831 PH Ex. 1

SIDLEY AUSTIN BROWN & WOOD LLP

BEIJING
BRUSSELS
CHICAGO
DALLAS
GENEVA
HONG KONG
LONDON

1501 K STREET, N.W.
WASHINGTON, D.C. 20005
TELEPHONE 202 736 8000
FACSIMILE 202 736 8711
www.sidley.com
FOUNDED 1866

Part of
Paper #31

LOS ANGELES
NEW YORK
SAN FRANCISCO
SHANGHAI
SINGAPORE
TOKYO
WASHINGTON, D.C.

WRITER'S DIRECT NUMBER
(202) 736-8914

WRITER'S E-MAIL ADDRESS
jkushan@sidley.com

October 22, 2003

By Hand Delivery

Stephen G. Kunin
US Department of Commerce
US Patent & Trademark Office
P/DCPEP
Crystal Park 2, 9th Floor
Arlington, VA

Re: <u>WWWTALKQ2 and WWWTALKQ3</u>

Dear Mr. Kunin:

Enclosed please find a compact disk with the above files on it.

Please feel free to contact me if you have any questions.

Sincerely,

Jeffrey P. Kushan /tnh

Jeffrey P. Kushan

JPK:tnh

Enclosure

Y2per #31

October 14, 2003

Mail Stop: ___ (Citation of Prior Art per 37 CFR 1.501)
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RE: Submission of Information Pursuant to 35 U.S.C. § 301 and 37 CFR 1.501 in relation to U.S. Patent No. 5,838,906

Dear Sir,

Enclosed, please find prior art publications to be included in the file wrapper of U.S. Patent No. 5,838, 906 ("the '906 patent") pursuant to 35 U.S.C. § 301 and 37 C.F.R. § 1.501.

The '906 patent claims Web browser computer programs and processes. The two references provided herewith are printed publications published more than one year prior to the filing date of the '906 patent. Each is prior art under 35 U.S.C. § 102(b) to the '906 patent. The two printed publications provided herewith were not cited, made of record or considered during the prosecution of the '906 patent. One set of copies is provided for inclusion in the file wrapper of the '906 patent. The second set of copies is provided to permit service by the Office on the patent owner.

One or more claims of the '906 patent are *prima facie* anticipated and/or obvious by the art being cited herein. The '906 patent also has gained significant notoriety in the Internet community because, due to its invalidity, it will unfairly and significantly impact a very wide audience of consumers and other users of the Internet. Examples of press coverage illustrating these concerns are enclosed as Attachment D. The undersigned represent a significant cross-section of the community of developers of Internet-related software who share these concerns, and believe the Director should, on his initiative, commence a reexamination of the '906 patent.

Existence of a Substantial New Question of Patentability

The '906 patent generally relates to the ability of a Web browser to handle an "object" in a Web page having a data format not natively supported by the Web browser and thus requiring an external "executable application" to display the object. The '906 patent acknowledges that certain prior art Mosaic browsers allowed users, through clicking on a link, to view and interact with such an object via a "helper application" in a separate window. In such prior art browsers, in response to a user's click on a link, the browser invokes the helper application is invoked to display the object in a separate window. Pursuant to the claims of the '906 patent, in response to the inclusion of an "EMBED text format," or tag, in the document, the browser automatically invokes the helper application to display the object "in-line" in the browser window.

The two enclosed references describe and relate to characteristics of Web browsers for implementing HTML standards. They are dated more than one year prior to the filing date of the '906 patent. They each describe the use of an EMBED tag to automatically invoke an external executable application in order to display and enable interactivity with an object in-line within the browser window. They each also inherently describe Web browsers including, in particular, the admitted prior art Web browsers of record. As such, each publication describes each claimed element of the inventions defined by at least claims 1 to 3 and 6 to 8 of the '906 patent and as such each publication anticipates these claims of the '906 patent. Alternatively, the newly cited printed publications, when considered in view of the admitted prior art Web browsers of record, render *prima facie* obvious the claimed subject matter of at least claims 1 to 3 and 6 to 8 of the '906 patent. As such, the two enclosed references each raise a substantial new question of patentability regarding the '906 patent.

Acknowledged Prior Art

The '906 patent acknowledges that Web browser computer programs were in the prior art. See, e.g., column 2, lines 9 to 12, which provides: "An example of a browser program is the National Center for Supercomputing Application's (NCSA) Mosaic software developed by the University of Illinois at Urbana/Champaign, Ill." More specifically, the inventors of the '906 patent indicate that the subject matter claimed as their invention concerns modifications of certain acknowledged prior art Web browser programs. See, e.g., column 8, lines 9 to 12, of the patent specification, which provides:

> "[t]he source code in Appendix A includes NCSA Mosaic version 2.4 source code <u>along with modifications to the source code to implement the present invention</u>" (emphasis added);

and column 13, lines 43 to 46 which provides:

> "that much of the source code in is [sic] pre-existing NCSA Mosaic code. Only those portions of the source code that relate to the new functionality discussed in this specification should be considered as part of the invention."

The inventors thus acknowledge that the features of Web browsers, at least to the degree reflected in version 2.4 of the NCSA Mosaic Web browser, are prior art to the claimed inventions.

Version 2.4 of the NSCA Mosaic Web browser, like all Web browsers, is a computer program that is implemented on and operated using a computer. The Mosaic program is designed to and preferably runs on a computer connected to the Internet to allow the user to retrieve documents over the Internet and display those documents on the computer. Such documents may contain "an icon, or other indicator, within the text" linked to a particular image file (column 2, lines 64 to 65) that users "may select ... to obtain the full image" (column 3, lines 2 to 3). As the '906 patent admits, when a user selects such an indicator, the Mosaic program "retrieves the corresponding full image ... and displays it by using external software" (column 3, lines 5 to 6) "in a separate window" (column 3, line 17). See generally column 2, line 56 through column 3, line 26 of the '906 patent where the patent describes the capabilities of the Mosaic browser, among others.

Differences Between the Claimed Invention and the Acknowledged Prior Art

The differences between the claims and the acknowledged prior art are nominal. Specifically, independent claims 1 and 6 require the computer program/process to process an "EMBED text format," or tag, which is used to automatically display, and enable interaction with, an external object within the browser document window via an external application. The '906 patent asserts that this was an improvement over the prior "helper application" technology employed by prior art browsers such as the Mosaic program in which the browser interprets a user selection of an embedded link to launch an associated external program in a separate window for data that the browser could not process natively. See generally column 3, lines 2 to 20 of the '906 patent.

The patent disclosure and claims specify that the EMBED functionality is expressed in terms of a tag that "specifies the location of … an object," having "type information associated with it utilized by the browser to identify and locate an executable application," where the tag is parsed by the browser "to automatically invoke said executable application … in order to display said object and enable interactive processing of said object" in the browser window. See, in particular, Table II of the '906 patent, appearing at column 12, line 54, along with the descriptive text associated with the table appearing at column 13, line 31. These portions of the specification of the '906 patent show that the preferred embodiment of the claimed invention involves use of an EMBED tag having an HREF attribute for specifying the location (e.g., a uniform resource locator, or URL) of an object to be displayed and a TYPE attribute for the MIME type of the object data, which the browser uses to identify, locate and launch an associated application to render that data.

Prior Art Being Submitted Herewith

1. <u>David Raggett, HTML+ (Hypertext markup language) (July 23, 1993) (hereinafter "*Raggett I*").</u>

Raggett I ("A proposed standard for a light weight presentation independent delivery format for browsing and querying information across the Internet") describes and discloses the functionality of Web browsers that comply with the draft HTML+ specification as of July 23, 1993 (i.e., more than one year before the filing date of the '906 patent). In particular, at page 6, lines 43 to 45, *Raggett I* indicates that such browsers must parse and process "the EMBED tag" contained within a document retrieved over the Internet. *Raggett I* discloses that the EMBED tag includes a TYPE attribute with information concerning the type of the embedded object data. The TYPE attribute, according to *Raggett I*, uses the well-known MIME protocol to enable the browser to identify, locate and invoke an external program to display foreign object data within the document being rendered ("the *type* attribute specifies a registered MIME content type and is used by the browser to identify the appropriate shared library or external filter to use to render the embedded data, e.g., by returning a pixmap"). As is the case with all other HTML tags described in *Raggett I*, the browser performs the related operations for the disclosed EMBED tag automatically upon parsing the tag, without user input.

According to *Raggett I*, "embedding" (page 6, line 40) of an object means displaying the object within the document being rendered. For example, *Raggett I* shows the use of the

EMBED tag to invoke an external program to display an equation or graphic directly in the display of the HTML-based Web page (see, page 6), and also discusses the use of the EMBED tag in combination with the FIG tag to display, for instance, "simple graphs etc. defined in an external format" (page 12, line 30) in the document being rendered and allow for "control of picture alignment and text flow" (page 12, line 17) among other things. See also, generally, page 12, line 13 to page 14, line 6. At page 6, line 47, *Raggett I* further discloses the use of external editor programs that allow for interaction with the displayed object data within the document ("Sophistocated [sic] browsers can link to external editors for updating and revising embedded data"). The '906 patent discloses a comparable TYPE attribute of an EMBED tag (Table II) and use of the MIME protocol for matching the type information to an external program for displaying foreign data within a Web browser window as is described in *Raggett I.*

The above-recited publication was widely disseminated in 1993 by and to, among others, the leaders in the efforts to standardize the Internet, who later became founding participants in the WWW Consortium (or "W3C", the leading standard-setting organization for the Internet). The publication was, has been and continues to be available to all interested persons through the Internet and through other means since on or prior to July 23, 1993.[1] As such, it is a "printed publication" within the meaning of 35 U.S.C. § 102(b). See M.P.E.P. § 2128 (2003).[2] The effective date of the printed publication is the date of its availability; namely, at least as early as July 23, 1993. See M.P.E.P. § 2128.[3] See also, the enclosed declaration from *Raggett I*'s author, David Raggett, which further authenticates the content and date of availability of the publication.

2. Posting of Dave Raggett, dsr@hplb.hpl.hp.com, to www-talk@nxoc01.cern.ch (June 14, 1993) (posting to WWW-Talk public mailing list) (hereinafter "*Raggett II*").

Raggett II is an email posting to the WWW-Talk email list (a public, archived and indexed discussion forum) by the author of *Raggett I* (the HTML+ draft specification) that was published on June 14, 1993.[4] It specifically discusses the implementation of the EMBED tag operation disclosed in the draft specification and further notes, in the "p.s.," that the foreign data that is to be rendered in-line by the external editor program need not be contained in the Web document, but may also be external to the document, referenced by a URL. (Compare the '906 patent, e.g., column 13, lines 27 to 28 ("HREF specifies a URL address as discussed above for a data object.").) In addition to repeating the operative description of the EMBED tag operations

[1] For example, a dated copy of the document currently can be retrieved from the Cite Seer: Scientific Research Digital Library site via http://citeseer.nj.nec.com/raggett93html.html. Also, dated entries in the WWW-TALK archives related to the referenced provisions of the HTML+ specification, as well as the original posting of the July 23rd HTML+ specification, are still available on-line today at http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q2.messages/467.html and http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q3.messages/282.html.

[2] M.P.E.P. § 2128 provides, in the section entitled "ELECTRONIC PUBLICATIONS AS PRIOR ART: Status as a 'Printed Publication,'" that: "An electronic publication, including an on-line database or Internet publication, is considered to be a 'printed publication' within the meaning of 35 U.S.C. 102(a) and (b) provided the publication was accessible to persons concerned with the art to which the document relates."

[3] M.P.E.P. § 2128 provides, in the section entitled "ELECTRONIC PUBLICATIONS AS PRIOR ART: Date of Availability," that: "Prior art disclosures on the Internet or on an on-line database are considered to be publicly available as of the date the item was publicly posted. If the publication does not include a publication date (or retrieval date), it cannot be relied upon as prior art under 35 U.S.C. 102(a) or (b)."

[4] The complete archives of the WWW-talk email list for the second and third quarters of 1993 are provided on the enclosed CD. The complete archives, or the individual posting, are each printed publications.

from the HTML+ specification, the body of the posted Raggett email provides guidance regarding how to connect a MIME type via an EMBED tag to the appropriate external rendering program ("e.g. via X resources or a config file") and regarding use of external programs and inter-process communications ("separate programs driven via pipes and stdin/stdout or as dynamically linked library modules (Windows DLLs)").

The above-recited publication was widely disseminated and publicly available through the Internet and through other means at least from June 14, 1993, and continues to be available on-line at http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q2.messages/ 467.html. It is thus a "printed publication" within the meaning of 35 U.S.C. §102(b) because it was a "contribution" to "electronic bulletin boards, message systems, and discussions lists" that were "accessible to the persons concerned with the art to which the document relates" when it was posted to the WWW-Talk list (see, e.g., M.P.E.P. § 707.05(e)).[5] It enjoys prior art effect as from the date of its posting (i.e., June 14, 1993), pursuant to M.P.E.P. § 2128, as noted above.

Comparison of the Claims to the Acknowledged and Newly Cited Art

In the context of independent claims 1 and 6, the NCSA Mosaic version 2.4 browser is a "computer program product" (e.g., a Web browser) that is "embodied" in a "computer usable medium" (e.g., installed in a computer or contained on a disk) for use in a "distributed hypermedia environment" having "at least one client workstation and one network server" (e.g., the Internet). The Mosaic program can run on "said client workstation" to "parse[] a first distributed hypermedia document" (e.g., an HTML document) "received over" the Internet to "identify text formats" (e.g., HTML tags and elements) and "respond[] to predetermined text formats to initiate processing specified by said text formats" in the hypermedia document in order "to display" the document in a browser window on "said client workstation." Furthermore, the Mosaic program can locate "an external object" having "type information associated with it utilized by said browser to identify and to locate an executable application external to" said hypermedia document. The Mosaic program can "invoke" said external application (e.g., an "external editor") "to display" the "external object." As implemented in version 2.4, said invocation and display occurs via another window (as opposed to within the browser window displaying the hypermedia document as required by the claims) when the user selects a hyperlink to the external object (as opposed to "automatically" as required by the claims). Version 2.4 of Mosaic also enables "interactive processing of" (e.g., editing of) the "external object." See, e.g., column 6, lines 32 to 35 of the '906 patent (i.e., prior art browsers permit some degree of interactive processing of the external object).

The only claim limitation not explicitly disclosed, described and implemented in the admitted prior art Mosaic Web browser is the proviso requiring the Web browser to parse an "embed text format" in a hypermedia document to "automatically invoke" an external application "to display" an external object within the browser window displaying the hypermedia

[5] For instance, a review of the University of Calgary archive site containing this posting demonstrates that more than 1,000 such postings were made during the three months surrounding the posting of the July 23rd HTML+ Specification (*Raggett I*) by the very people that were developing the Internet at the time. (See <http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q3.index.html>.) Moreover, the HTML+ Specification itself asks that comments be sent "to the WWW discussion group: www-talk@nxoc01.cern.ch." (*Raggett I* at page 1, footnote 1.)

document. *Raggett I* (i.e., the draft HTML+ specification), however, specifically describes just such an HTML "embed" tag for such purposes (i.e., automatically invoking an external program to render interactive objects in-line in an HTML document). This is reflected in the HTML+ specification and in the specification author's contemporaneous email to the WWW-Talk email list, both of which demonstrate that it was well-known in the browser field prior to the filing date of the '906 patent that the foreign data could be contained in a separate file referenced, for example by a URL. Moreover, the ability of a Web browser to retrieve and process data from both local and non-local sources is the inherent design of such browsers. Indeed, one of the first applications of HTML/Web browsers was the rendering, in a single document, of text and image files, where the image files were located in a file external to the file containing the text to be rendered.

An element by element comparison of claim 6[6] to the acknowledged and newly cited prior art is provided below in Table I. It should be noted that, as described in greater detail below, *Raggett I* and *II* each inherently describe each feature of the NCSA Mosaic version 2.4 Web browser, which is acknowledged by the owner of the '906 patent to be prior art.

Table I

Claim 6	**Acknowledged and Newly Cited Prior Art**
A computer program product for use in a system having at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment, the computer program product comprising: a computer usable medium having computer readable program code physically embodied therein, said computer program product further comprising: computer readable program code for causing said client workstation to execute a browser application	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (describing Internet, and use and function of browser programs, and noting that Mosaic is "an example of a browser program").
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;	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (same); *Raggett I* at page 3, lines 4 to 38 (discussing "Parsing HTML+ Documents").
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,	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (same).
wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client workstation,	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (same).
wherein said first distributed hypermedia	Mosaic, see '906 patent at column 1, line 19 to

[6] Note that claims 1 and 6 are nearly identical but for the type of invention (i.e., claim 1 claims a process, whereas claim 6 is directed to a "computer program product for use in…").

Table I

Claim 6	Acknowledged and Newly Cited Prior Art
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,	column 3, line 51 (same, including: "A distributed hypertext or hypermedia document typically has many links within it that specify many different data objects located in computers at different geographical locations connected by a network."); *Raggett II* at pages 1-2 (providing example of embedded text format and stating that: "The browser identifies the format of the embedded date from the "type" attribute, specified as a MIME content type;" and that "you can also put the foreign data in a separate file referenced by a URL").
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	Mosaic, see '906 patent at column 3, lines 5 to 6 (the Mosaic program "retrieves the corresponding full image ... and displays it by using external software") (emphasis added); *Raggett II* at page 1 (providing example of embedded text format and stating that: "The browser identifies the format of the embedded data from the "type" attribute, specified as a MIME content type;" and that "The functions could be implemented as separate programs ...") (emphasis added).
and wherein said embed text format is parsed by said browser to automatically invoke said executable application on said client workstation	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (noting that Mosaic is "an example of a browser program" and, as such, parses HTML documents accessed); *Raggett I* at page 3, lines 4 to 38 and page 6, lines 40 to 45 (discussing "Parsing HTML+ Documents" generally, and "the EMBED tag" specifically, as part of the initial processing of every HTML document accessed by a Web browser); *Raggett II* at page 1 (providing example of embedded text format and stating: "The browser identifies the format of the embedded data from the "type" attribute, specified as a MIME content type.").
in order to display said object	Mosaic, see '906 patent at column 3, lines 5 to 6 (the Mosaic program "retrieves the corresponding full image ... and displays it by using external software").
and enable interactive processing of said object	Mosaic, see '906 patent at column 6, lines 32 to 35 ("Also, while the present open distributed hypermedia system on the Internet allows users to locate and retrieve data objects it allows users very little, if any, interaction with these data objects."); *Raggett I* at page 6, line 47 ("Sophistocated [sic] browsers can link to external editors for updating and revising embedded data.").

Table I	
Claim 6	**Acknowledged and Newly Cited Prior Art**
within a display area	Mosaic, see '906 patent at column 3, lines 5 to 6 (the Mosaic program "retrieves the corresponding full image … and displays it by using external software").
created at said first location within the portion of said first distributed hypermedia document being displayed in said first browser-controlled window.	*Raggett I* at page 6, lines 40 to 45 and page 12, line 13 to page 14, line 6 (discussing various options when displaying embedded objects in-line, such as text wrapping around the object) and at page 34, lines 1 to 20 (in section entitled "Notes for Implementers" stating: "It is generally better to avoid displaying the retrieved document in a new window, unless explicitly requested by the user …."); *Raggett II* at page 1 ("Well both of these will be possible with the HTML+ DTD, by using the capability to embed foreign formats inline in the HTML+ source …") (emphasis added).

The Newly Cited References Anticipate Claims 1, 2, 3, 6, 7 and 8

As shown above, *Raggett I* and *II* each fully disclose the allegedly novel features of claims 1 and 6; namely, the use of an EMBED tag to automatically invoke an external application to display an external object inline within the same browser window displaying the document containing the EMBED tag. The remaining limitations of claims 1 and 6 are all admitted by the inventors of the '906 patent to be disclosed in prior art Web browsers such as Mosaic. See column 8, lines 9 to 12 and column 13, lines 43 to 46. Those same prior art Web browsers are inherently disclosed and described by *Raggett I* and by *Raggett II*, making each reference fully anticipatory.

Raggett I and *II* each refer to Web browsers that are acknowledged to be prior art in the '906 patent (see, e.g., *Raggett I*, page 15, lines 43 to 45). The inherent features and characteristics of such Web browsers, such as Mosaic, include the ability to render HTML-compliant documents. HTML is the predecessor standard to the HTML+ specification that is the basis of the *Raggett I* and *II* disclosures. The set of elements that make up the HTML specification is found in its entirety in, and is added to by, the HTML+ specification. Both HTML and HTML+ are implementations of the Standard Generalized Markup Language (SGML). Consequently, references in *Raggett I* and *II* to prior art Web browsers inherently are described by the disclosure of HTML in *Raggett I* and *II*.

Moreover, those of skill in the browser coding art, upon reading *Raggett I* and *II*, would immediately infer the inclusion of such prior art browsers in the teachings of these two disclosures. See M.P.E.P. § 2144.01 (2003) ("[I]n considering the disclosure of a reference, it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom.") (quoting *In re Preda*, 401 F.2d 825, 826, 159 USPQ 342, 344 (CCPA 1968)). This stems from the fact that

Raggett I and *II* each define and describe the functional and other characteristics of computer programs that are HTML+ compliant Web browsers. The discussion in *Raggett I* and *II* concerning new features that prior art browsers should be modified to incorporate *necessarily* includes a full description of the prior art Web browsers themselves. See *Atlas Powder Co. v. Ireco, Inc.*, 190 F.3d 1342, 1346, 51 USPQ2d 1943, 1946 (Fed. Cir. 1999) ("[A] prior art reference may anticipate when the claim limitation or limitations not expressly found in that reference are nonetheless inherent in it. Under the principles of inherency, if the prior art necessarily functions in accordance with, or includes, the claimed limitations, it anticipates.") (internal citations omitted).

Included -- through explicit references and inherently due to the fact that the HTML+ specification builds upon and expands the original HTML specification -- in the disclosure of the HTML+ specification by *Raggett I* and *II* is the original HTML specification See, e.g., *Raggett I* at page 2, line 3 ("HTML+ follows on from an earlier standard - HTML. see [Berners-Lee 93a]."), at page 3, line 40 ("This format is designed to be largely compatible with the earlier format HTML.") and at page 33, lines 1 to 37 (discussing compatibility with HTML, for example, by listing and describing each obsolete tag from HTML and how to map to HTML+). Because the HTML+ specification, like the earlier HTML specification, describes the functionality that Web browsers must possess to be fully compliant with the specification, one of skill in the art would immediately "envisage" both the prior art Web browsers that support HTML and the modified versions of those browsers that comply with the new HTML+ specification. See M.P.E.P. § 2131.02 (in chemical context, stating that a reference may be relied upon for what one of skill in the art would "at once envisage" upon reading the reference).

Particularly when they are considered in light of their inherent disclosures of admitted prior art Web browsers, *Raggett I* and *II* disclose and therefore anticipate each claimed limitation of claims 1 and 6 of the '906 patent. Furthermore, as claims 2, 3, 7 and 8 recite only inherent features present in prior art Web browsers, these claims add no further limitations relative to claims 1 and 6 that would distinguish them from the anticipating disclosures of *Raggett I* and *II*.

Claims 1, 2, 3, 6, 7 and 8 are Also *Prima Facie* Obvious Over the Prior Art

As set forth above, claims 1, 2, 3, 6, 7 and 8 are anticipated by *Raggett I* and *II*. In the alternative, these claims are *prima facie* obvious when the acknowledged prior art is taken in view of *Raggett I* and *Raggett II* because these disclosures specifically suggest modifying the prior art to incorporate the differences between the claims and the acknowledged prior art.

The Level of Ordinary Skill in the Art for Purposes of Obviousness

The person of ordinary skill in the relevant art to the claimed invention is a software programmer. The '906 patent acknowledges that the act of modifying the Mosaic prior art browser to implement the functionalities described and claimed in the patent was well within the skill of the art. For example, at column 13, lines 51 to 59, the patent states:

> "In general, the flowcharts in this specification illustrate one or more software routines executing in a computer system such as computer system 1 of FIG. 1. The routines may be implemented by any means as is known in the art. For example, any number of

> computer programming languages, such as 'C', Pascal, FORTRAN, assembly language, etc., may be used. Further, various programming approaches such as procedural, object oriented or artificial intelligence techniques may be employed."

In addition, at column 16, lines 51 to 53, the patent specifies that:

> "It will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the appended claims. For example, various programming languages and techniques can be used to implement the disclosed invention. … Many such changes or modifications will be readily apparent to one of ordinary skill in the art."

Thus, based on the admissions within the '906 patent, a software programmer could readily implement the noted functionality into the acknowledged prior art Mosaic Web browser, the source code for which was readily available (also as acknowledged in the '906 patent specification).

The *Prima Facie* Obviousness of Claims 1 and 6

The printed publications provided herewith were not considered by the PTO during the original prosecution of the '906 patent. When they are considered in view of the acknowledged prior art (e.g., the version 2.4 Mosaic Web browser) by a person of ordinary skill in the art, they render the claimed invention defined by claims 1 and 6 of the patent *prima facie* obvious.[7]

As noted above, the differences between the claimed invention and the acknowledged prior art Mosaic version 2.4 Web browser are limited to the Web browser parsing an "embed text format" in a hypermedia document (e.g., an HTML document) to "automatically invoke" an external application "to display" an external object within the browser window displaying the hypermedia document. *Raggett I* and *Raggett II* each specifically disclose implementing this functionality in Web browsers.

The two printed publications provided herewith thus provide specific motivation and guidance to a person of ordinary skill to modify the acknowledged prior art NCSA Mosaic version 2.4 browser (and other prior art browsers) to arrive at the claimed invention. Indeed, for a Web browser to be fully complaint with *Raggett I* (the HTML+ specification), which was publicly disseminated more than a year prior to the filing date of the '906 patent, the Web browser must possess the functionality disclosed therein. As such, it is difficult to envision a document that could provide more specific motivation to modify prior art Web browsers to provide the disclosed functionality. Furthermore, as acknowledged and admitted by the inventors of the '906 patent (e.g., column 13, lines 51 to 59 and column 16, lines 51 to 53), the act of modifying the Mosaic prior art browser to implement the features called for by *Raggett I* was well within the abilities of a person having an ordinary level of skill in the relevant art (e.g.,

[7] Pursuant to M.P.E.P. §2143 (2003), "[t]o establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations."

software programming). Thus, modification of prior art Web browsers (e.g., NCSA Mosaic version 2.4) by such a person to implement the functionalities described in *Raggett I* or in *Raggett II* would have been *prima facie* obvious to a person of ordinary skill in the art. . Further comparison of the '906 patent specification to *Raggett I* and *Raggett II* confirms this conclusion. As noted above, Table II (column 12, line 54, with descriptive text through column 13, line 31) of the '906 patent shows the preferred embodiment of an EMBED tag with HREF and TYPE attributes which the browser uses to identify, locate and launch associated external applications. The EMBED tag TYPE and HREF attributes, and their descriptions, disclosed in Table II of the '906 patent and the surrounding text are nearly identical to the EMBED tag TYPE attribute disclosed in *Raggett I* (page 6, lines 43 to 46) and to the HREF attribute disclosed elsewhere in *Raggett I* (compare '906 patent at column 13, lines 27 to 28 ("HREF specifies a URL address as discussed above for a data object."), with *Raggett I*, page 13, line 23 (defining HREF as: "A URL specifying the link to traverse when clicked.")). The enclosed publications thus disclose not only the same functionality but precisely the same means of implementing the same functionality in Web browsers (i.e., the same "EMBED" tag is used to initiate the same browser behavior that provides the same results as the claimed subject matter of the '906 patent).

Moreover, the enclosed publications enable, as the '906 patent claims, Web browsers to provide the user with more functionality (e.g., through displaying and/or editing new data formats) without changing the browser code. Compare, '906 patent, column 11, lines 52 to 55, *Raggett I*, page 6, and *Raggett II*, cover page. As noted above, the enclosed publications were promulgated to the WWW community more than a year before the filing of the '906 patent for the purpose of implementing this very same capability in prior art Web browsers.

Claims 2, 3, 7 and 8 are *Prima Facie* Obvious

Claims 2 and 7 of the '906 patent add an additional limitation to claims 1 and 6 respectively; namely, that the process or computer program provide for "interactively controlling" the external application "via inter-process communications" between the browser and the external application. The patent specification indicates that "inter-process communications" are a "protocol to exchange information between browser client and application client", and exemplify such communications by referring to the prior art "XEvent interprocess communication protocol" (column 9, line 8 to 10). See also column 16, lines 29 to 32, wherein the '906 patent discusses how "the browser process, Mosaic, communicates with the [external application] process via inter-client communications mechanisms such as provided in the X-Window environment." (Emphasis added.) The added limitations specified in claims 2 and 7 thus refer to characteristics and properties of the acknowledged prior art.

As noted above, claims 1 and 6 are *prima facie* obvious over the acknowledged prior art Mosaic version 2.4 Web browser when taken in view of *Raggett I* and *II*, independently and in combination. The acknowledged prior art, along with *Raggett I* and *II*, also disclose the additional limitation of claims 2 and 7 as noted above. For example, *Raggett I* discloses that "[s]ophistocated [sic] browsers can link to external editors for updating and revising embedded data" (see page 6, line 47). Similarly, *Raggett II* notes that such "separate programs" (e.g., "external editors") can be "driven via pipes and stdin/stdout" (see cover page). An "external editor" is, by definition, a controllable external application, and "pipes and stdin/stdout" is an example of "inter-process communications" for use in transferring data between, among other programs, a browser and an external application.[8] *Raggett I* and *II*, thus, clearly disclose the additional limitation of claims 2 and 7 and provide specific motivation to one of ordinary skill in the art to modify the NCSA Mosaic version 2.4 Web browser to incorporate the above-noted claimed features. Also as noted above, the '906 patent indicates that a person of ordinary skill in the art has the requisite abilities to implement such features (e.g., column 13, lines 51 to 59 and column 16, lines 51 to 53). Claims 2 and 7, thus, are *prima facie* obvious when the acknowledged prior art NCSA Mosaic version 2.4 Web browser is taken in view of *Raggett I* and *II*, considered individually or collectively.[9]

Claims 3 and 8 add a further limitation calling for "the communications to ... continue to be exchanged between the controllable application and the browser even after the controllable application program has been launched." Similar to the discussion in footnote 9 above, this limitation, however, adds nothing to claims 2 and 7 (or even claims 1 and 6) of the '906 patent. To interactively control an external application, as each of claims 1, 2, 6 and 7 requires, the communications between the browser and the external application must continue after the external application is launched. Claims 3 and 8 thus add no patentable distinction and, for the reasons provided above in relation to claims 1, 2, 6 and 7, are *prima facie* obvious in the light of the acknowledged Mosaic prior art browser in combination with *Raggett I* and *II*.

* * *

[8] "Pipes are IPC (interprocess communication) features of the UNIX, Windows, and OS/2 operating systems." See <http://www.linktionary.com/p/pipes.html> (Tom Sheldon's Linktionary.com, an online networking dictionary).

[9] This is not surprising given that the specification of the '906 patent admits that the additional limitation of claims 2 and 7 is a simple use prior art network capability for its intended purpose (column 9, lines 12 to 13 (X-Windows)). Moreover, at a fundamental level, the '906 patent effectively concedes that this limitation cannot render the otherwise obvious claims 1 and 6 patently distinct. Independent claims 1 and 6 already include a limitation requiring the "external application" to "enable interactive processing" of the external object. In other words, claim 1 and 6 inherently include the "interactively controlling ... via inter-process communications" limitation. After all, to "enable interactive processing" (claims 1 and 6), there must be some type of "inter-process communications" between the browser and an "interactively controll[ed]" external application (claims 2 and 7). The additional limitation of claims 2 and 7, if it can even be called a limitation, is therefore an empty one that merely parrots limitations already included in the underlying independent claims 1 and 6, and thus is certainly as obvious as the underlying independent claims.

In conclusion, the two printed publications provided herewith anticipate at least claims 1, 2, 3 and 6, 7 and 8 of the '906 patent. The acknowledged prior art, when taken in view of the newly cited prior art provided herewith also provide specific motivation and guidance to a person of ordinary skill to modify the NCSA Mosaic version 2.4 browser to arrive at the claimed invention. As such, these disclosures render claims 1, 2, 3, 6, 7 and 8 of the '906 patent *prima facie* obvious to a person of skill in the art.

Very truly yours,

James Brimson
Associate General Counsel
America Online, Inc.

Loren Hillberg
Senior Vice President and General Counsel
Macromedia, Inc.

Andrew Culbert
Associate General Counsel
Microsoft Corporation

In conclusion, the two printed publications provided herewith anticipate at least claims 1, 2, 3 and 6, 7 and 8 of the '906 patent. The acknowledged prior art, when taken in view of the newly cited prior art provided herewith also provide specific motivation and guidance to a person of ordinary skill to modify the NCSA Mosaic version 2.4 browser to arrive at the claimed invention. As such, these disclosures render claims 1, 2, 3, 6, 7 and 8 of the '906 patent *prima facie* obvious to a person of skill in the art.

Very truly yours,

James Bramson
Chief Counsel
America Online, Inc.

Loren Hillberg
Senior Vice President and General Counsel
Macromedia, Inc.

Andrew Culbert
Associate General Counsel
Microsoft Corporation

Adobe

Adobe Sy[redacted]s Incorporated
345 Park Avenue
San Jose, CA 95110-2704
Phone 408 536.6000
Fax 408 537.6000

15 October 2003

Commissioner for Patents
Attention: Hon. Steven Kunin
Deputy Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RE: Potential Director-Ordered Reexamination of U.S. Patent No. 5,838,906 pursuant to 35 U.S.C. §303(a)

Dear Deputy Commissioner Kunin:

As a leading company in the software industry, we are writing to you with regard to U.S. Patent No. 5,838,906, Doyle. We urge the Director of the United States Patent and Trademark Office to exercise his authority pursuant to 35 U.S.C. §303(a), and initiate a Director Ordered Reexamination of it. We have reviewed the Criteria for Initiating a Director Ordered Reexamination, dated August 3, 2000, and, while we agree that such reexamination orders should be rare, we believe the present circumstances regarding the Doyle patent meet the stringent criteria.

In particular, we believe that (a) there is "compelling reason" to order reexamination, and (b) at least one claim in the Doyle patent is *prima facie* unpatentable over patents or printed publications. With regard to critera (b), it has come to our attention that patents or publications have been cited to you under the provisions of 35 U.S.C. §301. It is our further understanding that such art raises a substantial new question of patentability, sufficient to justify a reexamination of said Doyle patent.

By this letter, we would like to focus your attention on the first criteria, in particular, the "compelling reason" requirement. Specifically, we believe that "a significant concern about the patentability of the claimed subject matter has been expressed by a substantial segment of the industry, and that there is substantial media publicity adverse to the patent alleging conspicuous unpatentability of the claims."

The Doyle patent has been the subject of widespread concern within the industry to which it pertains. That community includes, in particular, companies, organizations, and individuals that develop web browsers and technology solutions that work within web browsers. In addition, significant concerns have been expressed within the broader community of owners and users of web sites on the Internet regarding changes that would have to be implemented in web browsers to avoid infringing the Doyle patent. Further still, the negative implications of the Doyle patent have been the subject of significant media publicity. In support of this, we direct your attention to recent news articles appearing in the software trade press that discuss the widespread impact of changes to the Internet.

We would note that the Doyle patent is the subject of litigation in the Northern District of Illinois, brought by the assignee, Eolas, Inc. against Microsoft Corp., and that Microsoft has been found to have infringed the current claims. Furthermore, Microsoft has recently announced that they will make changes to their browser to deal with this alleged infringement, and that such changes will affect an enormous segment of the Internet-using community.

Accordingly, we believe that the rare circumstances justifying a Director-ordered Reexamination of the Doyle patent have been met. As it is our understanding that it is your authority to review potential Director-Ordered Reexaminations on behalf of the Director, and make recommendations to him with regard to ordering them, we respectfully request that you favorably consider such a request and recommend to the Director that he order the reexamination of U.S. Patent No. 5,838,906.

Respectfully submitted,

MeMe Jacobs Rasmussen
Associate General Counsel
Adobe Systems Incorporated

A

08324443.021204

831 PH Ex. 2

O I P E JC64
OCT 24 2003
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

TC 2100, CPK2, 2Y11

Patent No: 5,838,906

Issued: November 17, 1998

For: Distributed Hypermedia Method for Automatically Invoking External Application Providing Interaction and Display of Embedded Objects within a Hypermedia Document

RECEIVED
JAN 2 9 2004
Technology Center 2100

**CITATION OF PRIOR ART UNDER 35 U.S.C. § 301 AND 37 CFR 1.501
IN RELATION TO U.S. PATENT NO. 5,838,906**

Mail Stop: Prior Art Department (Citation of Prior Art per 37 CFR 1.501)
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RECEIVED
OCT 2 9 2003
OFFICE OF PETITIONS

Dear Sir,

On behalf of the World Wide Web Consortium, the primary standard-setting organization for the World Wide Web,[1] please find enclosed two prior art publications to be included in the file wrapper of U.S. Patent No. 5,838,906 ("the '906 patent") pursuant to 35 U.S.C. § 301 and 37 C.F.R. §1.501. The enclosed publications are prior art to the '906 patent under 35 U.S.C. § 102(b). They were never considered by the United States Patent & Trademark Office during the prosecution of the '906 patent. These publications, taken alone, anticipate at least claims 1, 2, 3, 6, 7 and 8 of the '906 patent, and, taken together with the Mosaic browser that was acknowledged in the patent as prior art, plainly render those claims invalid as obvious under 35 U.S.C. § 103.

As the Commissioner may be aware, the '906 patent is the subject of a patent infringement suit brought by Eolas Technologies, Inc. and the Regents of the University of California (the patent's exclusive licensee and owner, respectively) against Microsoft Corporation. The suit alleged that Microsoft's Internet Explorer, the most widely used program in the world for browsing the World Wide Web, infringed claims of the '906 patent. A jury in that case recently found against Microsoft and awarded Eolas and the University of California in excess of $500 million. Microsoft is appealing that verdict, but has also stated publicly that it intends in any event to redesign Internet Explorer in a manner that it believes plainly does not infringe the '906 patent. Although Microsoft's proposed redesign, as we understand it, involves only a small portion of

[1] The World Wide Web is a network of information resources that can be accessed through the Internet. A list of the member companies of the World Wide Web Consortium is available at http://www.w3.org/Consortium/Member/List.

Internet Explorer, it would render Microsoft's browser incompatible with globally-accepted standards and impair the operation of millions of Web pages. The cost to the larger World Wide Web community of fixing the problems created by such a change to Internet Explorer is incalculable, but would likely require changes to millions of Web pages, as well as changes to Web page authoring tools and other software and systems designed for the World Wide Web. This enormous expense and attendant incalculable disruption, not to mention the threat the '906 patent as construed by the court poses to other browsers widely used in the Web community, are completely unwarranted because we strongly believe that the '906 patent is invalid in view of prior art, submitted herewith, that was never previously considered by the United States Patent & Trademark Office. While we understand that the submitted prior art was introduced during the course of the recent trial proceedings, the issue of whether it renders the '906 patent invalid was never considered.[2] In view of the pervasive negative impact of the '906 patent on the larger World Wide Web community, which is unwarranted in view of the patent's invalidity, the World Wide Web Consortium believes that the Director should, on his initiative, commence a reexamination of the '906 patent.

The '906 patent is generally directed to a Web browser able to invoke external programs to display portions of a Web page that the browser cannot directly display itself. A Web browser may not be capable of displaying certain types of image data, for example, in which case the browser would invoke a separate program that is capable of doing so. The sole difference between the web browser described in the '906 patent and typical browsers that the patent acknowledges as prior art, is that with prior art browsers, the image in such cases is displayed in its own window, separate from the main browser window, whereas, with the '906 browser the image is displayed in the same window as the rest of the Web page, without the need for a separate window. But that feature (i.e., displaying, or embedding, an image generated by an external program in the same window as the rest of a Web page) had already been described in the prior art publications submitted herewith and was known to the Web development community. The claims of the '906 patent are therefore plainly obvious in view of this prior art.

Even prior to the development of this feature in Web browsers, software developers had recognized the usefulness of adding the same functionality to prior art word processing programs, which display documents instead of Web pages. For example, more than a year before the '906 patent was filed, a word processing program called Write, provided with Microsoft Windows 3.1, enabled users to embed into Write documents graphic images created with the Paint program. The Write program would invoke the Paint program to display the illustration within the same window as the rest of the document. The '906 patent thus added nothing to the art — it only applied a well known concept in the display of documents to the display of Web pages, and even then, did so after the enclosed Raggett publications had disclosed the same thing for web pages.

The two enclosed references are printed publications published more than one year prior to the filing date of the '906 patent. Each is therefore prior art to the '906 patent under 35 U.S.C. § 102(b). Neither reference was cited, made of record or considered during the prosecution of the '906 patent. One set of copies is provided for inclusion in the file wrapper of the '906 patent. The second set of copies is provided to permit service by the Office on the patent owner.

[2] We understand the court entered a judgment as a matter of law that other prior art (but not the two Raggett publications) differed from the claimed subject matter and that the issue of invalidity over the Raggett publications was not put to the jury or otherwise considered.

The Raggett I and Raggett II Publications

The two enclosed publications relate to HTML+, a proposed specification extending the features of Hypertext Markup Language ("HTML"), the standard language in which Web pages were, and still are, written. The first publication ("*Raggett I*," Exhibit A hereto) is a draft of the HTML+ specification, which was made publicly available for comment on July 23, 1993. *Raggett I* was authored by Dave Raggett, a researcher at Hewlett Packard Laboratories, who attempted in that document to pull together comments regarding extensions to HTML from the participants in www-talk, a public mailing list hosted by Tim Berners-Lee, the founder of the Web and now the Director of the World Wide Web Consortium. The second publication ("*Raggett II*," Exhibit B hereto) is a message posted on June 14, 1993 to the public www-talk mailing list, describing the EMBED tag in HTML+. The EMBED tag described in *Raggett I* and *II* is identical in all material respects to the EMBED tag described in the '906 patent, which in turn was the basis for its claims.

As described in *Raggett I*, the EMBED tag enables a browser to display in-line (i.e. without going to a separate browser window) information rendered by an external application or external shared library. That is, it enabled the browser to display the information rendered by the external application, or shared library, in the same window displaying the information rendered by the browser. (*Raggett I*, p. 6, last para.). The example given in *Raggett I* is the display by a browser of an equation rendered using EQN, a program that formats and displays mathematical equations:

<embed type="application/eqn">2 pi int sin (omega t)dt </embed>

Specifically, in this example, "*2 pi int sin (omega t)dt*" is the embedded data to be rendered as a formatted equation and "*type="application/eqn"*" specifies the external program, EQN, capable of rendering that data. *Raggett I* also described using the EMBED tag in combination with the FIG tag in order to display in-line images having data formats that were not recognized by the browser. (*Raggett I*, p. 12).

The particular external program, or shared library, that must be used to render the data in the EMBED tag is identified by the TYPE attribute of the EMBED tag. *Raggett I* used the well-known MIME protocol to identify, locate and invoke an external program or shared library capable of rendering data of the specified type. (*See id.* ("the *type* attribute specifies a registered MIME content type and is used by the browser to identify the appropriate shared library or external filter to use to render the embedded data, e.g., by returning a pixmap")). As is the case with all other HTML tags described in *Raggett I*, the browser performs the related operations for the disclosed EMBED tag automatically upon parsing the tag, without user input. *Raggett I* further disclosed the use of external editor programs that allow users to <u>interact</u> with the displayed object data within the document. (*See id.* ("Sophistocated [sic] browsers can link to external editors for updating and revising embedded data")). The '906 patent discloses a comparable TYPE attribute of an EMBED tag (Table II) and use of the MIME protocol for matching the type information to an external program for displaying foreign data within a Web browser window, precisely as earlier described in *Raggett I*.

Raggett II further explained that the embedded, or "foreign," data that is to be rendered in-line does not need to be contained within the EMBED tag, as in the example in *Raggett I*, but may instead be located in a separate file referenced by a URL. (*See Raggett II*, last sentence). A URL, or Uniform Resource Locator, specifies the location of a file anywhere on the Internet. In addition, *Raggett II* repeated the operative description of the EMBED tag operation from *Raggett I* and provided multiple suggestions for implementing the EMBED tag operation. For example, it explained how to bind a MIME type to the appropriate external rendering program ("e.g. via X resources or a config file") and provided suggestions for implementing the external programs (for example, via "separate programs driven via pipes and sdin/stdout or as dynamically linked library modules (Windows DLLs)").

Raggett I also explained that HTML+, including the EMBED tag, is "for use within the World Wide Web" and, in particular, that "[i]nformation browsers can display information . . . in HTML+ format." *Raggett I* at page 1. It further explained that the World Wide Web is a client-server environment in which hypermedia documents are retrieved across the Internet. *Raggett I* at page 1 ("The World Wide Web is a wide area client-server architecture for retrieving hypermedia documents across the Internet.").

Raggett I was widely disseminated in 1993 by and to, among others, the leaders in the effort to standardize the World Wide Web, including the founding participants in the World Wide Web Consortium, again today's leading standard-setting organization for the World Wide Web. The publication was, has been and continues to be available to all interested persons through the Internet and through other means since on or prior to July 23, 1993. As such, it is a "printed publication" within the meaning of 35 U.S.C. §102 (b). *See* M.P.E.P. § 2128 (2003) (stating, in a section entitled "ELECTRONIC PUBLICATIONS AS PRIOR ART: Status as a 'Printed Publication'" that: "An electronic publication, including an on-line database or Internet publication, is considered to be a 'printed publication' within the meaning of 35 U.S.C. 102(a) and (b) provided the publication was accessible to persons concerned with the art to which the document relates."). The effective date of the printed publication is the date of its availability; namely, at least as early as July 23, 1993. *See* M.P.E.P. § 2128 (stating, in section entitled "ELECTRONIC PUBLICATIONS AS PRIOR ART: Date of Availability" that: "Prior art disclosures on the Internet or on an on-line database are considered to be publicly available as of the date the item was publicly posted. If the publication does not include a publication date (or retrieval date), it cannot be relied upon as prior art under 35 U.S.C. 102(a) or (b)."). A dated copy of the document currently can be retrieved from the Cite Seer: Scientific Research Digital Library site via http://citeseer.nj.nec.com/raggett93html.html (a pdf version of *Raggett I*, which can be viewed using Adobe Acrobat, can be retrieved by clicking on the "PDF" hyperlink located in the upper right corner of the Web page). Also, dated entries in the WWW-TALK archives relating to provisions of the HTML+ specification, as well as the original posting of the July 23, 1993 HTML+ specification, are currently available on-line at http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q2.messages/467.html and http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q3.messages/282.html.

Raggett II was also widely disseminated and publicly available through the Internet and through other means at least since June 14, 1993, and is currently available on-line at http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q2.messages/467.html. It is a "printed publication" within the meaning of 35 U.S.C. §102(b) because it was a "contribution" to "electronic bulletin boards, message systems, and discussion lists" that were "accessible to the persons concerned with the art to which the document relates" when it was posted to the WWW-Talk list (see, e.g., M.P.E.P. §§ 707.05(e), 2128).[3] It enjoys prior art effect as of the date of its

[3] For instance, a review of the University of Calgary archive site containing this posting demonstrates that more than 1,000 such postings were made during the three months surrounding the posting of the July 23rd HTML+ Specification (*Raggett I*) by the very people that were

posting (i.e., June 14, 1993), pursuant to M.P.E.P. § 2128 (see, e.g., "ELECTRONIC PUBLICATIONS AS PRIOR ART: Date of Availability").

The NSCA Mosaic Web Browser and Other Acknowledged Prior Art

The '906 patent acknowledges that Web browsers were in the prior art and in fact describes its alleged invention in terms of modifications to one such prior art browser, the NCSA Mosaic browser, Version 2.4. *See, e.g.*, '906 patent, column 3, lines 9 to 12 (stating that "An example of a browser program is the National Center for Supercomputing Application's (NCSA) Mosaic software developed by the University of Illinois at Urbana/Champaign, Ill."); *see also id.*, column 8, lines 9 to 12 ("[t]he source code in Appendix A includes NCSA Mosaic version 2.4 source code along with modifications to the source code to implement the present invention")(emphasis added); *id.*, column 13, lines 43 to 46 (stating "that much of the source code in is [sic] pre-existing NCSA Mosaic code" and that "[o]nly those portions of the source code that relate to the new functionality discussed in this specification should be considered as part of the invention."). The patent thus acknowledges that the features of Web browsers, at least to the degree reflected in version 2.4 of the NCSA Mosaic Web browser, were prior art to the claimed inventions.

NSCA Mosaic Web browser, version 2.4, like all Web browsers, is a computer program that enabled users to retrieve documents over the Internet and display those documents on a computer monitor. Such documents may contain, for example, "an icon, or other indicator, within the text" linked to a particular image file that users "may select … to obtain the full image." (See '906 patent, column 2, line 64 to 65, column 3, lines 2 to 3). When a user selects such an indicator, the Mosaic program "retrieves the corresponding full image … and displays it by using external software" "in a separate window." (*Id.*, column 3, lines 5-7, 16-18; *see also* column 2, line 56 through column 3, line 26 (describing the capabilities of the Mosaic browser, among others).

Differences Between the Claimed Invention and the Prior Art

The sole difference between claims 1 and 6[4] and the NCSA Mosaic browser, Version 2.4, is that the claims require a browser to process a so-called "embed text format," and the Mosaic browser did not have this capability as claimed. In particular, the claimed browser must process an "embed text format" that specifies the location of an "object external" to a hypermedia document (*i.e.*, a document of the type typically displayed by browsers, containing text as well as non-text portions such as graphics, video, sound, etc.). The browser in turn utilizes "type information" associated with the external object to identify, locate and automatically invoke an external "application" that enables the browser to display the object within the hypermedia document being displayed in a browser-controlled window. The '906 patent asserts that the "embed text format" is an improvement over the "helper application" technology employed by prior art browsers such as

developing the World Wide Web at the time. (See <http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q3.index.html>.) Moreover, the HTML+ Specification itself asks that comments be sent "to the WWW discussion group: www-talk@nxoc01.cern.ch." (*Raggett I* at page 1, footnote 1.)

[4] Note that claims 1 and 6 are nearly identical but for the type of invention (i.e., claim 1 claims a process, whereas claim 6 is directed to a "computer program product for use in…").

the Mosaic program in which the browser launched an external program in a separate window to display data that it cannot process natively. *See, e.g.*, '906 patent, column 3, lines 2 to 20.

The '906 patent describes the "embed text format" functionality in terms of an EMBED tag. *See, in particular,* '906 patent, column 12, line 54 and Column 13, line 31, Table II and descriptive text. The described EMBED tag has an HREF attribute for specifying the location (e.g., a uniform resource locator, or URL) of an object to be displayed and a TYPE attribute for the MIME type of the object data, which the browser uses to identify, locate and launch an associated application to render that data.

In the context of independent claims 1 and 6, the NCSA Mosaic browser, version 2.4, is a "computer program product" (e.g., a Web browser) that is "embodied" in a "computer usable medium" (e.g., installed in a computer or contained on a disk) for use in a "distributed hypermedia environment" having "at least one client workstation and one network server" (e.g., the Internet). The Mosaic program can run on "said client workstation" to "parse[] a first distributed hypermedia document" (e.g., an HTML document) "received over" the Internet to "identify text formats" (e.g., HTML tags and elements) and "respond[] to predetermined text formats to initiate processing specified by said text formats" in the hypermedia document in order "to display" the document in a browser window on "said client workstation." Furthermore, the Mosaic program can locate "an external object" having "type information associated with it utilized by said browser to identify and to locate an executable application external to" said hypermedia document. The Mosaic program can "invoke" said external application (e.g., an "external editor") "to display" the "external object." As implemented in Mosaic version 2.4, that invocation led to the invoked object being displayed in another window, as opposed to within the browser window displaying the hypermedia document as required by the claims, when the user selected a hyperlink to the external object (as opposed to "automatically" as required by the claims).[5]

The only claim limitation not explicitly disclosed, described and implemented in the admittedly prior art Mosaic browser is the "embed text format" feature, in which a browser "automatically invoke[s]" an external application "to display" an external object within the browser window displaying the hypermedia document. That feature, however, is plainly disclosed in *Raggett I* and *Raggett II* — they specifically describe a substantially identical HTML "embed" tag for automatically invoking an external program to render interactive objects in-line in an HTML document. *Raggett II*, in particular, specifically stated that external, or foreign, data (*i.e.*, an external object) can be contained in a separate file referenced, for example by a URL. Moreover, the ability of a Web browser to retrieve and process data from both local and non-local sources is an inherent feature of such browsers. Indeed, one of the first applications of HTML/Web browsers was the rendering in a document displayed in a single window of text and images, where the image data was contained in files separate from those containing the text.

[5] *Raggett I*, for example, also disclosed these same features as the Mosaic Version 2.4 browser. In particular, it disclosed an "information browser[]," *i.e.*, a "computer program product," that can be used to display documents in HTML+ format (a successor to the HTML format then widely in use). *Raggett* I at pages 1-2. It also explained the HTML+ is "for use within the World Wide Web" and that the World Wide Web "is a wide area client-server architecture for retrieving hypermedia documents across the Internet. *Id.* at page 1. It also described "pars[ing] hypermedia documents" (*see id.* at page 3), and "utiliz[ing] [a] browser to display" a hypermedia document (*see id.* at page 1). In general, all the basic browser functions of Mosaic Version 2.4 are inherent in *Raggett I* since such functions are required to display HTML-type hypermedia documents.

An element by element comparison of claim 6-8 to the acknowledged and newly cited prior art is provided below in Table I. It shows that each and every element of each of claims 6-8 is present in the Mosaic version 2.4 browser in combination with *Raggett I* and *Raggett II*, and in *Raggett I* and *II* themselves (*i.e.*, even without relying on Mosaic version 2.4). Claims 1-3 are comparable to claims 6-8, respectively, and each and every element of those claims are also present in the acknowledged and newly cited prior art for the same reasons provided in Table I.

Table I		
	Acknowledged Prior Art	**Newly Cited Art**
6. A computer program product for use in a system having at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment, the computer program product comprising: *a computer usable medium having computer readable program code physically embodied therein, said computer program product further comprising: computer readable program code for causing said client workstation to execute a browser application*	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (describing the Internet, and the use and function of browser programs, and noting that Mosaic is "an example of a browser program").	*Raggett I* at page 1 (explaining that "HTML+ is a simple SGML based format for wide-area hypertext documents, for use within the World Wide Web," that "[t]he World Wide Web is a wide area client-server architecture for retrieving hypermedia documents across the Internet," and that "[i]nformation browsers can display information … in the HTML+ format")
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;	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (same).	*Raggett I* at page 3 (discussing "Parsing HTML+ Documents").
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,	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (same).	*Raggett I* at page 1 (explaining that "HTML+ is a simple SGML based format for wide-area hypertext documents, for use within the World Wide Web," and that "[t]he World Wide Web is a wide area client-server architecture for retrieving hypermedia documents across the Internet").
wherein the portion of said first hypermedia document is displayed within a first browser-controlled	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (same).	*Raggett I* at page 1 (explaining that "[i]nformation browsers can

Table I

	Acknowledged Prior Art	Newly Cited Art
window on said client workstation,		display information … in the HTML+ format")
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,	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (same).	*See Raggett II* at pages 1-2 (providing an example of an EMBED tag (*i.e.*, an embedded text format) and stating that the foreign (*i.e.*, embedded) data can be put "in a separate file referenced by a URL"). *See also Raggett I* at p. 12 (explaining that the image for the "fig" tag, which is used to display, e.g., images and graphics, can be "defined by a link to an external document.")
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	Mosaic, see '906 patent at column 3, lines 5 to 6 (the Mosaic program "retrieves the corresponding full image … and displays it by using external software").	*Raggett I* at page 6 (explaining that the "type attribute" to the EMBED tag "specifies a registered MIME content type and is used by the browser to identify the appropriate shared library or external filter to use to render the embedded data, e.g., by returning a pixmap."); *Raggett II* at page 1 (explaining that "[t]he browser identifies the format of the embedded data from the "type" attribute [to the EMBED tag], specified as a MIME content type;" and further explaining that the type information is used to identify, e.g., a "separate program[]" or "dynamically linked library" for rendering the data).
and wherein said embed text format is parsed by said browser to automatically invoke said executable application on said client workstation	Mosaic, see '906 patent at column 1, line 19 to column 3, line 51 (noting that Mosaic is "an example of a browser program" and, as such, parses HTML documents accessed).	*Raggett I* at pages 3 and 6 (discussing "Parsing HTML+ Documents" generally, and "the EMBED tag" specifically, as part of the automatic processing of an HTML+ document by a Web

Table I		
	Acknowledged Prior Art	**Newly Cited Art**
		browser); *Raggett II* at page 1 (explaining that "[t]he browser identifies the format of the embedded data from the "type" attribute, specified as a MIME content type."). As explained above, *Raggett I* and *II* describe using the "type" attribute to the EMBED tag to identify an external application program or shared library capable of rendering the embedded data. The browser then invokes the identified application or shared library, which in turn returns, for example, "a pixmap." *Raggett I*, p. 6; *Raggett II, p. 1.*
in order to display said object	Mosaic, see '906 patent at column 3, lines 5 to 6 (the Mosaic program "retrieves the corresponding full image … and displays it by using external software").	The purpose of the EMBED tag described in *Raggett I* and *Raggett II* is to display in-line information rendered by an external application program or shared library. *See, e.g., Raggett I* at page 6 (explaining that the "appropriate shared library or external filter [*i.e.*, application program]" is used to "render the embedded data, e.g. by returning a bitmap."). *See also, e.g. Raggett II* at page 1 (explaining that "[b]rowsers can then be upgraded to display new formats without changing their code at all").
and enable interactive processing of said object		*Raggett I* at page 6, line 47 ("Sophistocated [sic] browsers can link to external editors for updating and revising embedded data.").
within a display area created at said first location within the portion of said first distributed hypermedia		*Raggett II* at page 1 (explaining in response to emails regarding embedding

Table I

	Acknowledged Prior Art	Newly Cited Art
document being displayed in said first browser-controlled window.		equations and encapsulated Postscript within documents to be displayed on the Web (*e.g.*, HTML documents) that "both of these will be possible with the HTML+ DTD, by using the capability to embed foreign formats inline in the HTML+ source …") (emphasis added). *See also Raggett I* at pages 6 and 12 (describing the EMBED tag, which is used to embed data having an external format within a Web page); *see also*, *id.*, at page 34 (explaining, in a section entitled "Notes for Implementers," that "[i]t is generally better to avoid displaying the retrieved document in a new window, unless explicitly requested by the user.").
7. The computer program product of claim 6, wherein said executable application is a controllable application and further compising: *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.*		*See Raggett I* at page 6 (describing inter-process communication between the browser and an external editor: "[s]ophistocated [sic] browsers can link to external editors for creating or revising embedded data"). Also *Raggett I* and *II* describe having the browser use shared libraries, such as DLLs, for rendering data in external formats. *Raggett I* at page 6, *Raggett II* at page 1. Such shared libraries would necessarily be controlled through inter-process communications with the browser that invoked them since shared libraries are not independently executable

Table I

	Acknowledged Prior Art	Newly Cited Art
		(that is, they cannot execute unless they are invoked by another program, such as the browser here).
8. 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 program has been launched.		Again *Raggett I* at page 6 explains that "[s]ophistocated [sic] browsers can link to external editors for creating or revising embedded data". Since the browser displays information rendered by the external program, here the editor, the operation of such an external editor plainly requires continuing communication between the browser and the editor. Otherwise a user would not see displayed the changes being made to the embedded data during the process of revising that data.

Raggett I and *II* Anticipate Claims 1-3 and 6-8

As shown in Table I above, *Raggett I* and *II* collectively disclose each and every element of claims 1-3 and 6-8. In addition, *Raggett I* and *II* comprise a single prior art publication because both were posted on or incorporated by reference in the same Website at the same time more than a year before the filing date of the '906 patent. Specifically, all messages sent to the www-talk email list, including *Raggett II* and a message containing a link to *Raggett I* (*see* Exhibit C hereto), were also posted on the http://eies2.njit.edu:80/wmail.html Website (*see* Exhibit D hereto). Thus, as of July 23, 1993, both *Raggett I* (which is dated July 23, 1993) and *Raggett II* (which is dated June 14, 1993) were effectively published on a single Website. Since *Raggett I* and *II* comprise a single publication and disclose each and every element of claims 1-3 and 6-8, they thus anticipate those claims.

Claims 1-3 and 6-8 are also Obvious Over the Mosaic Version 2.4 Browser in View of *Raggett I* and *Raggett II*

In addition to being anticipated by *Raggett I* and *II*, as set forth above, claims 1-3 and 6-8 are also obvious over the acknowledged Mosaic browser in view of *Raggett I* and *II*. *Raggett I* and *II* specifically teach those of ordinary skill in the art to modify a prior art browser, such as the

Mosaic browser, to incorporate the allegedly new features of claims 1-3 and 6-8, rendering those claims obvious.

The Level of Ordinary Skill in the Art

The person of ordinary skill in the relevant art to the claimed invention is a software programmer with at least a bachelor's degree in Computer Science, and five years of programming experience in Internet, Web and browser technology, including specific experience with programming in HTML. However, even assuming a lower level of ordinary skill in the art, the claims of the '906 patent would still have been obvious, given that the enclosed prior art describe precisely what the '906 patent claims as its invention in precisely the same context.

The Prima Facie Obviousness of Claims 1-3 and 6-8

The printed publications provided herewith were not considered by the PTO during the original prosecution of the '906 patent. When considered in view of the acknowledged prior art (e.g., Mosaic Web browser, version 2.4) by a person of ordinary skill in the art, they render the claimed invention defined by claims 1-3 and 6-8 of the patent *prima facie* obvious.

As described above, the only difference between the claimed invention and the prior art Mosaic browser is that the Mosaic browser was not capable of processing an "embed text format" in a hypermedia document to "automatically invoke" an external application "to display" an external object within the browser window displaying the hypermedia document, as claimed. But *Raggett I* and *Raggett II* however specifically disclose implementing this functionality in a Web browser.

Raggett I and *II* thus provided specific motivation and guidance to a person of ordinary skill to modify the acknowledged prior art NCSA Mosaic version 2.4 browser (and other prior art browsers) to arrive at the claimed invention. Indeed, *Raggett I* (the HTML+ specification), which was publicly disseminated more than a year prior to the filing date of the '906 patent, required Web browsers to possess this functionality in order to be compliant with the proposed specification. As such, it is difficult to envision a document that could have provided greater motivation to modify a Web browser to provide the features called for therein. Furthermore, as acknowledged and admitted by the inventors of the '906 patent (*see, e.g.*, column 13, lines 51 to 59 and column 16, lines 51 to 53), the act of modifying the Mosaic prior art browser to implement the features called for by *Raggett I* and *II* was well within the abilities of a person having an ordinary level of skill in the relevant art (e.g., software programming). *Raggett I* and *Raggett II*, considered individually or in combination, in view of the acknowledged prior art, therefore establish a *prima facie* case of obviousness of claims 1-3 and 6-8.

Further comparison of the '906 patent specification to *Raggett I* and *Raggett II* leaves no doubt as to the accuracy of this conclusion. As described above, Table II (column 12, line 54, with descriptive text through column 13, line 31) of the '906 patent shows the preferred embodiment of an EMBED tag with HREF and TYPE attributes, which the browser uses to identify, locate and launch associated external applications. *Raggett I* and *II* use nearly identical language (*see, e.g. Raggett I*, page 6; *Raggett II*, last sentence) to describe the attributes of the EMBED tag. The enclosed publications thus disclose not only the same functionality but precisely the same means of implementing that functionality in Web browsers (i.e., the same "EMBED" tag is used to initiate the same browser behavior that provided the same result as the claimed subject matter of the '906 patent).

Moreover, the enclosed publications enable, as the '906 patent claims, Web browsers to provide the user with more functionality (e.g., through displaying and/or editing new data formats) without changing the browser code. Compare, '906 patent, column 11, lines 52 to 55, *Raggett I*, page 6, and *Raggett II*, page 1. Again, the enclosed publications were promulgated to the World Wide Web community more than a year before the filing of the '906 patent for the purpose of implementing this very same capability in prior art Web browsers.

Thus, the two printed publications provided herewith, taken in view of the admittedly prior art NCSA Mosaic version 2.4 browser, provided specific motivation and guidance to persons of ordinary skill to modify the NCSA Mosaic version 2.4 browser to arrive at the claimed invention. As such, these disclosures support a *prima facie* finding of obviousness of claims 1-3 and 6-8 of the '906 patent and render those claims obvious to a person of skill in the art.

Conclusion

The two *Raggett* publications provided herewith anticipate at least claims 1-3 and 6-8 of the '906 patent. In addition, the acknowledged prior art Mosaic version 2.4 browser, when considered together with the two *Raggett* publications, render at least claims 1-3 and 6-8 obvious. In view of the invalidity of these claims and the considerable adverse impact the '906 patent will have on the larger World Wide Web community, a Director initiated reexamination is appropriate.

Respectfully submitted,

Attorneys for Submitter
World Wide Web Consortium.

Date: October 23, 2003

Barry D. Rein (Reg. No. 22,411)
Kenneth L. Stein (Reg. No. 38,704)
PENNIE & EDMONDS LLP
1155 Avenue of the Americas
New York, New York 10036-2711
(212) 790-9090

831 PH Ex. 3

UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER SECRETARY OF COMMERCE FOR
INTELLECTUAL PROPERTY AND
DIRECTOR OF THE UNITED STATES PATENT
AND TRADEMARK OFFICE
Alexandria, Virginia 22313

CONTROL NUMBER	ORDER DATE	PATENT NUMBER	PATENTEE
90/006, 831	10-30-03	5,838,906	Doyle et al.

EXAMINER	
Caldwell, Andrew	
ART UNIT	PAPER NUMBER
2157	1

DATE MAILED: October 30,2003

Townsend and Townsend and Crew, LLP
Two Embarcadero Center
Eight Floor
San Francisco, CA 94111-3834

90006831 103003

DIRECTOR INITIATED ORDER FOR REEXAMINATION

Attachment(s): ☒PTO-892. □PTO-1449.
□ Other: ______________________

Response Time For Patent Owner's Statement:

TWO MONTHS from the date hereof. 37 CFR 1.530(b).

Notes: If the patent owner does not file a timely statement under 37 CFR 1.530(b), reexamination will proceed in accordance with 37 CFR 1.550(a).

An identification of the claims, the references relied on, and the rationale of the decision to order reexamination is attached.

REEXAMINATION ORDER:

Pursuant to 37 CFR 1.520, reexamination is ordered. Note the attached decision.

INTRODUCTION

Pursuant to 37 CFR 1.520, the Director of the United States Patent and Trademark Office has determined that the prior art discussed below raises a substantial new question of patentability as to claims 1-3 and 6-8 of U.S. Patent No. 5,838,906 (hereinafter, the '906 patent).

DECISION

Current Office guidelines provide that the policy of the Director is to order reexamination on his own initiative when it is apparent, after a review of the prosecution history, that there is a "comelling reason" to order reexamination, and at least one claim in a patent is prima facie unpatentable over prior patents and/or printed publications. Circumstances that can meet the "compelling reason" requirement include: (1) an examining practice, policy or procedure was not followed before the grant of a patent which resulted in a failure to consider patents and/or printed publications which *prima facie* make any claim(s) unpatentable, **and/or** (2) a significant concern about the patentability of the claimed subject matter has been expressed by a substantial segment of the industry, and/or there is substantial media publicity (e.g., the Internet or the news services) adverse to the patent alleging conspicuous unpatentability of the claims. In the case of the '906 patent, a substantial outcry from a widespread segment of the affected industry has essentially raised a question of patentability with respect to the '906 patent claims. This creates an extraordinary situation for which a Director ordered reexamination is an appropriate remedy.

I. THE '906 PATENT

The '906 patent is drawn to a method for running embedded program objects in a computer network environment. The method includes the steps of providing at least one client workstation and one network server coupled to the network environment where the network environment is a distributed hypermedia environment; displaying, on the client workstation, a portion of a hypermedia document received over the network from the server, where the hypermedia document includes an embedded controllable application; and interactively controlling the embedded controllable application from the client workstation via communication sent over the distributed hypermedia environment.

The invention of the '906 patent allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way. The invention not only allows the user to use a hypermedia format to locate and retrieve program objects, but also allows the user to interact with an application program located at a remote computer. Interprocess communication between the hypermedia browser and the embedded application program is ongoing after the program object has been launched.

Claims 1-3 and 6-8 of the '906 Patent:

Claims 1-3 of the '906 patent are drawn to a method for running an application program in a computer network environment. The method comprising, in brief, the steps of:

> providing at least one client workstation and one network server coupled to the network environment, wherein the network environment is a distributed hypermedia environment;
>
> executing, at the client workstation, a browser application, that parses a first distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processing by the text formats;
>
> utilizing the browser to display, on the client workstation, at least a portion of a first hypermedia document received over the network from the server, wherein the portion of the first hypermedia document is displayed within a first browser-controlled window on the client workstation, wherein the first distributed hypermedia document includes an embed text format, located at a first location in the first distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document, wherein the object has type associated with it utilized by the browser to identify and locate an executable application external to the first distributed hypermedia document, and wherein the embed text format is parsed by the browser to automatically invoke the executable application to execute on the client workstation in order to display the object and enable interactive processing of the object within a display area created at the first location within the portion of the first distributed hypermedia document being displayed in the first browser-controlled window.

Claims 6-8 of the '906 patent are drawn to a computer program product for use in a system having at least one client workstation and one network server coupled to the network environment, wherein the network environment is a distributed hypermedia environment. The computer program product comprises, in brief:

> a computer usable medium having computer readable program code physically embodied therein, the computer program product further comprises:
>
> > computer readable program code for causing the client workstation to execute a browser application to parse a first distributed hypermedia document to identify text formats included in the distributed hypermedia document and to respond to predetermined text formats to initiate processes specified by the text formats;

> computer readable program code for causing the client workstation to utilize the browser to display, on the client workstation, at least a portion of a first hypermedia document received over the network from the server, wherein the portion of the first hypermedia document is displayed within a first browser-controlled window on the client workstation, wherein the first distributed hypermedia document includes an embed text format, located at a first location in the first distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document, wherein the object has type information associated with it utilized by the browser to identify and locate an executable application external to the first distributed hypermedia document, and wherein the embed text format is parsed by the browser to automatically invoke the executable application to execute on the client workstation in order to display the object and enable interactive processing of the object within a display area created at the first location within the portion of the first distributed hypermedia document being displayed in the first browser-controlled window.

II. PRIOR ART

A. Berners-Lee, T., et al., *Hypertext Markup Language (HTML)*, Internet Draft, IETF, pages 1-40, (June 1993).

B. Raggett, D., *HTML+ (Hypertext Markup Language)*, (July 23, 1993). Hereinafter referred to as "Raggett I."

C. Raggett, D., Posting of Dave Raggett, dsr@hplb.hpl.hp.com towww-talk@nxoc01.cern.ch (WWW-TALK public mailing list), (Posted June 14, 1993). Hereinafter referred to as "Raggett II."

III. THE '906 PATENT AND THE PRIOR ART

Applying the teachings of Berners-Lee, Raggett I, and Raggett II to the claims of the '906 patent:

A substantial new question of patentability exists with respect to claims 1-3 and 6-8 of the '906 patent in view of prior art acknowledged by the patentee in the '906 patent and the newly cited teachings of Berners-Lee, Raggett I, and Raggett II."

Regarding claim 1 of the '906 patent, patentee acknowledges that the admitted prior art teaches a portion of the claimed invention of claim 1 of the '906 patent, namely a method comprising:

> "providing at least one client workstation" (**See USP '906: Figure 2, element 130; Col. 4, Lines 32-40** which indicate that "small computer" 130 can be a client) "and one network server" (See **USP '906: Figure 2, element 132**) "coupled to a network environment" (See **USP '906: Figure 2, element 100 Internet**), "wherein the network environment is a distributed hypermedia environment" (See **USP '906: Col. 5 lines 24-25**);
>
> "executing, at the client workstation, a browser application" (See **USP '906: Col. 3 lines 9-13**), "that parses a first distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by the text formats" (See **USP '906: Col. 1, lines 1-Col. 3, line 51, with particular emphasis on Col. 2, line 63-Col. 3, line 25 showing a browser executing on client that parses and then displays a hypermedia document; where the user clicks on a link/image icon causing the browser to invoke a viewer application displaying the image in a separate window**); and
>
> "utilizing the browser to display, on the client workstation, at least a portion of a first hypermedia document received over the network from the server, wherein the portion of the first hypermedia document is displayed within a first browser-controlled window on the client workstation." (See **USP '906: Figure 1, element 10 as hypermedia document displayed on client; Col. 2 lines 28-36**).

While the admitted prior art describes a method in which a hypermedia page (See **USP '906: Figure 1, element 10**) is displayed in a browser (See **USP '906: Col. 1, lines 1-Col. 3, line 51, particularly Col. 2, line 63-Col. 3, line 25**), the prior art does not teach, as in claim 1 of the '906 patent, the particular steps used by the browser in order to process and display the hypermedia page. To summarize, the prior art does not teach a method wherein the browser application parses a first distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by the text formats.

Nevertheless, Berners-Lee teaches that HTML browsers parse HTML. (See **Berners-Lee at p. 2 as printed – paragraph starting "Implentations of ..."**) The parsing is used to identify characters interpreted as markup elements, such as the various tags (**see Berners-Lee at page 5**) in the structured text example, and to associate text with various tags. These tags correspond to the claimed "text formats." Berners-Lee also teaches that the browser processes the HTML by rendering it into a displayable form. (**See Berners-Lee at p. 3, definition of rendering**). Berners-Lee also discusses how specific markup elements are to

be rendered. (**See for example, Berners-Lee at p. 14, typical rendering of address tag; p. 15 typical rendering of block quote**). Berners-Lee therefore is argued to teach a method in which a browser application parses a first distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by the text formats.
It is argued that it would have been readily apparent to a skilled artisan to combine (1) the teachings of Berners-Lee regarding the processing of HTML documents performed by a browser, with (2) the HTML browser of the patent admitted prior art in light of the statement made by the prior art that its hypermedia system is designed to handle hypermedia documents according the HTML markup standard. (See **USP '906: Col. 5, lines 28-31**).

Regarding the processing of the claimed "text formats,"patentee acknowledges that the prior art teaches a method wherein a browser invokes an external viewer program to process various file formats not handled directly by the browser. (See **USP '906: Col. 3, lines 13-20**). Specifically, the prior art describes an example wherein the file format not handled by the browser is an image file in ".TIF" or ".GIF" format and the browser invokes an image viewer program to display the full image in a separate window. (See **USP '906: Col. 3 lines 13-20**). While the prior art teaches that certain tags may cause the browser to invoke external applications to process particular file formats, these applications do not display their data in the browser window. Therefore, patentee notes that the prior art does not teach the portion of the method of claim 1 of the '906 patent 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;
>
> 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
>
> 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 interactive processing of 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."

However, Raggett I teaches various extensions to the HTML specification including an EMBED tag that provides a simple form of object level embedding. (**See Raggett I: p. 6 "Embedded data in an external format" and p. 26 embedded.**) For example, Ragget I teaches an HTML document including an EMBED tag that identifies embedded data in a foreign format. **(See Raggett I: p. 6 <embed ...> and </embed> tags.)** This embedded data is an object that cannot be directly processed by the browser. The foreign format data, or object, is embedded in the HTML document by placing it between the <embed ...> and </embed> tags. **(See Raggett I: p. 6 "2 pi int sin (omega t)dt" as an example of embedded foreign data.)** Raggett I describes the example of an embedded equation, where the browser calls a program for rendering an equation by providing ascii character information to an external program and receives a pixmap image of the equation from the external program that is then displayed in the browser window. **(See Raggett I: p. 6, particularly the last ten lines.)** Thus it is argued that Raggett I teaches "a first distributed hypermedia document that includes an embed text format, located at a first location in said first distributed hypermedia document," that is used to identify embedded foreign data. Raggett I also teaches that the embed tags include a type attribute specifying a registered MIME content type that is used by the browser to identify the appropriate external filter to use to render the embedded foreign data. **(See Raggett I: p. 6 type="application/eqn".)** Raggett I thus teaches a method wherein "the 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 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." Although Raggett I describes an example where the browser calls a program for rendering an equation in ASCII character format into a pixmap image of the equation, Raggett I does also recognize that more sophisticated browsers can link to external editors for creating or revising embedded data. These external editors that create or revise the embedded data would work in the same way as the simple example of providing equation support. **(See Raggett I: p. 6.)** However, the ability to create and revise the embedded data allows the user to interactively process the data within the browser window. Thus it is asserted that Raggett I teaches a method which "enables interactive processing of 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."

It would have readily apparent to a skilled artisan to combine (1) Raggett I's teachings regarding extensions to the HTML standard (i.e., the proposed HTML+ Specification) allowing the embedding of data in foreign formats within web pages with (2) the method as taught by patentee's admitted prior art. This combination would have been apparent based on Raggett I's acknowledgment that this particular extension to HTML is advantageous and it represents a "substantial improvement." **(See Raggett I: p. 1 2nd paragraph of abstract)**.

The combination of patentee's admited prior art in view of Berners-Lee and Raggett I does not explicitly teach a system wherein "the embed text format specifies the location of at least a portion of an object external to the first distributed hypermedia document." Raggett I describes a method in which the object itself is embedded in the HTML document. (See

Raggett I: p. 6 embedded data in an external format – see example on the last two lines of the page where the object, the text representation of the equation, is within the embed tags).

Raggett II, though, teaches putting the foreign data in a separate file and then referencing that file by a URL in the HTML+ embed tag. **(See Raggett II: last line.)** It is thus argued that Raggett II describes a system wherein "the embed text format specifies the location of at least a portion of an object external to the first distributed hypermedia document."

It would have been readily apparent to a skilled artisan to modify the method discussed above, combining the teachings of patentee's admitted prior art in view of Berners-Lee and Raggett I, by further substituting a URL which references a separate file containing foreign data for the embedded foreign data within the hypermedia document of the combination. Such a further modification would have been apparent based on Raggett II's explicit suggestion to make such a substitution. (See Raggett II: last line.)

Regarding claim 2 of the '906 patent, Raggett II teaches a method wherein "said executable application is a controllable application" and the method further comprises the step of "interactively controlling said controllable application on said client workstation via inter-process communications between said browser and said controllable application." (It is noted that Raggett II functions could be implemented as separate programs driven via pipes and stdin/stdout or as dynamically linked library modules.)

Regarding claim 3 of the '906 patent, the combination of patentee's admitted prior art in view of Berners-Lee, Raggett I, and Raggett II is argued to teach the invention substantially as claimed. (See the rejection of claim 2, above.) However, the combination of the patentee's prior art in view of Berners-Lee, Raggett I, and Raggett II does not explicitly teach the additional limitation of claim 3. Nevertheless, Raggett I does teach that sophisticated browsers can link to external editors for creating or revising embedded data. **(See Raggett I: p. 6)**. The fact that the creating and revising is performed by an external *editor* would suggest to a skilled artisan that the creating and revising is an interactive process controlled by the browser user. The use of an editor to create or revise an object suggests a continued interaction between the browser and the external editor during the editing process. A skilled artisan would therefore reasonably infer that the combination of the admitted prior art in view of Berners-Lee, Raggett I, and Raggett II teaches a method wherein "communications to interactively control said controllable application continue to be exchanged between the controllable application" (i.e, the external editor) and the browser even after the controllable application program has been launched.

Regarding claims 6-8 of the '906 patent, such claims are *computer program product claims* which correspond to method claims 1-3, respectively. Since they do not teach or define above the information in the corresponding method claims, the discussion and application, *supra*, of the admitted prior art in combination with the newly cited references of Berners-Lee, Raggett I, and Raggett II to method claims 1-3 is applied to claims 6-8, respectively.

CONCLUSION

In view of the admitted prior art of the '906 patent and the teachings of Berners-Lee, Raggett I, and Raggett II, a substantial new question of patentability is raised as claims 1-3 and 6-8 of U.S. Patent No. 5,838,906. Reexamination of all of the claims of U.S. Patent No. 5,838,906 is hereby ordered under 37 CFR 1.520.

Stephen G. Kunin

Stephen G. Kunin
Deputy Commissioner for Patent
Examination Policy

5838906.cmr

831 PH Ex. 4

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/006,831	10/30/2003	5838906		9718

30080 7590 02/26/2004

LAW OFFICE OF CHARLES E. KRUEGER
P.O. BOX 5607
WALNUT CREEK, CA 94596-1607

EXAMINER	
CALDWELL, A.	
ART UNIT	PAPER NUMBER
2157	9

DATE MAILED: 02/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

90006831 . 022604

PTO-90C (Rev 10/03)

Office Action in Ex Parte Reexamination	**Control No.** 90/006,831	**Patent Under Reexamination** 5838906	
	Examiner Andrew Caldwell	**Art Unit** 2151	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

a☒ Responsive to the communication(s) filed on *January 5 & 30, 2004* . b☐ This action is made FINAL.
c☒ A statement under 37 CFR 1.530 has not been received from the patent owner.

A shortened statutory period for response to this action is set to expire *two* month(s) from the mailing date of this letter.
Failure to respond within the period for response will result in termination of the proceeding and issuance of an *ex parte* reexamination certificate in accordance with this action. 37 CFR 1.550(d). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).**
If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

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

1. ☒ Notice of References Cited by Examiner, PTO-892.
2. ☐ Information Disclosure Statement, PTO-1449.
3. ☐ Interview Summary, PTO-474.
4. ☐ _____.

Part II SUMMARY OF ACTION

1a. ☒ Claims *1-10* are subject to reexamination.

1b. ☐ Claims _____ are not subject to reexamination.

2. ☐ Claims _____ have been canceled in the present reexamination proceeding.

3. ☐ Claims _____ are patentable and/or confirmed.

4. ☒ Claims *1-10* are rejected.

5. ☐ Claims _____ are objected to.

6. ☒ The drawings, filed on *30 October 2003* are acceptable.

7. ☐ The proposed drawing correction, filed on _____ has been (7a)☐ approved (7b)☐ disapproved.

8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some* c)☐ None of the certified copies have

1☐ been received.

2☐ not been received.

3☐ been filed in Application No. _____ .

4☐ been filed in reexamination Control No. _____.

5☐ been received by the International Bureau in PCT application No. _____.

* See the attached detailed Office action for a list of the certified copies not received.

9. ☐ Since the proceeding appears to be in condition for issuance of an *ex parte* reexamination certificate except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte* Quayle, 1935 C.D. 11, 453 O.G. 213.

10. ☐ Other: _____

cc: Requester (if third party requester)

Information Disclosure Statement

The Earlier Viola Source Code (dated May 12, 1993) and the Later Viola Source Code (dated May 27, 1993) that was provided on a CD accompanying the information disclosure statement filed on January 5, 2004 (paper no. 6) has not been considered. Claims in an ex parte reexamination proceeding are examined on the basis of patents or printed publications. 37 CFR 1.552(a). The Applicants have neither admitted that the Viola source code is prior art nor provided any evidence that the Viola source code is a publication. The Applicants have merely pointed to a ruling of a U.S. District Court that raises questions as to whether the Viola source code was publicly available. Accordingly, the information has been placed in the application file but has not been considered as to the merits.

Claim Rejections - 35 USC § 103

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

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

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary.

The Prior Art as Applied to Claims 1-3 and 6-8:

Berners-Lee, T., et al., Hypertext Markup Language (HTML), Internet Draft, IETF, pages 1-40, (June 1993).

Raggett, D., HTML+ (Hypertext Markup Language), (July 23, 1993). Hereinafter referred to as "Raggett L"

Raggett, D., Posting of Dave Raggett, dsr@hplb.hpl.hp.com towww-talk@nxocOl.cern.ch (W-WW-TALK public mailing list), (Posted June 14, 1993). Hereinafter referred to as "Raggett II."

Claims 1-3 and 6-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over the admitted prior art in the `906 patent and the newly cited teachings of Berners-Lee, Raggett I, and Raggett II.

Regarding claim 1 of the `906 patent, the admitted prior art teaches a portion of the claimed invention of claim 1 of the `906 patent, namely a method comprising:

"providing at least one client workstation" (**See USP `906: Figure 2, element 130; Col. 4, Lines 32-40 which indicate that "small computer" 130 can be a client**) "and one network server" (**See USP `906: Figure 2, element 132**) "coupled to a network environment" (**See USP `906: Figure 2, element 100 Internet**), "wherein the network environment is a distributed hypermedia environment" (**See USP `906: Col. 5 lines 24-25**);

"executing, at the client workstation, a browser application" (**See USP `906: Col. 3 lines 9-13**), "that parses a first distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by the text formats" (**See USP `906: Col. 1, lines 1-Col. 3, line 51, with particular emphasis on Col. 2, line 63-Col. 3, line 25 showing a browser executing on client that parses and then displays a hypermedia document; where the user clicks on a link/image icon causing the browser to invoke a viewer application displaying the image in a separate window**); and

"utilizing the browser to display, on the client workstation, at least a portion of a first hypermedia document received over the network from the server, wherein the portion of the first hypermedia document is displayed within a first browser-controlled window on the client workstation." (**See USP `906: Figure 1, element 10 as hypermedia document displayed on client; Col. 2 lines 28-36**).

While the admitted prior art describes a method in which a hypermedia page (**See USP `906: Figure 1, element 10**) is displayed in a browser (**See USP `906: Col. 1, lines 1-Col. 3, line 51, particularly Col. 2, line 63-Col. 3, line 25**), the admitted prior art does not teach, as in claim 1 of the `906 patent, the particular steps used by the browser in order to process and display the hypermedia page. To summarize, the admitted prior art does not teach a method wherein the browser application parses a first distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by the text formats.

Nevertheless, Bemers-Lee teaches that HTML browsers parse HTML. (**See Berners-Lee at p. 2 as printed - paragraph starting; "Implentations of ..."**) The parsing is used to identify characters interpreted as markup elements, such as the various tags (**see Berners-Lee at page 5**) in the structured text example, and to associate text with various tags. These tags correspond to the claimed "text formats." Bemers-Lee also teaches that the browser processes the HTML by rendering it into a displayable form. (**See Berners-Lee at p. 3, definition of rendering**). Berners-Lee also discusses how specific markup elements are to be rendered. (**See for example, Berners-Lee at p. 14, typical rendering of address tag; p.15 typical rendering of block quote**). Berners-Lee therefore teaches a method in which a browser application parses a first distributed hypermedia document to identify text formats included in the distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by the text formats.

It would have been obvious to a skilled artisan to combine (1) the teachings of Berners-Lee regarding the processing of HTML documents performed by a browser, with (2) the HTML browser of the admitted prior art in light of the statement made by the admitted prior art that its hypermedia system is designed to handle hypermedia documents according the HTML markup standard. (**See USP `906: Col. 5, lines 28-31**).

Regarding the processing of the claimed "'text formats," patentee acknowledges that the prior art teaches a method wherein a browser invokes an external viewer program to process various file formats not handled directly by the browser. (**See USP `906: Col. 3, lines 13-20**). Specifically, the prior art describes an example wherein the file format not handled by the browser is an image file in ".TIF" or ".GIF" format and the browser invokes an image viewer program to display the full image in a separate window. (**See USP `906: Col. 3 lines 13-20**). While the prior art teaches that certain tags may cause the browser to invoke external applications to process particular file formats, these applications do not display their data in the browser window. Therefore, the admitted prior art does not teach the portion of the method of claim 1 of the '906 patent 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;
>
> 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
>
> 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 interactive processing of 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."

However, Raggett I teaches various extensions to the HTML specification including an EMBED tag that provides a simple form of object level embedding. (**See Raggett I: p. 6 "Embedded data in an external format" and p. 26 embedded.**) For example, Ragget I teaches an HTML document including an EMBED tag that identifies embedded data in a foreign format. (**See Raggett 1: p. 6 <embed ...> and <embed> tags.**) This embedded data is an object that cannot be directly processed by the browser. The foreign format data, or object, is embedded in the HTML document by placing it between the <embed ...> and </embed> tags. (**See Raggett 1: p. 6 "2 pi int sin (omega t)dt" as an example of embedded foreign data.**) Raggett I describes the example of an embedded equation, where the browser calls a program for rendering an equation by providing ascii character information to an external program and receives a pixmap image of the equation from the external program that is then displayed in the browser window. (**See Raggett 1: p. 6, particularly the last ten lines.**) Raggett I therefore teaches "a first distributed hypermedia document that includes an embed text: format, located at a first location in said first distributed hypermedia document," that is used to identify embedded foreign data. Raggett I also teaches that the embed tags include a type attribute specifying a registered MIME content type that is used by the browser to identify the appropriate external filter to use to render the embedded foreign data. (**See Raggett 1: p. 6 type="application/eqn".**) Raggett I thus teaches a method wherein "the 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 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." Although Raggett I describes an example where the browser calls a program for rendering an equation in ASCII character format into a pixmap image of the equation, Raggett I does also recognize that more sophisticated browsers can link to external editors for creating or revising embedded data. These external editors that create or revise the embedded data would work in the same way as the simple example of providing equation support. (**See Raggett 1: p. 6.**) However, the ability to create and revise the embedded data allows the user to interactively process the data within the browser window. Raggett I therefore teaches a method which "enables interactive processing of 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."

It would have been obvious to a skilled artisan to combine (1) Raggett I's teachings regarding extensions to the HTML standard (i.e., the proposed HTML+ Specification) allowing the embedding of data in foreign formats within web pages with (2) the method as taught by patentee's admitted prior art. This combination would have been obvious based on Raggett I's acknowledgment that this particular extension to HTML is advantageous and it represents a "substantial improvement." (**See Raggett 1: p. 1 2nd paragraph of abstract**).

The combination of patentee's admitted prior art in view of Berners-Lee and Raggett I does not explicitly teach a system wherein "the embed text format specifies the location of at least a portion of an object external to the first distributed hypermedia document." Raggett I describes a method in which the object itself is embedded in the HTML document. (**See Raggett I: p. 6 embedded data in an external format - see example on the last two lines of the page where the object, the text representation of the equation, is within the embed tags**).

Raggett II, though, teaches putting the foreign data in a separate file and then referencing that file by a URL in the HTML+ embed tag. (**See Raggett II: last line.**) It is thus argued that Raggett II describes a system wherein "the embed text format specifies the location of at least a portion of an object external to the first distributed hypermedia document."

It would have been readily apparent to a skilled artisan to modify the method discussed above, combining the teachings of the admitted prior art in view of Berners-Lee and Raggett I, by further substituting a URL which references a separate file containing foreign data for the embedded foreign data within the hypermedia document of the combination. Such a further modification would have been apparent based on Raggett II's explicit suggestion to make such a substitution. (**See Raggett II: last line.**)

Regarding claim 2 of the `906 patent, Raggett II teaches a method wherein "said executable application is a controllable application" and the method further comprises the step of "interactively controlling said controllable application on said client workstation via interprocess communications between said browser and said controllable application." (It is noted that Raggett II functions could be implemented as separate programs driven via pipes and stdin/stdout or as dynamically linked library modules.)

Regarding claim 3 of the `906 patent, the combination of patentee's admitted prior art in view of Berners-Lee, Raggett I, and Raggett II teaches the invention substantially as claimed. (See the rejection of claim 2, above.) However, the combination of the patentee's prior art in view of Berners-Lee, Raggett I, and Raggett II does not explicitly teach the additional limitation of claim 3. Nevertheless, Raggett I does teach that sophisticated browsers can link to external editors for creating or revising embedded data. (**See Raggett I: p. 6.**) The fact that the creating and revising is performed by an external editor would suggest to a skilled artisan that the creating and revising is an interactive process controlled by the browser user. The use of an editor to create or revise an object suggests a continued interaction between the browser and the external editor during the editing process. A skilled artisan would therefore reasonably infer that the combination of the admitted prior art in view of Berners-Lee, Raggett I, and Raggett II teaches a method wherein "communications to interactively control said controllable application continue to be exchanged between the controllable application" (i.e., the external editor) and the browser even after the controllable application program has been launched.

Regarding claims 6-8 of the `906 patent, such claims are computer program product claims which correspond to method claims 1-3, respectively. Since they do not teach or define above the information in the corresponding method claims, the discussion and application, supra, of the admitted prior art in combination with the newly cited references of BernersLee, Raggett I, and Raggett II to method claims 1-3 is applied to claims 6-8, respectively.

The Prior Art as Applied to Claims 4-5 and 9-10:

Reichard, K., et al., X11R6: the Rumored Changes (Release 6 of the X Window System), Unix Review, vol. 11, no. 5, p. 101(5), pp. 1-4 as printed, May 1993.

Cox, B., Object Oriented Programming: An Evolutionary Approach, Addison-Wesley, pp. 1-12, 1987.

Claims 4-5 and 9-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of the admitted prior art in the `906 patent in view of Berners-Lee, Raggett I, and Raggett II, as applied to claims 3 and 8 above, and further in view of Reichard and Cox.

Regarding claim 4, the combination of the admitted prior art in view of Berners-Lee, Raggett I, and Raggett II teaches the invention substantially as claimed. (See the rejection of claim 3, above.) The combination also describes a method in which the browser is implemented to run on an X Windows platform (**See USP `906: Col. 8 lines 10-16**). The combination teaches that the controllable applications (i.e., external editors) for creating and revising embedded data executes on the same machine as the browser (**See USP '906: Col. 3 lines 15-16; Col. 6 lines 34-39.**) Since the examples of external editors all produce output directed to the computer's graphical user interface (**See Raggett I: p. 6 listing the rendering of mathematical equations and simple drawings using TeX and eqn as examples**), it would have been obvious to a skilled artisan that the combination's controllable application would be implemented to run on an X Windows platform as well.

However, the combination of the admitted prior art in view of Berners-Lee, Raggett I, and Raggett II does not explicitly teach the additional limitations of claim 4.

Nevertheless, Reichard teaches an extension to the X Windows system, the Fresco toolkit, that allows the linking and embedding of object components, where the objects can be distributed between processes on a single machine or across a network on many machines. (**See Reichard: p. 2 first two paragraphs in Objects Everywhere section.**)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the browser and controllable application (i.e, external editor) of the combination of the admitted prior art in view of Berners-Lee, Raggett I, and Raggett II's using Reichard's distributed object toolkit because of Cox's teaching that applying object oriented techniques to software makes the software more tolerant to change (**See Cox: p. 8 last three lines.**)

Once the browser and the controllable application both support distributed objects, it would have been obvious to move the controllable application (i.e., external editor) to a remote machine across the network based on Reichard's explicit suggestion. (**See Reichard: p. 2 first paragraph in Objects Everywhere section.**) The combination of the admitted prior art in view of Berners-Lee, Raggett I, Raggett II, Reichard, and Cox therefore teaches a method wherein additional instructions for controlling said controllable application reside on a network server (i.e., a remote machine across the network).

As to the remaining steps introduced in the claim, these steps all flow logically from the movement of the controllable application from the client workstation to a network server. The step of issuing, from the client workstation, one or more commands to the network server flows logically from the fact that user editing commands entered at the browser computer must be transmitted from the client workstation to the controllable application executing on the remote machine. The step of executing, on the network server, one or more instructions in response to the commands is taught by the controllable application (i.e, the external editor) executing on the remote machine. The step of sending information from said network server to said client workstation in response to said executed instructions is taught by the controllable application returning a result of the editing process to the client workstation. The step of processing said information at the client workstation to interactively control said controllable application is taught by the client workstation rendering the result of the edit in the browser window, thus allowing the user to see the results of the editing operation so the user can decide what editing operation to perform next.

Regarding claim 5, the combination of the admitted prior art in view of Berners-Lee, Raggett I, Raggett II, Reichard, and Cox teaches that the results returned by the controllable application residing on the network server are displayed in the browser window. The instructions performing this function are additional instructions for controlling said controllable application reside on said client workstation.

Regarding claims 9-10 of the `906 patent, such claims are computer program product claims which correspond to method claims 4-5, respectively. Since they do not teach or define above the information in the corresponding method claims, the discussion and application, supra, of the admitted prior art in combination with the newly cited references of Berners-Lee, Raggett I, Raggett II, Reichard, and Cox to method claims 4-5 is applied to claims 9-10, respectively.

Conclusion

The patent owner is reminded of the continuing responsibility under 37 CFR 1.565(a), to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving Patent No. 5,838,906 throughout the course of this reexamination proceeding. See MPEP §§ 2207, 2282 and 2286.

In order to ensure full consideration of any amendments, affidavits or declarations, or other documents as evidence of patentability, such documents **must** be submitted in response to this Office action. Submissions after the next Office action, which is intended to be a final action, will be governed by the requirements of 37 CFR 1.116, which will be strictly enforced.

A shortened statutory period for response to this action is set to expire **two months** from the mailing date of this action.

Extensions of time under 37 CFR 1.136(a) do not apply in reexamination proceedings. The provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Further, in 35 U.S.C. 305 and in 37 CFR 1.550(a), it is required that reexamination proceedings "will be conducted with special dispatch within the Office."

Extensions of time in reexamination proceedings are provided for in 37 CFR 1.550(c). A request for extension of time must be filed on or before the day on which a response to this action is due. The mere filing of a request will not effect any extension of time. An extension of time will be granted only for sufficient cause, and for a reasonable time specified.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andrew Caldwell, whose telephone number is (703) 306-3036. The examiner can normally be reached on M-F from 9:00 a.m. to 5:30 p.m. EST.

If attempts to reach the examiner by phone fail, the examiner's supervisor, Glenton Burgess, can be reached at (703) 305-4792. Additionally, the fax numbers for Group 2100 are as follows:

Fax Responses: (703) 872-9306

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

Andrew Caldwell

Andrew Caldwell
703-306-3036
February 25, 2004

90006831 . 022504

831 PH Ex. 5

12
Pl.
55

PATENT
Attorney Docket No.: 006-1-1
Client Reference No: 94-108-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re reexamination application of: DOYLE et al.	Examiner: Caldwell, A. T.
Application No.: 90/006,831	Art Unit: 2157
Filed: October 30, 2003	Interview Request Continuation Sheet
For: DISTRIBUTED HYPERMEDIA METHOD FOR AUTOMATICALLY INVOKING EXTERNAL APPLICATION PROVIDING INTERACTION AND DISPLAY OF EMBEDDED OBJECTS WITHIN A HYPERMEDIA DOCUMENT	

BRIEF DESCRIPTION OF THE ARGUMENTS

I. CLAIMS 1 AND 6.

A. Scope of the claim

1. executable application is automatically invoked when embed text format is parsed by the browser in order to display the object and allow in-place interaction while the web page is being displayed.

B. Exhibits

1. Animation of scope of claim 6.
2. Flow chart
3. Slides

II. THE DISCLOSURE OF THE REFERENCES

A. Applicant's admitted prior art (Mosaic browser application)

1. The browser application is utilized as a viewer to read HTML documents published on the World Wide Web.
2. The browser retrieves a published Web Page in response to a user's command and stores a local copy of the retrieved HTML page source files in a temporary cache.
3. There is no further interaction with the published source HTML document files after they are retrieved unless the user clicks the refresh button.
3. The browser parses the local copy of the HTML page to form a rendered image of the page which is displayed by the browser to the user.
4. The browser allows an author to use the IMG and FIG tags to embed, in a source HTML document, in-line graphic images which are treated as characters when the page is rendered.
5. The IMG and FIG tags include a *src* attribute that identifies an image data file external to the document that is retrieved by the browser and rendered into a static graphic image.

6. The user could invoke a helper application, which operated as an independent program, from Mosaic to view data in non-native format. When the helper application became active Mosaic would become inactive.

B. Berners-Lee

1. A specification for the HTML mark-up language used by Web authors to describe the structure and desired content of their pages.
2. Describes a model in which Web pages are written by an author, then distributed by a Web server to a browser, and then viewed as static items by the browser's user.
3. The user views a page and then clicks a hyperlink or button, or enters some text in an address field, to view another page.

C. Raggett I

1. Is a web-posted Document entitled **HTML+** that proposes a set of slight modifications to Berners-Lee.
2. States that inlined static graphic images are treated like characters and inserted as part of the text.
3. Defines an EMBED tag that extends the concept of inlined static images to support foreign data formats that cannot be rendered by the browser itself.
4. Gives the example of rendering ASCII text included within the embed tag into a static image of an equation.
5. States that the EMBED tag can be used as a substitute for the src attribute within a FIG tag
6. States that an external filter or shared library identified by the browser based on a MIME type is a rendering application that renders embedded data by returning a static image such as a pixmap.
7. States that sophisticated browsers can link to an external editing application to edit embedded data.

D. Raggett II

1. Is an email message stating that the EMBED tag of Raggett I has the capability of to embed foreign formats, such as equations and encapsulated Postscript, inline in the HTML+ source.
2. States that X resources or a config file can be used to bind MIME content type to the rendering application for the format.
3. States that the source file holding the foreign data can be external to the HTML+ source and referenced by a URL.

III. ARGUMENT

A. The references do not disclose or teach the features recited in claims 1 and 6.

1. The rendering application and external editing application described in Raggett I operate in completely different ways to perform different functions.

a. The rendering application is invoked by the browser to render foreign data, into a static graphic image, to be treated as a character and inserted into the text. The rendering application returns the static image and terminates.

b. The external editing application would be launched as a helper application in a separate window. There would be no communication between the helper application and Mosaic after the external editing application was launched.

c. Raggett I and II both teach that the rendering applications can be implemented as filter applications using pipes. This teaching requires that a rendering application must complete its processing and terminate prior to the display of the web page.

d. Raggett I's teaching to use the EMBED tag as a substitute for the src attribute in FIG tags requires that EMBED tags must return static images for display in the web page.

2. The statement in Raggett I that sophisticated browsers could link to an external editing application teaches away from the claimed element of automatically invoking an executable application in order to display the object and to enable in-place interaction.

3. The external editor is to be used to "create or revise" images to be embedded. This activity can only be undertaken by the author of the web page, before the page is published to the end user.

4. The rendered image of the source HTML+ document would not change if the foreign data held in the EMBED source file were edited.

5. There is no teaching in Raggett I or II to create any new editors, for use with the EMBED tag, or to modify any existing editors. Therefore he is proposing the use of existing editors, none of which were able to provide the claimed functionality of being automatically invoked in order to display the object and allow in-place interaction while the web page is being displayed.

B. Real World Considerations.

1. Raggett I and Raggett II were both exhibits at the trial of Eolas Technologies and the University of California v. Microsoft, in the summer of 2003, and Raggett testified as to their meaning. In his testimony, he admitted, regarding the "external editor" described in the EMBED tag specification, that "you might want to be able to pop up a kind of like an editor for mathematics which might have menus. So it's a simple thing. You might pop up a separate window with a pallet with different kinds of mathematical symbols."

2. The EMBED tag described in Raggett I and Raggett II was abandoned by Raggett before the end of 1993, and was never implemented or accepted as an addition or modification to the HTML of Berners-Lee.

3. Authors continued to used the IMG and FIG tags of Berners-Lee to embed inline static images and never adopted the proposed EMBED tag of Raggett I and Raggett II.

PTO/SB/21 (08-00)

Please type a plus sign (+) inside this box → [+]

Approved for use through 10/31/2002. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	REX Control No. 90/006,831
Filing Date	10/30/2003
First Named Inventor	Michael D. Doyle
Group Art Unit	2151
Examiner Name	Andrew Caldwell
Attorney Docket Number	006-1-1
Total Number of Pages in This Submission	5

ENCLOSURES *(check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment / Response
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53

- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition Routing Slip (PTO/SB/69) and Accompanying Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s)

- ☐ After Allowance Communication to Group
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
- ☐ Proprietary Information
- ☐ Status Letter
- ☒ Other Enclosure(s) *(please identify below)*:

Applicant Initiated Interview Request Form

Remarks: The Commissioner is authorized to charge any additional fees to Deposit Account 502267.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Charles E. Krueger Reg No. 30,077
Signature	
Date	April 22, 2004

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on this date:
NOT MAILED; FAXED TO (70~~2~~3) 746-5507 ATTENTION EXAMINER ANDREW CALDWELL April 22, 2004

Typed or printed name	~~Sharon D. Krueger~~ Charles E. Krueger
Signature	Date April 22, 2004

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be send to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

SF 1244040 v1

PTOL-413A (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Applicant Initiated Interview Request Form

Reexamination Control No.
Application No.: 90/006,831 First Named Applicant: Michael D. Doyle

Examiner: A. Caldwell Art Unit: 2151 Status of Application: Reexamination of U.S. Patent No. 5,838,906

Tentative Participants:
(1) Michael D. Doyle (2) Charles E. Krueger
(3) ________ (4) ________

Proposed Date of Interview: 4/27/04 Proposed Time: 2:00 (AM/PM)

Type of Interview Requested:
(1) [ ] Telephonic (2) [X] Personal (3) [ ] Video Conference

Exhibit To Be Shown or Demonstrated: [X] YES [ ] NO
If yes, provide brief description: Computer animation and slides.

Issues To Be Discussed

Issues (Rej., Obj., etc)	Claims/ Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) Rej.	1 and 6	Applicants' admitted prior art, Berners-Lee, Raggett I and Raggett II	[ ]	[ ]	[ ]
(2)			[ ]	[ ]	[ ]
(3)			[ ]	[ ]	[ ]
(4)			[ ]	[ ]	[ ]

[ ] Continuation Sheet Attached

Brief Description of Arguments to be Presented:

See continuation sheet attached.

An interview was conducted on the above-identified application on ________.

NOTE:
This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).
This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.

(Applicant/Applicant's Representative Signature) (Examiner/SPE Signature)

This collection of information is required by 37 CFR 1.133. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 21 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

831 PH Ex. 6

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO	CONFIRMATION NO.
90/006,831	10/30/2003	5838906		9718

30080 7590 04/27/2004

LAW OFFICE OF CHARLES E. KRUEGER
P.O. BOX 5607
WALNUT CREEK, CA 94596-1607

EXAMINER	
Andrew Caldwell	
ART UNIT	PAPER NUMBER
2151	13

DATE MAILED: ~~04/27/2004~~

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

***Ex Parte* Reexamination Interview Summary**	**Control No.** 90/006,831	**Patent Under Reexamination** 5838906
	Examiner Andrew Caldwell	**Art Unit** 2151

19
pcl
55

All participants (USPTO personnel, patent owner, patent owner's representative):

(1) *Andrew Caldwell* (3) *Charles Krueger*

(2) *Michael Doyle* (4) *Pinchus Laufer, Elizabeth Dougherty (PTO)*

Date of Interview: *27 April 2004*

Type: a)□ Telephonic b)□ Video Conference
c)☒ Personal (copy given to: 1)□ patent owner 2)☒ patent owner's representative)

Exhibit shown or demonstration conducted: d)☒ Yes e)□ No.
If Yes, brief description: *See attachment*

Agreement with respect to the claims f)☒ was reached. g)□ was not reached. h)□ N/A.
Any other agreement(s) are set forth below under "Description of the general nature of what was agreed to..."

Claim(s) discussed: *1 & 6*.

Identification of prior art discussed: *Berners-Lee, Raggett I & II, and Mosaic*.

Description of the general nature of what was agreed to if an agreement was reached, or any other comments:
Mr. Doyle presented the material in the attachment entitled "Interview with Examiner Andrew Caldwell April 27, 2004." It was agreed that a written response incorporating these arguments would be filed. Mr. Doyle also provided various definitions from the Microsoft Press Computer Dictionary, a copy of which is attached.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims patentable, if available, must be attached. Also, where no copy of the amendments that would render the claims patentable is available, a summary thereof must be attached.)

A FORMAL WRITTEN RESPONSE TO THE LAST OFFICE ACTION MUST INCLUDE PATENT OWNER'S STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. (See MPEP § 2281). IF A RESPONSE TO THE LAST OFFICE ACTION HAS ALREADY BEEN FILED, THEN PATENT OWNER IS GIVEN **ONE MONTH** FROM THIS INTERVIEW DATE TO PROVIDE THE MANDATORY STATEMENT OF THE SUBSTANCE OF THE INTERVIEW (37 CFR 1.560(b)). THE REQUIREMENT FOR PATENT OWNER'S STATEMENT CAN NOT BE WAIVED. **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).**

Andrew Caldwell
Examiner's signature, if required

cc: Requester (if third party requester)

[illegible]

'906 Patent Reexamination

Interview with Examiner Andrew Caldwell
April 27, 2004

Michael D. Doyle, Ph.D.
Charles E. Krueger, Attorney

13

906 Patent Reexamination

- A decade ago, before ease of interactivity had become a key ingredient to the popular success of the Internet, the World Wide Web was in transition from laboratory to dormitory. Far from today's easy-to-use browser technology with seemingly ubiquitous interactivity, the World Wide Web then consisted of a large collection of static pages through which a user could navigate using a Web browser. As the technology progressed, still images were added to the Web collection; however the user was still only able to access the information, not interact with it. While early Web participants struggled to implement helper applications, researchers at the University of California were already examining the potential of the Web to become a platform for fully interactive embedded applications: The '906 invention was born.

'906 Patent Reexamination

- Claims 1 and 6
  - Scope of the claim
    - Executable application is automaticaly invoked, when an embed text format is parsed by the browser, in order to display the object and allow in-place interaction while the web page is being displayed
    - Animation of claim 6

The References

- Berners-Lee
  - Provides a specification for the HTML document language
- Raggett I and II
  - Proposed use of a tag called EMBED for specification of static inline images
- Mosaic
  - Early web browser that supported helper applications

The Grounds of Rejection

- States that Raggett I's embed text format, type information, and automatic invocation are equivalent to 906 teachings
- States that external editors provide interactive control of embedded data
  - "These external editors that create or revise the embedded data would work the same way as the simple example of providing equation support"
- States that the claimed invention would have been obvious over Mosaic in view of Berners-Lee, Raggett I and Raggett II

Summary of References Arguments

- Rendering application and external editor operate in different ways to perform different functions
- The external rendering application of Raggett I and II would cease execution as it returned a static image to the browser, prior to the image being displayed to the user
- The use of EMBED within the FIG tag requires that EMBED return a static and non-interactive image
- The rendered image of the source HTML+ document would not change if the end user modified the locally-downloaded copy of the embedded image
- The statement in Raggett I that a browser could be made to link to an external editor teaches away from the claimed element of automatically invoking an executable application in order to display the object and to enable in-place interaction

HTML+ Specification
Inline graphics

- "treated like characters"
  - Therefore they are **static** pixmaps
  - "Sophistocated HTML+ editors should allow authors to modify images using an external editor. Larger images should be specified with the FIG tag"
  - Raggett I teaches here that only the web page author would need to modify an inline graphic image.
    - As Berners-Lee teaches, it is the web page **author** who creates and publishes the page content for retrieval by the end-user. Only the **author** can change the source data.

HTML+ Specification EMBED tag

- "for mathematical equations and simple drawings"
- "Images and complex drawings are better specified using the FIG or IMG elements."
    - It should be noted that 906 technology is used by modern browsers for complex datatypes that browsers can't handle on their own
    - Raggett I **teaches away** from this use
    - This is because use of EMBED for larger or more complex graphics would have a negative impact on page display speed – because the rendering application would have to **finish** computation **before** the page is displayed

Eolas v. Microsoft Summer of 2003

Ragget I and Ragget II were exhibits at trial, and Dave Raggett, himself, testified about them. Princeton Professor Edward Felten also testified, giving an expert opinion about the meaning of the Raggett documents.

Edward Felten testimony :

Q Now, does the work that Mr. Raggett did with the embed text have any relationship to what the embed text is used for in the '906 patent?

A No, it's an **entirely different** thing. If you are looking for similarities between them, it doesn't go much beyond having the text called "embed."

"...And so really what's happening here with HTML Plus is a slightly fancier way of putting **static** images into web pages. There's **no interactivity** here, and some of the other elements required in the '906 claims are also absent."

HTML+ Specification EMBED tag -- Filter

- Raggett I and II's filter application renders data and then **returns** a pixmap
- Execution ends **before** browser uses returned data to render page
- Raggett I gives two examples which result in static images in the web page
- Filters are **non-interactive**
- Raggett I and II teach implementing the rendering filter applications through UNIX pipes

HTML+ Specification EMBED tag -- Pipes

- UNIX pipes are treated as files by the calling program

- In this context, reading the data stream from a pipe is just like reading from a file stream
  - The src attribute specifies a **static** graphic file
  - The ability to substitute an EMBED tag for the src attribute in the FIG tag shows that, to the FIG tag code, EMBED would have behaved like a **static** graphic file

HTML+ Specification FIG tag

- Teaches that you can use the EMBED element in place of the src attribute in order to define the image data
  - You can substitute the EMBED-defined pipe, for the src-defined file stream because, to the FIG tag code, they look the same
  - FIG tag is clearly intended for use with **static** data
  - Image maps are a feature of FIG
    - They provide pre-defined active areas that can be associated with hypertext links
    - A user's click on one of these active areas would cause the browser to fetch a new web document
    - If the image data in an image map changes, the active areas lose their semantic correspondance, they lose their meaning
    - Since Raggett I teaches that EMBED should work with image maps, it **cannot** refer to a method for specifying dynamically-changing image data

HTML+ Specification FIG tag

- A mouse click can **only** mean **one** thing at a time
- The image map feature of the FIG tag would have obviated any ability to interact with EMBED-based images beyond the simple clicking of an image map
- Any mouse clicks on an EMBED-based FIG-tag image would have been captured by the image map code of the FIG tag. The **EMBED**-based image, itself, would have to be **dead**.
- This means that the use of the proposed EMBED tag, itself, was appropriate only for the **non-interactive** display of image data.

HTML+ Specification FIG tag

- If Raggett I had meant EMBED to support image data that can be dynamically changed and interactively controlled during the viewing of the Web page, this would have created a logical inconsistency that would have required discussion in the section of the FIG tag specification relating to image maps

- Since the reference was actually referring to **static and non-interactive** image data, no logical inconsistency existed, so no discussion was given

HTML+ Specification
EMBED tag -- Editor

- "Sophistocated browsers can link to external editors for creating or revising embedded data"
  - In the context of Raggett I, a browser that supported helper apps would be a sophisticated browser
  - It is important to note that "Sophistocated" modifies browser, not the external editor
- "linked to"
  - Means hyperlinked
    - Therefore the editor is **not automatically invoked**
  - Combination with Mosaic teaches that the helper application paradigm should be used
    - External application would not be automatically invoked and would open in a **separate window**
    - **No ongoing communications** between browser and external app

Eolas v. Microsoft
Summer of 2003

Raggett testimony:

Q Let me direct your attention to the fourth line from the bottom where it says, "Sophisticated browsers can link to external editors for creating or revising embedded data." Do you see that, sir?

A Yes.

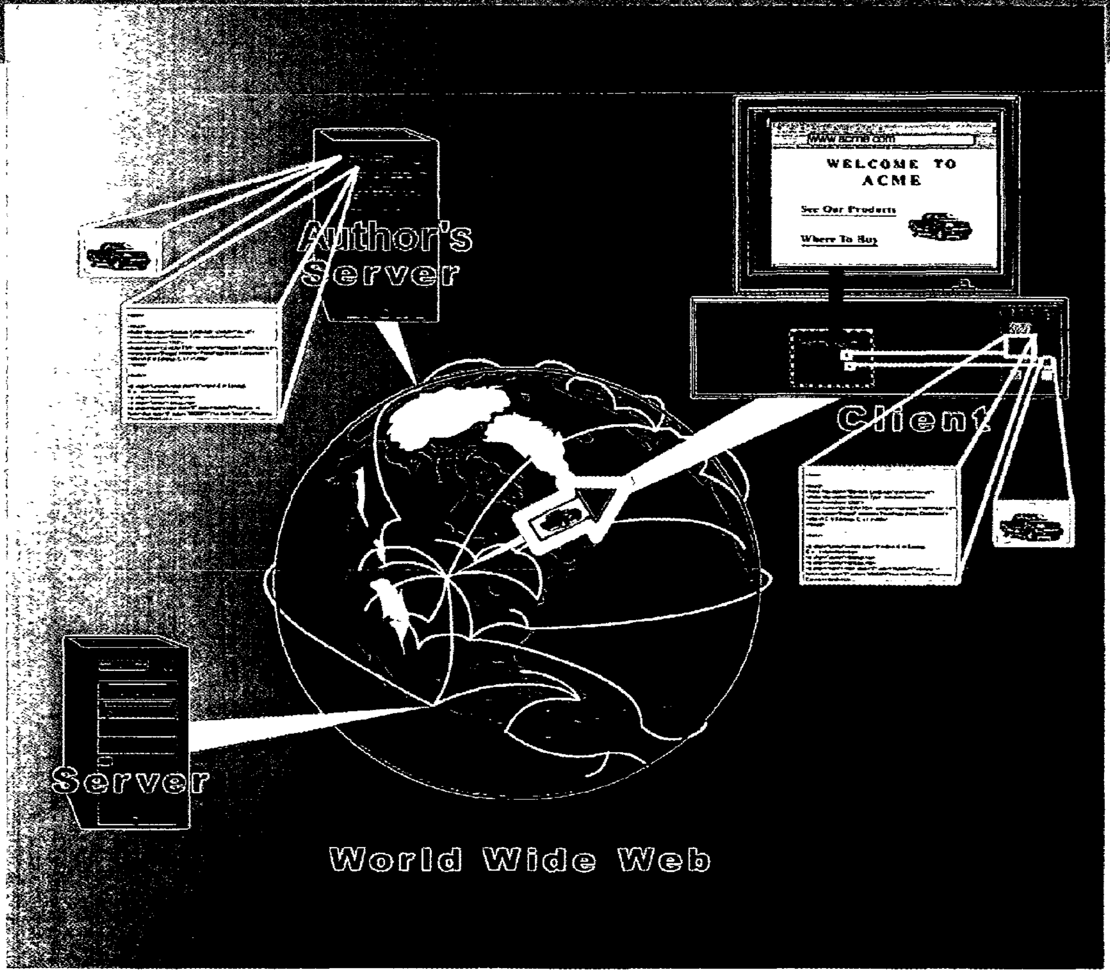
Q What does that mean?

A In the example of the mathematical equation, you might want to be able to **pop up** a kind of like an editor for mathematics which might have menus. So it's a simple thing. You might **pop up a separate window** with a pallet with different kinds of mathematical symbols.

HTML+ Specification EMBED tag – Editor

- "create or revise"
  - Can only be done by the web page **author** prior to publishing the page on the author's server.
  - The rendering application on the end-user's computer cannot be used to “create or revise”
    - The rendering application is not even active while the creating or revising is occurring
- It would be self-defeating for remote browser to try to edit a locally-downloaded image
  - Raggett I and II provide **no teaching** for how any editing program would work
  - Since the user would only be editing the **local** temporary file in the cache
  - The end user can't upload changes back to the server, **only** the web page **author** can do that
  - The first time the page is refreshed or returned to the changes would be **overwritten. The page display would be unchanged, and would not reflect any changes made by the end-user.**

Editor vs. Client



HTML+ Specification EMBED tag – Editor

- "**Sophistocated**" modifies the term "**browser**", **not "external editors"**
- There is **no teaching or suggestion** in Raggett I or II of creating new editors or modifying existing editors
  - "It allows authors to continue to use familiar standards, such as TeX and eqn."
- No existing external editor at the time of the filing of the 906 patent was able to communicate with a browser to dynamically change the rendered view of a web page.

Summary of References Arguments

- Rendering application and external editor operate in **different** ways to perform **different** functions
- The external rendering application of Raggett I and II would cease execution as it **returned** a static image to the browser, **prior** to the image being displayed to the user
- The use of EMBED within the FIG tag requires that EMBED return a **static and non-interactive** image
- The rendered image of the **source** HTML+ document would not change if the end user modified the locally-downloaded copy of the embedded image
- The statement in Raggett I that a browser could be made to link to an external editor **teaches away** from the claimed element of automatically invoking an executable application in order to display the object and to enable in-place interaction

Real World Considerations

- The early demonstrations of the '906 invention were **enthusiastically received** by the scientific visualization community, and Dr. Doyle was invited to present it, in 1993 and 1994, at many prestigious institutions and at several highly-regarded conferences.
- The EMBED tag of Raggett I and II was **abandoned** by Raggett, after the www-talk group asked him to drop it, and it was never implemented by others
- Authors continued to use the IMG and FIG tags of Berners-Lee to embed inline graphics and **never** adopted the proposed EMBED tag of Raggett I and II

Real World Considerations

- Raggett testimony in Eolas v. Microsoft, 2003

CROSS EXAMINATION

Q Netscape plug-ins had the ability to interact with an embedded program object in a web page, right, sir?

A That is correct.

REDIRECT EXAMINATION

Q And you envisioned that, didn't you?

A **I can't say I did.** ...

831 PH Ex. 7

16548 U.S.PTO
05/11/04

PATENT
Attorney Docket No.: 006-1-1
Client Reference No: 94-108-1

14
PL
5-13

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re reexamination application of:	Examiner: Caldwell, A. T.
DOYLE et al.	Art Unit: 2151
Application No.: 90/006,831	Response
Filed: October 30, 2003	
For: DISTRIBUTED HYPERMEDIA METHOD FOR AUTOMATICALLY INVOKING EXTERNAL APPLICATION PROVIDING INTERACTION AND DISPLAY OF EMBEDDED OBJECTS WITHIN A HYPERMEDIA DOCUMENT	

Commissioner for Patents

Sir:

In response to the Office Action mailed 03/12/2004, please consider the following remarks:

REMARKS

Claims 1-10 have been reexamined and are now pending in the application. Reexamination and reconsideration of all outstanding rejections and objections is requested.

Claims 1 and 6 are rejected under 35 U.S.C. §103(a) as being unpatentable over the admitted prior art in the U.S. Patent No. 5,838,906 ('906 patent) and the newly cited teachings of Berners-Lee, Raggett I, and Raggett II.

Introduction

Included with this response are Rule 132 Declarations by Professor Edward W. Felten, Professor of Computer Science at Princeton University ("Felten"), traversing the rejections of claims 1 and 6 of U.S. Patent No. 5,838,906 ("the '906 patent"), by Dr. Michael Doyle, one of the named inventors on the '906 patent ("Doyle"), stating facts relating to reactions by experts in the field at the time the technology recited in claims 1 and 6 of the '906 patent was introduced, and by Charles E. Krueger, attorney of record ("Krueger"), setting forth testimony from the Eolas v. Microsoft trial and other exhibits. References to these declarations will be made in the following arguments.

It is Applicants' position that the references referred to below as Raggett I and Raggett II are not publications according to 35 U.S.C. §102. However, for the purposes of the following arguments those references are being treated as if they were prior art.

Outline of the Argument for Claims 1 and 6

A. The Claimed Invention
B. Description of the References
    1. Applicants' Admitted Prior Art (Mosaic)
    2. Berners-Lee
    3. Raggett I
    4. Raggett II
C. What the References Teach to a Person Having Ordinary Skill in the Art (PHOSA)
    1. Level of Skill in the Art
    2. The References
D. The Examiner's Reasoning
    1. Raggett I
    2. Raggett II
E. Traverse

Part I. The proposed combination does not show the claimed features of the Applicants' invention. There is no suggestion or teaching in Raggett I or II, singly or in combination, of modifying Berners-Lee and/or applicants' admitted prior art to automatically invoke an external application to execute on a client computer, when an embed text format is parsed, to display and interactively control an object in a display window in a document being displayed in a browser-controlled window on the client computer.

a. The rendering application and external editor of Raggett I and II operate in different ways to perform different functions. The use of an external editor to create and revise embedded data does not teach in-place interaction with an object displayed within a browser-controlled window.

b. There can be no interactivity with the image displayed in the browser window because the external rendering application of Raggett I and II would cease execution when it returned a static image to the browser, prior to the image being displayed to the user.

c. The teaching in Raggett I and II to use Raggett's proposed EMBED tag within the FIG tag requires that Raggett's proposed EMBED tag return a static and non-interactive image.

d. The rendered image of the source HTML+ document displayed by the browser would not change if the end user utilized an external editor to modify the locally-downloaded copy of the embedded image.

e. The statement in Raggett I that a browser could be made to link to an external editor teaches away from the claimed element of automatically invoking an executable application on the client workstation in order to display the object and to enable in-place interaction.

PART II. The teachings of the references would have to be fundamentally modified to meet the limitations recited in claims 1 and 6. Such modifications could only be implemented by impermissibly utilizing claims 1 and 6 as a roadmap to modify the teachings of the references.

PART III. The secondary considerations of failure of others and professional approval further support the conclusion of non-obviousness.

a. Professional approval.

b. Failure of others to follow Raggett I and Raggett II to implement the claimed technology.

DETAILED ARGUMENT

A. The Claimed Invention.

The invention, as recited for example in claims 1 and 6, is for use in a system having at least one client workstation and one network server coupled to a network environment.

The claims recite a browser application, executed on the client workstation, that parses a hypermedia document to identify text formats in the document and responds to predetermined text formats to initiate processing specified by the text formats.

The browser displays a portion of a first distributed hypermedia document, received over the network from the network server, in a browser-controlled window. The hypermedia document includes an embed text format, located at a first location in the hypermedia document, that specifies the location of at least a portion of an object external to the hypermedia document. The object has associated type information utilized by the browser to identify and locate an executable application external to the hypermedia document.

When an embed text format is parsed by the browser, the executable application is automatically invoked, as a result of the parsing, to execute on the client workstation.

When the automatically invoked application executes on the client workstation, the object is displayed within a display window created within the portion of the hypermedia document being displayed and interactive processing of said object is enabled.

B . Description of the References

1. Applicants' Admitted Prior Art

The specification of the '906 patent (Applicants' Admitted Prior Art) describes a browser application, e.g., Mosaic, that functions as a viewer to view HTML documents. There are several ways to retrieve an HTML document from a network server, all of which require user interaction with the browser. [Felten, paragraph 8] When the browser is launched on a client workstation, a home page may be retrieved, a URL saved in a favorites list may be selected, or a link in a displayed page may be selected. The browser then retrieves a selected HTML published source document from a network server utilizing a uniform resource locator (URL) that locates the HTML document on the network and stores a temporary local copy of the HTML source document in a cache on the client workstation.

The browser application then parses the local copy of the HTML document, renders the temporary local copy of the HTML document into a Web page , and displays the rendered Web page in a browser-controlled window. [Felten, paragraph 21] During the rendering step, the browser may retrieve information external to the local copy of the HTML document, such as source files referenced by IMG tags, render the images from the retrieved files as static graphic images, and insert the images into the Web page of the HTML document, for display to the user.

There is no further interaction with the source HTML document or the local copy of the source HTML document subsequent to its being rendered and displayed. If a user believes the source HTML document has changed (s)he can click a refresh button in the browser GUI which causes the browser application to retrieve the source HTML document from the network server again, store a local copy again, parse and render again the newly retrieved local copy of the source HTML document, and replace the display of the previous version of the retrieved source HTML document with the subsequently retrieved version in the browser-controlled window or another window. For example, if the source HTML document were a price list of goods the user might refresh the document to determine if the prices had changed.

Although the browser application passively displays links, from text or picture elements of a first hypermedia document to other external data objects, a user may browse by actively selecting links to retrieve information identified by a link. The retrieved information either replaces the first hypermedia document or is displayed in a separate window other than the window displaying the hypermedia document. Mosaic has the capability of allowing the user to interactively invoke an external application to open a new window to display file types that cannot be displayed by Mosaic (helper applications).

Some browsers, such as Mosaic, include the capability of rendering images in certain formats, such as GIF and JPG, designated as a native format. These images may be placed inline in an HTML document using the IMG element, which specifies a source location, URL, of the source file to be rendered by the browser, and displayed in the rendered format of the document. All static images referenced by IMG or FIG tags referenced in the HTML document must be retrieved by the browser prior to rendering the HTML document.

For data formats that can not be rendered by the browser application itself, i.e., data in a foreign or non-native format, Mosaic launches helper applications, in response to a user's interactive command, in a separate window to view certain types of file types. As described in the specification, the mechanism for specifying and locating a linked object is an HTML anchor "element" that includes an object address in the format of Uniform Resource Locator (URL).

Many viewers exist that handle various file formats such as ".TIF," ".GIF," etc. When a user commands the browser program to invoke a viewer program (helper application), typically by clicking on an anchor with a mouse, the viewer is launched as a separate program. The viewer program displays the image in a separate "window" (in a windowing environment) or on a separate screen. This means that the browser program is no longer active while the viewer program is active. The viewer program is completely independent of the browser after being invoked by the browser so that there is no communication between the viewer program and the browser program after the viewer program has been launched.

As a result, the viewer program continues to run, even after the browser program execution is stopped, unless the user explicitly stops the viewer program's execution.

Mosaic was a significant advance that made the WWW easily accessible and gave Web page authors a powerful tool to provide simplified user-activated access to viewing of hypermedia documents and related external data objects anywhere on the WWW network.

2. Berners-Lee (Berners-Lee, T., et al., Hypertext Markup Language (HTML), Internet Draft, IETF, pages 1-40, (June 1993)

The Berners-Lee reference is a specification for the HTML markup language. HTML is a language used by Web page authors to describe the structure and desired contents of their pages. A browser parses an HTML document to determine its structure and then displays the specified items as a rendered Web page within a browser window.

This reference describes a model in which Web pages are written by a Web page author, then distributed by a Web server to a browser, and viewed as a Web page displayed in the browser window by the browser's user. The user views a page, and then clicks a hyperlink or button, or enters some text, to select another page to view.

3. Raggett I (Raggett, D., HTML+(Hypertext Markup Language), (July 23, 1993))

The position of the Applicants is that Raggett I is not a publication complying with 35 U.S.C. §102. However, in the following it will be assumed that Raggett I is prior art.

Raggett I is a document entitled "HTML+ (Hypertext Markup Language) A proposed standard for a light weight presentation independent delivery format for browsing and querying information across the internet" [emphasis added]. In pertinent part, Raggett I generally relates to allowing Web page authors to display static images of equations and simple drawings in a Web page. At page 3, describing the HTML+ Document Format, it is stated that "HTML+ departs slightly from pure presentation independence by allowing Web page authors to specify rendering hints to give Web page authors greater control over the final appearance of documents."

At pages 4 and 5, Inlined Graphics or Icons are discussed. It is stated that these elements are treated like characters in the text and an example of the IMG tag is given:

This line has a egyptian hieroglyph at the end of the
line. <img src = "ankh.tiff">

It is further stated that the URL notation is used to name the source of the graphics data and that sophisticated HTML+ editors should allow Web page authors to modify images using an external editor. It is also stated that larger inlined images should be specified with the FIG tag.

At page 6, Raggett's proposed EMBED tag is described that provides a simple form of object level embedding that is very convenient for mathematical equations and simple drawings. Raggett's proposed EMBED tag would allow Web page authors to continue to use familiar standards, such as *TeX* and *eqn*. It is also stated that images and complex drawings are better specified by using the FIG or IMG elements.

Raggett's proposed EMBED tag would utilize a type attribute to specify a MIME content type to be used by a browser to identify a rendering application, such as a shared library or external filter, used to render embedded data. An example of rendering the embedded data is given as returning a pixmap which is a data structure holding a static image.

An example of Raggett's proposed EMBED tag is given as follows:

```
<embed type="application/eqn">2 pi int sin(omega t)dt</embed>
```

In this example the embedded data is "2 pi int sin(omega t)dt" and the type information is "application/eqn". In this example, the embedded data is processed by the *eqn* application to render a static graphic image of the embedded data in the following form:

$$2\pi \int \sin(\omega t)dt$$

The reference also states that sophisticated browsers can link to external editor applications for creating and revising embedded data.

It is also stated at page 12 that when using the FIG tag, instead of using a *src* attribute, an EMBED element can be included immediately following the <FIG> tag and that this is useful for simple graphs etc. defined in an external format.

At page 13 the *ismap* attribute of the FIG tag is described. It is stated that arbitrary areas of the figure can be designated as hypertext links.

There is no disclosure in the reference relating to building a browser or how a browser works.

4. Raggett II (Raggett, D., Posting of Dave Raggett, dsr@hplb.hpt.hp.com to www-talk@nxoc01.cern.ch (W-WWW-TALK public mailing list) (Posted June 14, 1993))

The position of the Applicants is that Raggett II is not a publication complying with 35 U.S.C. §102. However, in the following it will be assumed that Raggett II is prior art.

Raggett II is an email message from David Raggett to Torben Nielsen and Bill Janssen having the subject line "HTML+ support for eqn & Postscript".

This reference quotes an email from Nielsen stating that he has lots of documents he wants to put on the Web and that without support for equations it is quite difficult. It also quotes an email from Janssen stating he would like to send encapsulated Postscript in his documents.

The email then states that the HTML+ DTD makes both these requests possible by providing the capability to embed foreign data inline in the HTML source. The document then gives an example of Raggett's proposed EMBED tag and states that the browser identifies the format of the embedded data from the "type" attribute. It is also stated that building in support for a large number of formats has the danger of leading to very large programs for browsers and that this can be avoided by using a common API for rendering foreign formats, e.g., as rendering functions that take a sequence of bytes and return a pixmap.

It is then stated that browsers can then be upgraded to display new formats by binding MIME content type to the function name for that format and that the functions could be implemented as separate programs driven via pipes and stdin/stdout or as dynamically linked libraries (DLLs). It is also stated that foreign data can be put in a separate file referenced by a URL.

C. What the References Teach to a Person Having Ordinary Skill in the Art (PHOSA)

1. Level of Skill in the Art

The benchmark for a person having ordinary skill in the art (PHOSA) is a person who is just graduating from a good computer science program at a college or a university, not a star student but just a typical, average student, or a person who has gained equivalent knowledge in the industry. This person knows how to do things in conventional ways but does not exhibit an unusual level of innovative thinking. [Felten, paragraph 15]

2. The References

A person having ordinary skill in the art (PHOSA) would have been familiar with Mosaic as a browser application that allowed documents specified by URLs to be retrieved from a remote server over the Internet and to be displayed on a client computer in Mosaic's windows, and also would have been familiar with the various ways of using the Mosaic GUI to retrieve Web pages, e.g., using the favorites list or the refresh button. [Felten, paragraph 16]

The PHOSA would also have been familiar with HTML as described in Berners-Lee. A Web page author is able to compose HTML documents to be published using a simple text editor, such as MS notepad, and could utilize Mosaic to view the Web page based on the HTML document. [Felten, paragraph 24] A Web page author could then modify the document using the text editor and click the refresh button on Mosaic to view the changes

The PHOSA would have been familiar with inlined images, in particular the IMG and FIG elements of HTML. The IMG tag utilized a URL to reference source data as an external object to be rendered by Mosaic and inserted as part of the text in the rendered form of the source HTML document. [Felten, paragraph 21]

Raggett I proposes minor changes to Berners-Lee that have not yet been implemented. [Felten, paragraph 29,63] These changes relate to the techniques used by the author of an HTML+ document for defining the logical structure of a text document to be parsed by Mosaic. Raggett II is a response to email questions suggesting that authors of Web pages should be allowed to embed foreign formats, such as equations and static encapsulated postscript files, inline using the proposed HTML+ standard described in Raggett I. [Felten, paragraph 32]

Accordingly, the Raggett I and II references propose a minor modification of the inline imaging technique to allow data in foreign formats, i.e., data formats not capable of being rendered by Mosaic itself, to be rendered into a static image returned by an external rendering application. [Felten, paragraph 36]

Raggett I gives two concrete examples of an external rendering application that utilizes a text field which is part of the HTML+ document as input data, creates an image of an equation based on the text field, and returns a pixmap. Raggett I states that the rendering application could be an external filter. Raggett II states that the rendering applications could be implemented as separate programs driven via pipes and stdin/stdout or as dynamically linked library modules (Windows DLL).

The terms used in Raggett I and II have very precise meanings. Filters, pipes, and stdin/stout are well-known, well-defined UNIX terms. A filter is a program that reads its standard input (stdin) or designated input, transforms the input in some desired way, and then writes the output to its standard output (stdout) or designated output destination. A pipe connects two processes so that the output of one can be used as the input to the other. A DLL is a piece of code callable at run time in an MS windows environment that can be shared among processes. The term DLL does not specify any functionality of the code. [Felten, paragraph 54,55]

Thus, Raggett I and Raggett II teach that the rendering application would be a non-interactive application, such as a filter, that receives the foreign data, transforms the data into a static image, returns the data to Mosaic, and terminates prior to the Web page being displayed to the user. [Felten, paragraph 38-43]

Raggett I suggests that Mosaic would use a MIME type to identify an appropriate filter or DLL to render the data, that a "pixmap" is a term of art for a data structure describing a static graphic image, that "render" is a term of art that refers to the generation of a static image that is to be "displayed", and that "return" is a term of art referring to information produced by a program when the program terminates. [Felten, paragraphs 42,52]

Raggett I and II do not teach how to modify Mosaic to implement his proposed technique of embedded data in a foreign format. However, the references suggest that, during the step of rendering the local copy of the HTML document into a Web page to be displayed to the user, Mosaic would invoke an external rendering application that would access data in a foreign format and that would return a static graphic image, based on the foreign data, to Mosaic. The external rendering application would terminate after returning the static image and perform no further functions. Mosaic would then treat the returned static graphic image as a character and insert the static graphic image as part of the text when rendering the Web page to be displayed to the user. [Felten, paragraph 46]

The PHOSA familiar with Berners-Lee and Mosaic would be aware from using the IMG tag that image data can be maintained separately from the HTML source page and can be referenced by a URL included in the IMG tag. Thus, the PHOSA would understand that the statement in Raggett II that foreign data can be put in a separate file referenced by a URL is merely a restatement of the technique previously used by the IMG tag to create inline images inserted into the text.

Based on the discussion in Raggett I that Raggett's proposed EMBED tag can be used instead of the *src* attribute in the FIG tag, the PHOSA would understand that Raggett's proposed EMBED tag was to be utilized to retrieve static images. Further, based on the discussion of the *ismap* attribute of the FIG tag, the PHOSA would understand that only static images could be retrieved using the FIG tag so that areas of the returned static image can be designated as hypertext links. [Felten, 44]

Web page authors use text editors to create source HTML documents to be published and can utilize Mosaic to view the Web pages displayed based on those documents. The PHOSA would also realize that images to be presented inline must be created and at times revised and that a simple text editor cannot be utilized to create and revise these images. Thus the statement at page 5 of Raggett I, when describing inlined images, that "Sophistocated HTML+ editors should allow Web page authors to modify images using an external editor" means that a special external editor must be used to modify the image data. This external editor would be used to access the file containing the image data, modify the accessed data, and then save the data back to the image file.

The PHOSA would understand that the external editor and the rendering application invoked by Mosaic are different applications. The purpose of the external editor is to access the image file, modify the image file as determined by the Web page author, and then save the modified image file for subsequent publication on the author's Web server. [Felten, 47] The purpose of the rendering application is to access the image file, render a static graphic image such as a pixmap, return the static image to Mosaic, and terminate. Mosaic would then insert the static image into the page to be displayed. The Web page would then be displayed to the browser user.

Raggett I includes the statement "Sophistocated browsers can link to external editors for creating or revising embedded data". As described above, Berners-Lee and Mosaic teach that inline images cannot be edited by a simple text editor but require an external editor. Neither Raggett I or II describe how to link to the external editor.

The purpose of Mosaic is to function as a viewer application and not as an editing application. While the suggestion in Raggett I of a browser linking to an external editor to create or revise embedded data may be of some interest, the PHOSA would not fundamentally change Mosaic from its primary function as a viewer to implement this highly subsidiary editing function. Thus, the PHOSA, if choosing to implement this editing function, would utilize existing capabilities of Mosaic.

This editing function would only be useful to Web page authors editing the source HTML document to be published using the server and would be of no use to remote users viewing the source HTML document retrieved over the Internet on their client work stations. This is because the remote user would be unable to edit the image file that is referenced by the source HTML document to change the rendered form of the published source HTML document. [Felten, paragraph 24]

However, Mosaic can link to helper applications to display data in foreign formats. [Felten, paragraph 9] As stated above, the PHOSA does things in conventional ways and does not exhibit an unusual level of innovative thinking. Thus, the PHOSA would understand Raggett I to be suggesting that the browser would allow a Web page author to interactively click upon a link to a helper application in order to invoke an external editor which would be popped-up in a separate window if, when viewing a page being authored, it was desired to create or revise embedded data intended to be published. [Felten, paragraph 48]

A Mosaic helper application operates on a computer as a completely separate application from the browser application and there is no interaction of any kind between the two applications. [Felten, paragraph 9] There is no teaching in Raggett I or Raggett II relating to the operation of external editor programs. Existing editing programs could be used to create files and make changes to existing files. [Felten, paragraph 48] Raggett I and II teach that an existing editor program should be used for this function. [Felten, paragraph 48]

Only an author of a Web document can create or revise the source HTML document and any external files referenced by the source HTML document, and end-users viewing the source document can make no changes to the source document. [Felten, 24] Accordingly, an external editor executing on the client workstation could make no changes in the display of a source HTML document retrieved from a server over a network.

Thus, Raggett I and II teach that the inlined image feature of Mosaic could be slightly modified to utilize an external rendering application to render a data object in a foreign format and treat the rendered static graphic image like a character to be inserted into the text.

D. THE EXAMINER'S REASONING

1. Raggett I

The examiner states that Raggett I teaches a method wherein the "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 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".

The examiner recognizes that Raggett I describes an example where the browser calls a program for rendering an equation in ASCII character format into a pixmap image of the equation. However, it is then stated that Raggett I also recognizes that more sophisticated browsers can link to external editors for creating or revising embedded data.

The examiner then states that these external editors that create or revise embedded data would work in the same way as the simple example providing equation support and concludes that ability to create and revise the embedded data allows the user to interactively process the data within the browser window.

The examiner also concludes that it would have been obvious to a skilled artisan to combine (1) Raggett I's teaching regarding extensions to the HTML standard (i.e., the proposed HTML+ specification) allowing the embedding of data in foreign formats within Web pages with (2) the method as taught by patentee's admitted prior art. It is also stated that this combination would have been obvious based on Raggett I's acknowledgment that this particular extension to HTML is advantageous and it represents a "substantial improvement".

2. Raggett II

The examiner states that Raggett II teaches putting the foreign data in a separate file and then referencing that file by a URL in the HTML+ EMBED tag. It is also stated that Raggett II suggests combining the teachings of the admitted prior art in view of Berners-Lee and Raggett I by substituting a URL to reference external foreign data.

E. TRAVERSE

This rejection is respectfully traversed for the following reasons.

The entire Felten declaration is incorporated herein as an independent traverse of the rejection of claims 1 and 6. The following argument recapitulates parts of the traverse set forth in Felten, with citations to relevant parts thereof, and presents additional arguments not present in Felten.

PART I. The proposed combination does not show the claimed features of the Applicants' invention. There is no suggestion or teaching in Raggett I or II, singly or in combination, of modifying Berners-Lee and/or Applicants' admitted prior art to automatically invoke an external application to execute on a client computer, when an embed text format is parsed, to display and interactively control an object in a display window in a hypermedia document, received over a network from a network server, being displayed in a browser-controlled window on the client computer.

a. The rendering application and external editor of Raggett I and II operate in different ways to perform different functions. The use of an external editor to create and revise embedded data does not teach in-place interaction with an object displayed within a browser-controlled window.

It is stated in the Office Action that the external editors that create or revise embedded data would work in the same way as the simple example providing equations support. However, as described above, Raggett I describes two completely different executable applications and their relationship to the browser. The first is the <u>rendering application</u>, in the form of an external filter or dynamic linked library (DLL), that is identified by the browser and that renders the embedded data and returns a static image such as a pixmap. [Felten, paragraphs 36, 41, 42, 43] The second is an <u>external editor</u> used by a Web page author to create or revise image data in a file. [Felten, paragraph 47] Raggett I does not teach any relationship or interaction between the two applications.

Thus, a program that worked "in the same way" as the simple example of providing equation support, i.e., the <u>rendering application</u>, could not provide an editing capability or any other type of interactivity. [Felten, paragraph 59]

The statement in Raggett I that "Sophistocated browsers could link to external editors for creating and revising embedded data" does not refer to the rendering application. The statement would not teach or suggest any connection between the rendering application and the external editor to the PHOSA. Further, note that "sophistocated" modifies browsers not external editors. Accordingly, the statement does not teach or suggest any modifications to existing editing applications. [Felten, paragraph 48]

The Office Action states that the ability to create and revise embedded data allows the user to interactively process the data within the browser window. The functions of creating and revising data are performed by the external editor. However, an external editing application that would allow a user to interactively process the revised data within the browser window would have required a revolutionary redesign of existing editing applications, rendering applications, and browsers. [Felten, paragraph 49] External editors were designed to operate in their own windows. [Felten, paragraph 60] This redesign of an external editor to operate in a browser window could only have been accomplished using the teachings of the '906 patent as a roadmap.

This is because Raggett I gives no teaching relating to redesigning external editing applications and explicitly teaches away from such a redesign of the rendering application. [Felten, paragraph 60] With regard to the rendering application, Raggett I teaches that Web page authors can continue to use <u>familiar standards</u> such as *TeX* and *eqn* as the external editing application. [Felten, paragraph 41] Each of these applications is described in Raggett I and Raggett II as being implemented as a shared library, external filter, or DLL that would receive ASCII characters as inputs, convert the ASCII characters to a static image of an equation, and return the static image to the browser. [Felten, paragraphs 38-43]

Accordingly, there is no teaching or suggestion in Raggett I or Raggett II of modifying existing external editors or rendering applications to provide the claimed function of in-place interaction with an object displayed in the browser controlled window. [Felten, paragraphs 48,49]

> b. There can be no interactivity with the image displayed in the browser window because the external rendering application of Raggett I and II would cease execution when it returned a static image to the browser, prior to the image being displayed to the user.

Raggett I and Raggett II teach that a non-interactive application external to the HTML+ document renders a static image and returns a static image to the browser. The specific examples of rendering applications described in Raggett I and Raggett II, i.e., *eqn*, *TeX*, filters, pipes, and stdin/stdout, all describe non-interactive applications that receive data, transform the data, and return a static image without user interaction. [Felten , paragraphs 41, 42]

Further, all the image types described in Raggett I and Raggett II, i.e., a pixmap of an equation rendered by *TeX* or *eqn,* or an encapsulated Postscript file, are static, non-interactive images. [Felten, paragraph 50]

Also, Raggett I states that complex drawings should be specified using the FIG or IMG elements. These tags are utilized to embed static images in an HTML document and Raggett I teaches that Raggett's proposed EMBED tag can substitute for these tags when simple static images are to be rendered by an external rendering application. [Felten, paragraph 46]

Further, the rendering code, which is part of Mosaic, used to render images specified by the FIG and IMG tags, ceases execution after the images are rendered and prior to display of the Web page to the user by Mosaic. The use of Raggett's proposed EMBED tag in place of the *src* attribute in the FIG tag teaches that the external rendering application must cease execution when it returns the static image to the browser. [Felten, paragraph 44]

Accordingly, there can be no interaction with the image being displayed to the user within the browser window because the rendering application which creates the image has ceased executing and the image is static. [Felten, paragraph 42]

> c. The teaching in Raggett I and II to use Raggett's proposed EMBED tag within the FIG tag requires that Raggett's proposed EMBED tag return a static and non-interactive image. [Felten, paragraph 44]

Raggett I teaches uses of Raggett's proposed EMBED tag that require the returned image to be static. At page 12 it is stated that instead of using the *src* element Raggett's proposed EMBED element can be used following the FIG tag. It is known in the art that the FIG element is utilized to display static images in the displayed version of the HTML document. Since the use of Raggett's proposed EMBED tag, as a substiture for a *src*-defined static image file, in this context is not qualified, Raggett's proposed EMBED tag is required to return only a static image or it would cause the FIG tag to function incorrectly. [Felten, paragraph 44]

The requirement that Raggett's proposed EMBED tag return only a static image is further reinforced by the discussion in Raggett I of active areas at page 13. The *ismap* attribute described with respect to the FIG tag causes the browser to send mouse clicks on a figure back to the server using a selected coordinate scheme. Arbitrary areas of the figure can be designated as hypertext links. The Web page author thus creates a semantic correspondence between areas of the figure and Web pages that can be retrieved by clicking over these various areas. If the figure displayed were to be interactively changed then this semantic correspondence would be destroyed. Further, a mouse click can have only a single function. Since the *ismap* feature causes the browser to send mouse clicks to the server, the mouse click can not be utilized to interact with the image and the image must be static.

Thus, the ability to use Raggett's proposed EMBED instead of the *src* attribute with the FIG tag requires that a static and non-interactive image be returned. If this were not the case then Raggett I would require special discussion on the use of Raggett's proposed EMBED tag as an attribute with the FIG tag. No such discussion is included and thus Raggett I teaches that the image returned by Raggett's proposed EMBED tag must be static and non-interactive.

d. The rendered image of the source HTML+ document displayed by the browser would not change if the end user utilized an external editor to modify the locally-downloaded copy of the embedded image

As described above, the World Wide Web of Mosaic and Berners-Lee is a write-once-publish-many paradigm in which all users would see the same basic document content that the Web page author originally designed. This refers to the fact that a Web document author would create a single document file and publish that document merely by making it accessible on an Internet server. An unlimited number of users could then retrieve and view that document by simply entering the Internet address of the document (the URL) into their Internet-connected Web browsers executing on their client computers. The browsers would use a simple request/response protocol to retrieve specified documents and related data from remotely-networked Web server programs. [Felten, paragraph 20]

When a user clicks on a hyperlink in a document displayed by Mosaic, document text, image data, or external object data are retrieved by the browser executing on the client computer, over the network from the network server. These data are stored in local temporary cache files, to speed up document rendering for later requests for the same data. Since these data are only intended to be viewed, and not edited by the end-user, these cache files are only temporary, and are routinely automatically purged by the Web browser after a certain amount of time, or after it is determined that the user is not likely to need them again. This makes more efficient use of the end-user's disk drive space over time, preventing cluttering up of the disk drive with unneeded files. [Felten, paragraph 20]

Claims 1 and 6 recite that a browser application executes on the client workstation and displays a portion of a hypermedia document received over the network from the server and that an executable application executes on the client work station in order to display an object and enable interactive processing of the object within the browser-controlled window.

The Office Action reasons that the editing application is the executable application that would allow interaction with embedded data within the browser-controlled window. However, Raggett I and Raggett II describe techniques used by Web page authors to create source HTML documents to be placed on the server and published on the Web. In order to meet the limitations of claims 1 and 6 the editing application would have to execute on the client workstation, and have other characteristics as well.

As described above, in the write-once-publish-many paradigm of Mosaic and Berners-Lee only a temporary copy of the source HTML document is stored on the client machine. If an end-user utilized an editor executed on the client-workstation only a locally-cached data file could be edited. [Felten, paragraph 48] In the client-server model of Mosaic and Berners-Lee, the end user can't upload changes back to the server, only the Web page author can do that. If the end-user were to refresh the page the changes made to the locally-cached data would be overwritten and the display in the browser-controlled window would not change. [Felten, paragraph 23,24]

Accordingly, the references teach that an external editor executed on the client-workstation could provide no interactivity with embedded data displayed in the browser-controlled window. [Felten, paragraph 25,49]

> e. The statement in Raggett I that a browser could be made to link to an external editor teaches away from the claimed element of automatically invoking an executable application on the client workstation in order to display the object and to enable in-place interaction.

The statement in Raggett I that a browser could be made to link to an external editor teaches away from the claimed element of automatically invoking an executable application in order to display the object and to enable in-place interaction

Claim 6 recites:

> 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 interactive processing of 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.

In the Office Action, at page 5, lines 37-42, it is stated that the browser automatically invokes an executable application to display said object. This statement refers to the rendering application, i.e., the external filter or DLL, that renders the foreign data and returns a static image to the browser. It is the rendering application that is automatically invoked by the browser application. [Felten, 55]

The statement in Raggett I that "Sophisticated browsers can link to an external editor for creating or revising embedded data" does not suggest any similarity in function or manner of invocation between the rendering application as defined by Raggett I and II and the external editor mentioned in Raggett I.

In fact, Raggett I is very clear that such a sophisticated browser, if it did exist, would link to the external editor. As Raggett I states in the "Hypertext Links" section on page 4, "Links are defined with the <a href="#z1">A tag</a>..." Raggett I teaches that the browser user can click upon a link to navigate to another Web document. At the time the application was filed, the term "link" would teach the use of a helper application because that was the only mechanism provided by Mosaic to link to data in a foreign (non-native) format. [Felten, paragraph 9,10]

The author of Raggett I and II acknowledged this in sworn testimony at trial. Mr. Raggett was called as a witness by Microsoft Corporation in defense of a claim of infringement of claims 1 and 6:

> Question: Let me direct your attention to the fourth line from the bottom where it says, "Sophisticated browsers can link to external editors for creating or revising embedded data." Do you see that, sir?

> Answer: (Mr. Raggett) Yes.
> Question: What does that mean?
> Answer: (Mr. Raggett) In the example of the mathematical equation, you might want to be able to pop up a kind of like an editor for mathematics which might have menus. So it's a simple thing. You might pop up a separate window with a pallet with different kinds of mathematical symbols.
> Question: Could you use this technology with simple drawings?
> Answer: (Mr. Raggett) Sure. Pallets of drawing tools. [Krueger]

Thus, the only reasonable interpretation of the teaching of the quoted sentence, taken in the full context of the cited art and the level of skill of the art at the time the application was filed, is that sophisticated browsers could be modified to allow a Web page author to actively pop-up an external editing program operating as a separate program in a separate window from the browser. [Felten, paragraph 9,10] This sentence completely teaches away from the claimed concept of automatically invoking an external application to execute on the client workstation in order to display said object and to enable interactive processing within a display area in the browser-controlled window. [Felten, paragraph 48 (parenthetical)] The only application that would be automatically invoked in parsing Raggett's proposed EMBED tag of Raggett I and Raggett II is the <u>rendering application</u> which is not interactive.

Additionally, in a Mosaic/Raggett system the static graphic inline images are inserted into the rendered Web page displayed by the browser. On the other hand, the external editor operates as a helper application that is completely independent of the browser application. [Felten, paragraph 48] Thus, the Applicants' admitted prior art and Raggett I and Raggett II teach away from interactivity between a browser and an external editing application.

Accordingly, the required claim elements of <u>automatically invoking</u> an executable application to execute on the client workstation in order to display the object and enable interactive processing of the object within a display area in the browser-controlled window are not taught or suggested by the references. [Felten, paragraph 48] Moreover, the statement that browsers can link to the external editor application explicitly teaches away from the claimed step of automatically invoking. This statement also teaches that there is no interaction between the external editor application and the image being displayed in the browser-controlled window

PART II. The teachings of the references would have to be fundamentally modified to meet the limitations recited in claims 1 and 6. Such modifications could only be implemented by impermissibly utilizing claims 1 and 6 as a roadmap to modify the teachings of the references.

As set forth in detail above, the cited references teach a system where Web page authors create Web pages and publish the pages on network servers to be viewed by end-users on client-workstations connected to the network. [Felten, paragraphs 26, 29] Raggett I and Raggett II disclose a minor modification to the known technique of embedding of static images that allows the browser to call a rendering application to render embedded foreign data into a static image and return the static image to the browser prior to the browser's rendering the hypermedia document for display to the user. [Felten, paragraphs 29, 31] Raggett I also suggests that a sophisticated browser could link to an external editor for creating or revising embedded data.

The reasoning of the Office Action changes the Mosaic and Berners-Lee paradigm of publishing Web pages on a server that are passively viewed on a client-workstation into the '906 paradigm of allowing in-place interaction with objects displayed in a browser-controlled window.

Such a change is only possible by using the teachings of the '906 patent as a roadmap to modify the teachings of the references.

The majority of "impermissible hindsight" case law concerns situations where bits and pieces of the prior art are patched together utilizing the claims as a roadmap. Here, the bits and pieces do not even exist but must be fundamentally changed before being pieced together.

Accordingly, the prior art *itself*, without utilizing the teachings of the '906 patent to modify the teachings of the prior art, does not suggest or make obvious claims 1 and 6. [Felten, paragraph 60, 61]

Further, it is required that the references be considered in their entireties, i.e., including those parts that would argue against obviousness. As set forth in detail above, the discussions in Raggett I at pages 12 and 13 concerning the ability to use Raggett's proposed EMBED tag instead of the *src* attribute following the FIG tag teach that Raggett's proposed EMBED tag must return a static, non-interactive image to be inserted into the document and displayed by the browser [Felten, paragraph 41]. An interpretation of Raggett's proposed EMBED tag described in Raggett I to provide interactivity with an object displayed in the browser-controlled window would <u>contradict</u> the teachings of the entire reference.

PART III. The secondary considerations of failure of others and professional approval further support the conclusion of non-obviousness.

a. Professional approval.

As set forth in the Doyle declaration, early demonstrations of the '906 technology were enthusiastically received by the scientific visualization community. In several cases, experts in the field, after viewing the technology, reacted by inviting Dr. Doyle to make presentations at highly regarded conferences. [Doyle]

Additionally the public release of the '906 technology resulted in one of the inventors, Dr. Doyle, being invited to submit a cover article to Dr. Dobbs Journal, a very well-known industry-leading publication with a world-wide readership. [Doyle]

Such favorable reactions of experts in the field to the invention upon its initial public appearance support the conclusion that claims 1 and 6 would not have been obvious.

b. Failure of others to follow Raggett I and Raggett II to implement the claimed technology.

As set forth in the Felten declaration, there is no evidence that Raggett's proposed EMBED tag described in Raggett I and Raggett II was ever implemented or was ever used by persons of skill in the art to implement the features recited in claims 1 and 6. [Felten, paragraphs 45,63]

The author of Raggett I and II, himself, never contemplated the functionality described in Claims 1 and 6. Mr. Raggett acknowledged this in sworn testimony at trial. As discussed above, Mr. Raggett was called as a witness by Microsoft Corporation in defense of a claim of infringement of claims 1 and 6:

> Question: The Netscape plug-ins had the ability to interact with an embedded program object in a Web page, right, sir?
> Answer: That is correct.

MS. CONLIN: I have no further questions.

FURTHER RECROSS EXAMINATION BY MR. BAUMGARTNER:

Question: And you envisioned that, didn't you?

Answer: I can't say I did. I think Pei Wei's and other people's ideas contributed to these discussions about HTML Plus back in '93. And as you can see, we talked about dynamic linked libraries in that paragraph and other -- or shared objects or shared libraries for UNIX. So I think all the ideas were present before. [Krueger declaration]

Mr. Raggett admits here that he did not envision the '906 functionality. Mr. Raggett's statement regarding dynamic libraries, shared objects, and shared libraries for Unix is another example of using impermissible hindsight to attempt to piece together bits and pieces of the prior art using the teachings of the '906 patent as a roadmap. There is no disclosure in Raggett I or Raggett II relating to modifying or combining the DLLs, shared objects, or shared libraries referred to in his testimony to perform the functions recited in claims 1 and 6 -- a fact conceded by Mr. Raggett.

It was recommended by the World Wide Web Talk Group that the EMBED tag proposed in Raggett I and Raggett II be abandoned for technical reasons. Further in the 1996 paper entitled "Inserting objects into HTML", edited by David Raggett and including Tim Berners-Lee as an author [Krueger], a new tag <OBJECT> is defined that allows an HTML author to specify the data to be inserted into HTML documents as well as the code that can be used to display/manipulate that data.

This paper also states that developers have been experimenting with new ideas for dealing with new media. The EMBED tag proposed in Raggett I and Raggett II is not mentioned as one of the new ideas. Mr. Raggett would clearly have been aware of the teachings of Raggett I and Raggett II since he authored those references only three years before the publication of "Inserting objects into HTML". Further, the editor and authors of the paper were the leading developers of Internet technology. Accordingly, the complete lack of any mention of Raggett's proposed EMBED tag in that paper is clear evidence that Raggett's proposed EMBED tag was never interpreted to provide the interactive features of claims 1 and 6 of the '906 patent.

Further, the paper states that at the time "Inserting objects into HTML" was written, 1996, the IMG tag was the only technique defined in the HTML specification for inserting images into Web pages. This statement also establishes that Raggett's proposed EMBED tag defined in Raggett I and II had been abandoned and did not suggest techniques for inserting new media into Web pages.

Accordingly, the failure of others, aware of the teachings of Raggett I and Raggett II, to implement the '906 technology as recited in claims 1 and 6 supports the conclusion that claims 1 and 6 would not have been obvious. [Felten, paragraph 64]

Dependent Claims

Claims 2 and 7 are rejected over the same combination of references as claim 1. The examiner states that claims 2 and 7 have the same scope and only claim 2 is discussed below.

Claim 2 depends on claim 1 and adds the additional limitation that the executable application is a controllable application. The controllable application is controlled on the client workstation via inter-process communication between the browser and the controllable application.

The examiner reasons that interactively controlling the controllable application via interprocess communication is obvious because Raggett II functions could be implemented as separate programs driven via pipes and stdin/stdout or as dynamically linked library modules.

This rejection is respectfully traversed for the following reasons. First, claim 2 depends on claim 1 and is allowable for the same reasons. Further, the separate programs driven via pipes and stdin/stdout or as dynamically linked library modules are rendering applications that provide no interactivity. The only communication between these applications are the standard calling and return actions described above. [Felten, 53-55]

Accordingly, there is no teaching in the cited references that would have made claims 2 or 7 obvious.

Claims 3 and 8 are rejected over the same combination of references as claim 1. The examiner states that claims 3 and 8 have the same scope and only claim 3 is discussed below.

Claim 3 depends on claims 1 and 2 and further recites the limitation that communications to interactively control the controllable application continue to be exchanged between the controllable application and the browser after the controllable application has been launched.

The examiner reasons that the use of an editor to create or revise an object suggests an continued interaction between the browser and external editor during the editing process and, therefore, the PHOSA would infer that communications continue to be exchanged between the browser and controllable application after the controllable application has been launched.

This rejection is respectfully traversed for the following reasons. First, claim 3 depends on claims 1 and 2 and is allowable for the same reasons. Further, there is no communication between the browser and the external editor during the external process because web pages were written in one format (HTML) and viewed in another format displayed in the browser. The external editor and browser would operate as separate applications opened in separate windows with no communication between the two applications. [Felten, 24]. Additionally, the cited references teach the display of a static, non-interactive image in the web page being displayed. [Felten, 38-43]

Accordingly, there is no teaching in the cited references that would have made claims 3 or 8 obvious.

Claims 4-5 and 9-10 are rejected under 35 U.S.C. §103(a) as being unpatentable over the combination of the admitted prior art in the '906 patent in view of Berners-Lee, Raggett I and Raggett II above, and further in view of Reichard and Cox. The examiner states that claims 4-5 and 9-10 have the same scope and only claims 4-5 are discussed below.

Claim 4 depends on claims 1, 2, and 3 and adds the additional limitations that instructions for controlling the controllable application reside on the network server and that commands can be issued from the client workstation that cause instructions to execute on the network server. Information is sent from the network server to the client workstation in response to the instructions executed on the client workstation and the information is processed on the client workstation to interactively control the controllable application.

Cox describes an object-oriented programming paradigm that makes software more tolerant to change.

Reichard states that object-oriented programming is finally coming to X. A new toolkit supports the model for distributed objects that can be distributed between processes on a single machine or across a network to many machines.

One goal of this effort is to enable users to create compound documents by embedding user interface components into windows. One example is a word processor object.

The examiner reasons that the controllable application, the external editor, executes on the same machine as the browser and that it would be obvious to implement the controllable application in X Windows. It is further stated that Reichard teaches an extension to X Windows allowing linking of and embedding of objects across a network. It is stated that it would have been obvious to implement the browser and controllable application (i.e., external editor) in combination of applicants' admitted prior art in view of Berners-Lee, Raggett I and II using Reichard's distributed object toolkit because of Cox's teaching that applying object oriented techniques to software makes the software more tolerant.

This rejection is respectfully traversed for the following reason. First, claim 4 depends on claim 1-3 and is therefore allowable for the same reasons. Second, a complete redesign of the browser, rendering application, and external editor is required to obtain the combination of claim 4. However, the only suggestion of such a radical recombination cited is the teaching of Cox that object oriented techniques make software more tolerant.

As described in detail above, the interactivity recited in claim 1 was a revolutionary development over the cited prior art. Claim 4 recites specific techniques for providing interactive control based on instructions executed on a network server. There is no teaching in any of the references suggesting the combination of claim 4. As described above, the references teach away from interactivity with an object displayed in a Web page. [Felten, 61]

Claim 5 depends on claims 1, 2, 3, and 4 and recites the additional limitations that additional instructions for controlling the controllable application reside on the client workstation. This claim is allowable for the same reasons described above.

Accordingly, there is no teaching in the cited references that would have made claims 4-5 or 9-10 obvious.

Non-applied References

The cited references that have not been applied against the claims have been reviewed. These references are less relevant to the claims than the applied references and all pending claims are deemed allowable thereover.

CONCLUSION

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

If the Examiner believes a telephone conference would expedite prosecution of this Application, please telephone the undersigned at (925) 944-3320.

Respectfully submitted,

Charles E. Krueger
Reg. No. 30,077

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

6548 U.S. PTO
05/11/04

Please type a plus sign (+) inside this box → +

PTO/SB/21 (08-00)
Approved for use through 10/31/2002. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	REX Control No. 90/006,831
Filing Date	10/30/2003
First Named Inventor	Michael D. Doyle
Group Art Unit	2151
Examiner Name	Andrew Caldwell
Attorney Docket Number	006-1-1

Total Number of Pages in This Submission: 194

ENCLOSURES *(check all that apply)*

- □ Fee Transmittal Form
  - □ Fee Attached
- ☒ Amendment / Response
  - □ After Final
  - □ Affidavits/declaration(s)
- □ Extension of Time Request
- □ Express Abandonment Request
- □ Information Disclosure Statement
- □ Certified Copy of Priority Document(s)
- □ Response to Missing Parts/ Incomplete Application
  - □ Response to Missing Parts under 37 CFR 1.52 or 1.53

- □ Assignment Papers *(for an Application)*
- □ Drawing(s)
- □ Licensing-related Papers
- □ Petition Routing Slip (PTO/SB/69) and Accompanying Petition
- □ Petition to Convert to a Provisional Application
- □ Power of Attorney, Revocation Change of Correspondence Address
- □ Terminal Disclaimer
- □ Request for Refund
- □ CD, Number of CD(s)

- □ After Allowance Communication to Group
- □ Appeal Communication to Board of Appeals and Interferences
- □ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
- □ Proprietary Information
- □ Status Letter
- ☒ Other Enclosure(s) *(please identify below)*:

1) DECLARATION OF EDWARD W. FELTEN

2) DECLARATION OF MICHAEL D. DOYLE

3) DECLARATION OF CHARLES E. KRUEGER

4) INTERVIEW SUMMARY 3/15-16/04

5) INTERVIEW SUMMARY 4/27/04

6) RETURN POSTCARD

Remarks: The Commissioner is authorized to charge any additional fees to Deposit Account 502267.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Charles E. Krueger Reg No. 30,077
Signature	Charles E. Krueger
Date	May 10, 2004

CERTIFICATE OF MAILING

DEPOSITED WITH FEDERAL EXPRESS 5/10/2004 FOR OVERNIGHT DELIVERY TO ANNETTE MASIELLO PATENT SERVICES, ARLINGTON, VA.; TO BE HAND-DELIVERED BY MASIELLO PATENT SERVICES TO THE CENTRAL REEXAMINATION UNIT (CRYSTAL PLAZA 3\4, ROOM 3D68, ARLINGTON, VA) ON 5/11/2004.

Typed or printed name	Sharon D. Krueger Sharon D. Krueger 5/10/04