of satisfying the requirement of an "embed text format."

805. It is my view that the specification does not support the use of client-side scripting language, such as JavaScript, to implement the "embed text format." Among other reasons, explained below, I say this because the use of a scripting language in this way, and as HTML documents are interpreted in modern browsers, implies a much higher level functionality than the simple inline processing and output of predecessor web browsers — at a time when webpages themselves were much less sophisticated. Modern web browsers use a special interpreter for processing client-side scripts, such as scripts written in JavaScript, and render the HTML document in a tree-like structure called the DOM tree, which is also not disclosed or suggested by the specification. The JavaScript is often used to manipulate the DOM tree so that the web browser can render the DOM as the

content shown in the web browser. This idea is not taught or suggested in the specification.

806. JavaScript itself was not available before the filing date of the '906 patent. JavaScript was developed by an employee at Netscape and first introduced in beta releases of Netscape Navigator 2 in December 1995. *See Netscape and Sun Announce JavaScript, the Open, Cross-Platform Object Scripting Language for Enterprise Networks and the Internet*, December 4, 1995, available at:

<http://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html>; *Innovators on the Net: Brendan Eich and JavaScript*, June 24, 1998, available at:

http://web.archive.org/web/20080208124612/http://wp.netscape.com/comprod/columns/techvision/innovators_be.html.

807. HTML is a simple markup language with generic semantics that are used to represent information on a document, whereas a client-side scripting language is a programming language with entirely different syntax that allows a web page to have different and changing content depending upon user input, environmental conditions and events, or other variables. *See, e.g., W3C, Client-side Scripting and HTML*, March 14, 1997, available at: <http://www.w3.org/TR/WD-script-970314>. A client-side scripting language can also be used to modify the HTML document "as it is being parsed." *Id.*

808. The Mosaic HTML parser modified by the inventors and described in their preferred embodiment was not capable of parsing HTML documents so they could be modified as they were being parsed.

809. The specification does not demonstrate to a person of ordinary skill in the art that the inventors contemplated or fully possessed, as of the filing date, an invention wherein the "embed text format" comprises a script, including JavaScript, as opposed to merely representing an HTML tag as disclosed in the specification.

Thus, the written description is not satisfied by the specification as this limitation has been interpreted by Eolas.

b. The Specification Lacks An Adequate Written Description Of An "Embed Text Format" Other Than One Located At The "First Location"

810. I further find that the specification lacks support for an "embed text format" other than one located at the "first location" where the object is displayed. Each of the asserted claims of the '906 patent requires that the "embed text format [is] located at a first location in [the] first distributed hypermedia document," and that the object is "display[ed] . . . within a display area created at said first location within the portion of said first distributed hypermedia document being displayed . . ."

811. As discussed above, the only embodiment of an "embed text format" disclosed in the specification is the EMBED tag of Table II. *See* '906 col. 12:56-65. This tag does not include an attribute that specifies where the display area will appear in the displayed document. Like other HTML tags, the object is inserted in the same location as the tag itself. For example, the specification describes that the inventors modified the source code of the publicly available Mosaic browser, version 2.4, so that when the browser parsed an HTML file and identified the EMBED tag it would automatically launch VIS without a user clicking on a link, and the output from VIS would be displayed within the browser window at the location of the EMBED tag that was found in the HTML file. *See, e.g.,* '906 patent, 14:9-16:46. There is no disclosure that the output from VIS would be displayed at a location other than the location of the EMBED tag found in the HTML file. It is a simple in-line processing of the input stream of text from the HTML file where the location of each element identified within the stream alone determines its location in the browser window.

812. During prosecution and the two reexaminations, the applicants attempted to distinguish their invention over the prior art by arguing that the prior

art did not teach an embed text format that is located at the same location as where the object is displayed. *See, e.g.*, Applicants' Response, at 11 (June 2, 1997) ("Further, [in Mosaic] a display window is not created in the first hypermedia document at the location in the document of the embed text format as required by the claim."); Declaration of Edward W. Felten, at ¶¶ 51-52 (Sept. 27, 2007) (accompanying Applicants' Response (Oct. 1, 2007)) ("[Cohen's] LDESC tags cannot be the embed text format, because they do not satisfy the required claim element 'wherein said first distributed hypermedia document includes an embed text format, located at a first location . . .' This claim element requires that the embedded object be displayed at a location in the distributed hypermedia document (e.g., the Web page) that corresponds to the location of the embed text format within the document. . . . The LDESC tag does not appear in the document at the required location. Instead, the LDESC (link description) tag appears in the document file's prologue . . .").

813. Each of the asserted claims of the '985 patent requires that the "embed text format . . . corresponds to a first location in the document." During prosecution of the '985 patent, the applicants similarly attempted to distinguish their invention over the prior art by arguing that the prior art did not teach an embed text format that is located at the same location as where the object is displayed. *See, e.g.*, Applicants' Response, at 18 (March 11, 2005) ("Further, there is no teaching in NoteMail of parsing an embed text format at a first location and displaying and enabling interactive processing within the first location because, in NoteMail, the location of information is specified elsewhere, by the 'Format' data type."). When asked by the Examiner to identify support for the newly amended claims that included the "correspond to" language, the applicants did not acknowledge that the "corresponds to" language was a new element that required support, and simply listed as support the specification's EMBED tag — which as discussed earlier is located at the same location as where the object is displayed.

814. It is, therefore, my opinion that one of ordinary skill in the art cannot reasonably conclude, based on the specification of the patents-in-suit, that the inventors possessed, as of the filing date, an invention wherein the "embed text format" is located at a location other than one located at the "first location" where the object is displayed. Thus, the written description is not satisfied by the specification as this limitation has been interpreted by Eolas.

c. The Specification Lacks An Adequate Written Description Of An "Embed Text Format" That Is Identified By A Method Other Than Parsing

815. The specification disclosure for identifying an "embed text format" is limited to a simple in-line parsing of an HTML file and the attributes of the EMBED tag. It does not disclose that the identification of an "embed text format" can be accomplished by any method other than a straight in-line parsing of a stream of text.

816. For example, the specification describes that the inventors added routines in the HTMLparse.c file of Mosaic that checks whether each item parsed (e.g. word, tag or symbol) from a hypermedia document is an EMBED tag. If an EMBED tag is identified, a standard routine in HTMLParse.c is called to assign to the tag an enumerated type, which is an identifier with a unique integer associated with it. Once all of the text in the hypermedia document has been parsed, routines in a separate HTMLformat.c file are called to create an internal object representation of the EMBED tag with values for the "width and height of a window on the display to contain an image, position of the window, style, URL of the image object, etc." *See, e.g., '906 patent, 14:9-16:46.*

817. To the extent Eolas argues that parsing is not required to identify an "embed text format," the specification does not provide a written description of such a technique to show that the inventors were in possession of such an invention, nor does the specification provide sufficient disclosure to a person of ordinary skill in the art of an "embed text format" that is identified by any method other than parsing.