

Exhibit D



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.
90/007,858	12/22/2005	5838906	6620-66570-01	4371

30080 7590 07/30/2007

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

EXAMINER

ART UNIT PAPER NUMBER

DATE MAILED: 07/30/2007

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



DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

Stephen A Wright
KLARQUIST SPARKMAN LLP
121 SW SALMON STREET, SUITE 1600
PORTLAND, OR 97204

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. 90/007,858.

PATENT NO. 5838906.

ART UNIT 3992.

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Ex Parte Reexamination Communication	Control No. 90/007,858	Patent Under Reexamination 5838906	
	Examiner Joseph R. Pokrzywa	Art Unit 3992	

A SHORTENED STATUTORY PERIOD FOR RESPONSE TO THIS ACTION IS SET TO EXPIRE 2 MONTH(S) FROM THE MAILING DATE OF THIS LETTER. EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c). If the specified period for response is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

cc: Requester (if third party requester)

Office Action in Ex Parte Reexamination	Control No. 90/007,858	Patent Under Reexamination 5838906	
	Examiner Joseph R. Pokrzywa	Art Unit 3992	

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

- a Responsive to the communication(s) filed on _____. 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 2 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. <input checked="" type="checkbox"/> Notice of References Cited by Examiner, PTO-892. | 3. <input type="checkbox"/> Interview Summary, PTO-474. |
| 2. <input checked="" type="checkbox"/> Information Disclosure Statement, PTO/SB/08. | 4. <input checked="" type="checkbox"/> <u>PTO-1449</u> . |

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 _____ 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)

Art Unit: 3992

DETAILED ACTION

Reexamination

1. Claims 1-10 of U.S. Patent Number 5,838,906 (hereafter "the '906 Patent") are subject to reexamination.
2. A previous reexamination certificate for the '906 Patent (in reexamination number 90/006,831) was issued June 6, 2006.

Information Disclosure Statement

3. The references listed in the Information Disclosure Statements submitted on 6/12/07, 5/1/07, 1/29/07, 1/23/07, 1/22/07, 1/10/07, 11/6/06, 9/18/06, and 8/24/06 have been considered by the examiner (see attached PTO/SB/08As and PTO-1449s).
4. The examiner notes that MPEP 2256, under the heading "Prior Art Patents and Printed Publications Reviewed by Examiner in Reexamination" states, in part:

Where patents, publications, and other such items of information are submitted by a party (patent owner or requester) in compliance with the requirements of the rules, **the requisite degree of consideration to be given to such information will be normally limited by the degree to which the party filing the information citation has explained the content and relevance of the information.** The initials of the examiner placed adjacent to the citations on the form PTO/SB/08A and 08B or its equivalent, without an indication to the contrary in the record, do not signify that the information has been considered by the examiner any further than to the extent noted above. [emphasis added]

Art Unit: 3992

Further, MPEP 609.05(b) states:

The information contained in information disclosure statements which comply with both the content requirements of 37 CFR 1.98 and the requirements, based on the time of filing the statement, of 37 CFR 1.97 will be considered by the examiner. Consideration by the examiner of the information submitted in an IDS means that **the examiner will consider the documents in the same manner as other documents in Office search files are considered by the examiner while conducting a search of the prior art in a proper field of search.** The initials of the examiner placed adjacent to the citations on the ** PTO/SB/08A and 08B or its equivalent mean that the information has been considered by the examiner to the extent noted above. [emphasis added]

With this, the examiner notes that with the large number of references submitted in the above noted PTO/SB/08As and PTO-1449s, the references were considered to at least the “degree to which the party filing the information citation has explained the content and relevance of the information”, and in “the same manner as other documents in Office search files are considered by the examiner while conducting a search of the prior art in a proper field of search”.

Discussion of Prior Art Cited in the Request for Reexamination

5. The Request for Reexamination filed 12/22/05 (as seen on page 24 of the Request) alleges that independent claims 1 and 6 are anticipated by “Janssen” (noted as Exhibit B), as “would have been understood in the context of the general knowledge in the art evidenced by Raggett II, Mosaic, and X Window System art”. Additionally, the Request alleges that claims 1 and 6 are obvious over the references of “The Mosaic Admitted Prior Art”, “Raggett II” (noted as Exhibit C), and the reference of “Janssen”. It is noted that the Janssen reference is a posted response to the post made in the reference of Raggett II, both being posts in the www-talk email list. However, assuming that the Janssen posting and the Raggett II posting were both

Art Unit: 3992

adequately indexed and available to qualify as prior art printed publications, a rejection of independent claims 1 and 6 cannot be made because of the reasons that follow.

6. The examiner also notes that the previous reexamination proceeding (reexamination number 90/006,831) stated that the claims of the '906 Patent were patentable over a combination of references which included the Mosaic Admitted Prior Art and the Raggett II references, as well as two other references. As noted in the previous reexamination proceeding in the Office paper dated 1/20/06, on page 3 the examiner at that time stated:

“...the cited four-way combination of the patent owner’s admitted prior art (APA), Berners-Lee, Raggett I, and Raggett II, ‘does not explicitly teach of a method that ‘enables interactive processing of said object.’ The combination teaches a method that embeds static objects, as opposed to dynamic objects, with distributed hypermedia documents.’”

Further, the examiner of the previous reexamination proceeding noted on page 5 of the same paper above that:

“...it is the browser application (i.e., not an executable application separate from the browser application that makes the active areas “interactive” by waiting for a user input, typically in the form of a mouse click.’”

Art Unit: 3992

7. Currently, in viewing the Janssen reference alone, the Third Party Requester relies on common knowledge that one of ordinary skill in the art would have, in order to anticipate the independent claims. However, in this regard, MPEP 2216 [R-5] states, in part:

Questions relating to grounds of rejection other than those based on prior art patents or printed publications should not be included in the request and will not be considered by the examiner if included.

Further, 37 CFR 1.552 states:

37 CFR 1.552. Scope of reexamination in ex parte reexamination proceedings.

a) Claims in an ex parte reexamination proceeding will be examined **on the basis of patents or printed publications** and, with respect to subject matter added or deleted in the reexamination proceeding, on the basis of the requirements of 35 U.S.C. 112. [Emphasis added].

With this, in the instant case, the “common knowledge that one of ordinary skill in the art would have”, as suggested by the Third Party Requester, is not a patent or printed publication. It is noted, however, that “common knowledge” or “common sense” can be utilized if backed up by a printed publication. Thus, a rejection, as proposed by the Third Party Requester, having independent claims 1 and 6 being anticipated by the Janssen reference, which relies on common knowledge with the statement made by Janssen, would not be proper, as a patent or printed publication is needed upon which a rejection would be based.

Art Unit: 3992

8. Continuing, MPEP 2143 states in part that:

To 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. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Further, MPEP 2143.03 states:

To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. In re Fine, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

9. With this, the examiner believes that the combination of the Mosaic Admitted Prior Art, the Raggett II reference, and the Janssen reference are not seen to teach all of the claim limitations of independent claims 1 and 6, being a requirement to establish a prima facie case of obviousness. Particularly, the Raggett II posting teaches that a distributed hypermedia document includes an embed text format [being the "EMBED TYPE =" function], located at a location in a distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document [whereby the <EMBED TYPE="text/eqn" causes the browser to identify the format of the embedded data from the "type" attribute; also see page 2 of Raggett II, wherein "you can also put the foreign data in a separate file referenced by a URL"]. However Raggett II is not seen to teach that the object has type information associated

Art Unit: 3992

with it utilized by the browser to identify and locate an executable application external to the first distributed hypermedia document.

10. Continuing, Janssen describes executing an interactive processing of an object within a display area created at a location within a portion of a distributed hypermedia document being displayed in a browser-controlled window [whereby Janssen states that the functions of separate programs discussed by Raggett can be handled by having "...the browser create and manage an X-sub-window over the area where the inset is to be displayed, and pass the window ID of the sub window to the subprogram which understands the inset format, with the understanding that the program is to handle all events and refresh on the sub-window, but the browser gets to handle configuration and window movement"].

11. However, the examiner can find no evidence within the references of Janssen and Raggett that specifically teach of "automatically invoking the executable application to execute on said client workstation in order to display said object". Further, with the general statement in Raggett II that states "foreign data in a separate file referenced by a URL", one could interpret the claimed "location of at least a portion an external object". However, when viewing the references of Raggett II and Janssen, in the examiner's opinion, one of ordinary skill in the art at the time of the '906 Patent would not be further inclined to conclude that the subprogram which understands an inset format, as mentioned by Janssen, would be an external executable application, that executes and displays the external foreign data referenced by a URL, as mentioned by Raggett II, whereby the subprogram of Janssen is automatically invoked upon

Art Unit: 3992

parsing the hypermedia document, so that the subprogram **would display the external foreign data** from the separate file and **enable interactive processing of the external foreign data** from the separate file. The specific teaching that the browser is automatically invoking the subprogram which understands the inset format is just not discussed by the references of Janssen or Raggett II.

12. Further, the Third Party Requester discusses helper programs, stating that the use of helper programs was common knowledge at the time of the invention. While the use of helper programs may have been known, the examiner is unable to specifically find within the prior art submitted by the Third Party Requester, the particularly teaching of utilizing the helper programs for the specific limitations as claimed in independent claims 1 and 6. Particularly the helper program prior art submitted by the Third Party Requester (which will further be discussed below) are not seen to teach of executing a browser application at said client workstation that parses a first distributed hypermedia document, **having an embed text format included in the hypermedia document** that specifies the location of at least a portion of an object **external to the first distributed hypermedia document**, and subsequently, to **automatically invoke an external helper application** to execute a processing to display **the external object** and enable interactive processing of **the external object**. Thus, while helper programs were known and utilized in the prior art, as evidenced by the Third Party Requester's submissions, in the examiner's opinion, the claimed method is not seen to be common knowledge based on these known helper programs. Therefore, the examiner notes that the combination of limitations, as worded in independent claims 1 and 6, are not particularly seen in the proposed combination of

Art Unit: 3992

references of the Mosaic Admitted Prior Art, Raggett II and Janssen, in light of the common knowledge of known helper programs discussed by the Third Party Requester.

13. Particularly, the Third Party Requester relies on the understanding of those familiar with X Window System programming for the teaching of background helper programs. For instance on page 12 of the Request, the Third Party Requester states that "...for data in the 'ps' format, the [Mosaic] browser's default was to launch the Ghostscript application." But, the examiner notes that within Exhibit D submitted by the Third Party Requester, the reference states "Resource postscript ViewerCommand is now "dump" by default. This is because ghostview (the previous default), as currently shipped, runs ghostscript in unsafe mode and a malicious PostScript file could cause damage to your filesystem as a result." Thus, this statement by the Third Party Requester appears to conflict with the prior art of Exhibit D, as the Ghostscript application was not desired to be a default because it "could cause damage to your filesystem as a result".

14. Additionally, the Third Party Requester submitted and discussed further references that explain the state of the art at the time of the invention. One group, however, being the NCSA Mosaic references from the Request for Reexamination (which include Exhibits D, E, F, G, and H), are not seen to teach the combination of limitations requiring executing a browser application at said client workstation that parses a first distributed hypermedia document, having an **embed text format included in the hypermedia document that specifies the location of at least a portion of an object external to the first distributed hypermedia document, and**

Art Unit: 3992

subsequently, to automatically invoke an external application to execute a processing to display the object.

15. While the NCSA Mosaic references clearly teach of external programs that are invoked upon parsing certain file formats (being the external viewers, described in the Request for Reexamination, pages 12-14), the language of claim 1 (and similarly claim 6) requires that a portion of the object to be displayed by the external application is also **external to the hypermedia document**. The MIME types in the references are not seen to be external to the hypermedia document, but rather part of the hypermedia document. Further, while a hypertext to a subsequent second hypermedia document can be interpreted as the embed text format, this subsequent second hypermedia document is not seen as being automatically invoked by the parsing of the first hypermedia document, but rather would be invoked by a user clicking on the hyperlink.

16. Similarly, the references regarding XV (Exhibits I, J, and K), being an interactive image display for the X Windows system, as well as the references regarding ghostscript (Exhibits L and O), the reference regarding Xlib (Exhibit N), and the reference regarding X Windows (Exhibits M) each teach of utilizing external applications, which enable interactive processing of objects. However, once again, the references are not seen to teach the specific combination of limitations requiring executing a browser application at said client workstation that parses a first distributed hypermedia document, having an **embed text format included in the hypermedia document that specifies the location of at least a portion of an object external to the first**

Art Unit: 3992

distributed hypermedia document, and subsequently, to automatically invoke an external application to execute a processing **to display the object**.

17. Thus, while the prior art submitted in the Request for Reexamination teach various aspects of the limitations of independent claims 1 and 6, the individual pieces still lack specific teachings when viewing the claims as a whole. In this regard, MPEP 2141.02 [R-5] states:

In determining the differences between the prior art and the claims, the question under 35U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. *Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530, 218 USPQ 871 (Fed. Cir. 1983); *Schenck v. Nortron Corp.*, 713 F.2d 782, 218 USPQ 698 (Fed. Cir. 1983) (Claims were directed to a vibratory testing machine (a hard-bearing wheel balancer) comprising a holding structure, a base structure, and a supporting means which form “a single integral and gaplessly continuous piece.” Nortron argued the invention is just making integral what had been made in four bolted pieces, improperly limiting the focus to a structural difference from the prior art and failing to consider the invention as a whole. The prior art perceived a need for mechanisms to dampen resonance, whereas the inventor eliminated the need for dampening via the one-piece gapless support structure. “Because that insight was contrary to the understandings and expectations of the art, the structure effectuating it would not have been obvious to those skilled in the art.” 713 F.2d at 785, 218 USPQ at 700 (citations omitted).).

Therefore, in the examiner’s opinion, the prior art noted by the Third Party Requester, as cited in the Request for Reexamination dated 12/22/05, does not establish a prima facie case of obviousness for independent claims 1 and 6, and subsequently, will not be utilized in this Office action for a rejection of independent claims 1 and 6.

Discussion of Related Prior Art Cited in the Previous Reexamination Proceeding

18. The previous reexamination proceeding (90/006,831) further noted that the claims of the '906 Patent were patentable over the submitted Viola DX37 source code. The examiner explained on page 51 of the previous reexamination proceedings in the Office paper dated 1/20/06, that:

“While Viola DX37 supports hypermedia and a type of interpreted script-based interactive processing, the Examiner can find no indication from a comprehensive text search of the Viola DX37 files that such interactivity results from the use of parsed embed text format that specifies the location of an object external to the hypermedia document, where the browser application uses type information associated with the object to identify and locate an external executable application, and where the parsing step results in the browser automatically invoking the executable application to display the object and enable interactive processing of the object within the same browser-controlled window, when the instant '906 patent claims 1 and 6 are properly accorded the broadest reasonable interpretation consistent with the specification.”

19. Thus, while the Viola DX37 source code files were not effective in expressly teaching each of the limitations of independent claims 1 and 6, as noted above in the previous reexamination proceedings, the examiner notes that a new reference regarding Viola, noted as “A Brief Overview of the VIOLA Engine, and its applications”, written by Pei Wei, pages TT 05441 – TT 05600, which include the “Viola in a Nutshell: the Viola World Wide Web Toolkit, being

Art Unit: 3992

included on the Information Disclosure Statement dated 8/24/06, can be interpreted as teaching each of the limitations. A full discussion of the reference follows below.

Claim Rejections - 35 USC § 102

20. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

21. **Claims 1-10** are rejected under 35 U.S.C. 102(e) as being anticipated by “A Brief Overview of the VIOLA Engine, and its applications”, written by Pei Wei, pages TT 05441 – TT 05600, which include the “Viola in a Nutshell: the Viola World Wide Web Toolkit”, from <http://scam.xcf.berkeley.edu/~wei/viola/book>, having the date August 16, 1994, as seen on page

Art Unit: 3992

05450, being included on the Information Disclosure Statement dated 8/24/06 as Citation Number "CL", being the Defendant's Trial Exhibit Number 95 (hereafter referred to as "Viola").

Regarding *claim 1*, Viola discloses a method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment [see page TT 05462, wherein the diagram shown and the description provides "that only the graph field is continuously updated, not the rest of the document", and that the monitor application embedded in the document "maintains a continuous network connection to a server to listen to a data stream."; also see page TT 05463, whereby the Chess Board is received from across the network via the address of a network server of "<http://berkeley.ore.com>..."];

executing, at said client workstation, a browser application, that parses a first distributed hypermedia document to identify text formats included in said distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by said text formats [see program code on pages TT 05549 through TT 05586, whereby the browser of Viola inherently parses the document to identify text formats included in said distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by said text formats];

utilizing said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client

Art Unit: 3992

workstation [see pages TT 05462 – TT 05465, particularly page TT 05463, whereby the Chess Board is received from across the network via the address of “http://berkely.ore.com...”; also see page TT 05508, wherein “an HTML document can embed highly interactive scripts/applets. And it allows those applets (components) to be embedded inside a document, not in a separate external window.”],

wherein said first distributed hypermedia document includes an embed text format [interpreted as the “<LINK REL = “ ___”, seen on page 05517 and 05522, being an embedded 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 [see page TT 05460, wherein “One can now write highly interactive applets (little applications) which can be embedded into an HTML document or the dynamic toolbar.”; also see page 05508, wherein “A scripting language that is accessible from the HTML, such that an HTML document can embed highly interactive scripts/applets. And it allows those applets (components) to be embedded inside of a document, not in a separate external window.”; also see page TT 05517, wherein “what if you want to embed that applet inside a document? You can do so with the <LINK> tag like this: ...

<P> And, the time now is:

<LINK REL=“viola” HREF http://ora.com/viola/book/objs/showTime.v>”],

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 [see page TT 05517, whereby an applet file is created, wherein “Save that file under

Art Unit: 3992

the name "showTime.v", and put it somewhere that a web browser can get to it (doesn't matter whether it's via file: or http:)", and

wherein said embed text format is parsed by said browser to automatically invoke said executable application to execute on said client workstation in order to display said object and 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 [see page TT 05462, wherein "only the graph field is continuously updated, not the rest of the document", and seen within the diagram below:



whereby the monitor application embedded in the document "maintains a continuous network connection to a server to listen to a data stream"; also see page TT 05475, wherein "This **auto loading scheme** makes it feasible to build a vast number of useful objects and file them away in places like /usr/local/viola/lib, to be **loaded and instantiated automatically only when needed.**"; also see pages TT 05516 and TT 05517, wherein "lets build a little time/display applet that continuously update itself, and then show how to embed that applet in a HTML document" and also "what if you want to embed that applet inside a document? You can do so with the <LINK> tag like this: ...

<P> And, the time now is:

Art Unit: 3992

<LINK REL="viola" HREF <http://ora.com/viola/book/objs/showTime.v>>"].

Regarding **claim 2**, Viola discloses the method discussed above in claim 1, and further teaches that said executable application is a controllable application and further comprising the step of:

interactively controlling said controllable application on said client workstation via inter-process communications between said browser and said controllable application [*see pages TT 05461-TT 05465, TT 05516, TT 05517, a well as pages TT 05521-05523, whereby the Viola WWW browser interactively utilizes the applet stored on the external web server*].

Regarding **claim 3**, Viola discloses the method discussed above in claim 2, and further teaches that 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 [*see pages TT 05461-TT 05465, TT 05516, TT 05517, and TT 05521-05523*].

Regarding **claim 4**, Viola discloses the method discussed above in claim 3, and further teaches that additional instructions for controlling said controllable application reside on said network server [*see page TT 05462, wherein the diagram shown and the description provides "that only the graph field is continuously updated, not the rest of the document", and that the monitor application embedded in the document "maintains a continuous network connection to a server to listen to a data stream."*]; also see pages TT 05516 and TT 05517, a well as pages TT

Art Unit: 3992

05521-05523, whereby the Viola WWW browser interactively utilizes the applet stored on the external web server],

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

issuing, from the client workstation, one or more commands to the network server [see page TT 05517, whereby the applet is requested by the workstation];

executing, on the network server, one or more instructions in response to said commands [being the file described on pages TT 05516 and TT 05517 or the files discussed on pages TT 05522 and TT 05523];

sending information from said network server to said client workstation in response to said executed instructions [see page TT 05517; also see the figure on page TT 05522, being the response of the applet]; and

processing said information at the client workstation to interactively control said controllable application [see page TT 05517; also see the figure on page TT 05522, whereby the data defined in the applet is displayed at the workstation].

Regarding **claim 5**, Viola discloses the method discussed above in claim 4, and further teaches that said additional instructions for controlling said controllable application reside on said client workstation [see page TT 05517, wherein "Save the file under the name "shoTime.v" and put it somewhere that a web browser can get to it (doesn't matter whether it's via file: or http:)"]].

Art Unit: 3992

Regarding *claim 6*, Viola discloses 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 [see page TT 05462, wherein the diagram shown and the description provides "that only the graph field is continuously updated, not the rest of the document", and that the monitor application embedded in the document "maintains a continuous network connection to a server to listen to a data stream."; also see page TT 05463, whereby the Chess Board is received from across the network via the address of a network server of <http://berkely.ore.com>..."], the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein [being inherent in a system that utilizes the Viola browser], said computer program product further comprising:

computer readable program code for causing said client workstation to execute a browser application to parse a first distributed hypermedia document to identify text formats included in said distributed hypermedia document and to respond to predetermined text formats to initiate processes specified by said text formats [see program code on pages TT 05549 through TT 05586, whereby the browser of Viola inherently parses the document to identify text formats included in said distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by said text formats];

computer readable program code for causing said client workstation to utilize said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein the portion of said first hypermedia

Art Unit: 3992

document is displayed within a first browser-controlled window on said client workstation [*see pages TT 05462 – TT 05465, particularly page TT 05463, whereby the Chess Board is received from across the network via the address of “http://berkely.ore.com...”*; also *see page TT 05508, wherein “an HTML document can embed highly interactive scripts/applets. And it allows those applets (components) to be embedded inside a document, not in a separate external window.”*],

wherein said first distributed hypermedia document includes an embed text format [*interpreted as the “<LINK REL = “ ___ ”, seen on page 05517 and 05522, being embedded 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 [*see page TT 05460, wherein “One can now write highly interactive applets (little applications) which can be embedded into an HTML document or the dynamic toolbar.”*; also *see page TT 05508, wherein “A scripting language that is accessible from the HTML, such that an HTML document can embed highly interactive scripts/applets. And it allows those applets (components) to be embedded inside of a document, not in a separate external window.”*; also *see page TT 05517, wherein “what if you want to embed that applet inside a document? You can do so with the <LINK> tag like this: ...*

<P> And, the time now is:

<LINK REL=“viola” HREF http://ora.com/viola/book/objs/showTime.v>”],

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 [*see page TT 05517, whereby an applet file is created, wherein “Save that file under*

Art Unit: 3992

the name "showTime.v", and put it somewhere that a web browser can get to it (doesn't matter whether it's via file: or http:)", and

wherein said embed text format is parsed by said browser to automatically invoke said executable application to execute on said client workstation in order to display said object and 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 [see page TT 05462, wherein "only the graph field is continuously updated, not the rest of the document", and seen within the diagram below:



whereby the monitor application embedded in the document "maintains a continuous network connection to a server to listen to a data stream"; also see page TT 05475, wherein "This **auto loading scheme** makes it feasible to build a vast number of useful objects and file them away in places like /usr/local/viola/lib, to be **loaded and instantiated automatically only when needed.**"; also see pages TT 05516 and TT 05517, wherein "lets build a little time/display applet that continuously update itself, and then show how to embed that applet in a HTML document" and also "what if you want to embed that applet inside a document? You can do so with the <LINK> tag like this: ...

<P> And, the time now is:

Art Unit: 3992

<LINK REL="viola" HREF <http://ora.com/viola/book/objs/showTime.v>>"].

Regarding **claim 7**, Viola discloses the computer program product discussed above in claim 6, and further teaches that said executable application is a controllable application and further comprising:

computer readable program code for causing said client workstation to interactively control said controllable application on said client workstation via inter-process communications between said browser and said controllable application [*see pages TT 05461-TT 05465, TT 05516, TT 05517, a well as pages TT 05521-05523, whereby the Viola WWW browser interactively utilizes the applet stored on the external web server*].

Regarding **claim 8**, Viola discloses the computer program product discussed above in claim 7, and further teaches that 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 [*see pages TT 05461-TT 05465, TT 05516, TT 05517, and TT 05521-05523*].

Regarding **claim 9**, Viola discloses the computer program product discussed above in claim 8, and further teaches that additional instructions for controlling said controllable application reside on said network server [*see page TT 05462, wherein the diagram shown and the description provides "that only the graph field is continuously updated, not the rest of the document", and that the monitor application embedded in the document "maintains a*

Art Unit: 3992

continuous network connection to a server to listen to a data stream.”; also see pages TT 05516 and TT 05517, as well as pages TT 05521-05523, whereby the Viola WWW browser interactively utilizes the applet stored on the external web server], wherein said step of interactively controlling said controllable application includes:

computer readable program code for causing said client workstation to issue, from the client workstation, one or more commands to the network server [*see page TT 05517, whereby the applet is requested by the workstation*];

computer readable program code for causing said network server to execute one or more instructions in response to said commands [*being the file described on pages TT 05516 and TT 05517 or the files discussed on pages TT 05522 and TT 05523*];

computer readable program code for causing said network sever to send information to said client workstation in response to said executed instructions [*see page TT 05517; also see the figure on page TT 05522, being the response to the applet*]; and

computer readable program code for causing said client workstation to process said information at the client workstation to interactively control said controllable application [*see page TT 05517; also see the figure on page TT 05522, whereby the data defined in the applet is displayed at the workstation*].

Regarding *claim 10*, Viola discloses the computer program product discussed above in claim 9, and further teaches that said additional instructions for controlling said controllable application reside on said client workstation [*see page TT 05517, wherein “Save the file under*

Art Unit: 3992

the name "shoTime.v" and put it somewhere that a web browser can get to it (doesn't matter whether it's via file: or http:)].

22. **Claims 1-3 and 6-8** are rejected under 35 U.S.C. 102(e) as being anticipated by Cohen *et al.* (U.S. Patent Number 5,367,621, hereafter "Cohen"), when viewed with "Introducing NCSA Mosaic", written by the Software Development Group, National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, December 1993, being the Defendant's Trial Exhibit Number 226 (hereafter referred to as "NCSA Mosaic").

23. The examiner notes that MPEP 2131.01 states in part:

Normally, only one reference should be used in making a rejection under 35 U.S.C. 102. However, a 35 U.S.C. 102 rejection over multiple references has been held to be proper when the extra references are cited to:

- (A) Prove the primary reference contains an "enabled disclosure; "
- (B) Explain the meaning of a term used in the primary reference; or
- (C) Show that a characteristic not disclosed in the reference is inherent.

Thus, in the instant rejection of independent *claims 1 and 6*, the secondary reference of NCSA Mosaic is being utilized to show that the BookManager READ product of the primary reference of Cohen can be considered as a "browser application", therein proving that the Cohen reference has an "enabled disclosure". A full discussion follows below.

Art Unit: 3992

Regarding *claim 1*, Cohen discloses a method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment [see col. 4, lines 20-55, wherein "The link tags described herein specify hypertext links which are created within on-line documents and between on-line documents....Hypertext links connect elements in one part of an on-line document to elements in another part of the same document or in a separate on-line document or in an external file or database."; also see col. 9, lines 27-30, wherein "The softcopy book file 100 can be downloaded from a host through the LAN interface 210 to the workstation 200"];

executing, at said client workstation, a browser application [whereby workstation 200 executes the BookManager READ program seen in Figs. 6-7c, which can be interpreted as a "browser application", as further discussed below], that parses a first distributed hypermedia document to identify text formats included in said distributed hypermedia document and for responding to predetermined text formats to initiate processing specified by said text formats [see col. 2, lines 10-26; also see Figs. 1a and 1b; also see col. 6, lines 7-64];

utilizing said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client workstation [see col. 10, lines 12-29, wherein "In step 410, the page of text from the softcopy book is displayed on the display 208."],

Art Unit: 3992

wherein said first distributed hypermedia document includes an embed text format [interpreted as the multimedia link description tags LDESC included within the document, see col. 5, lines 8-38, wherein "The BookMaster tags are improved upon, in accordance with the invention, to provide a new multimedia link description tag LDESC in the prologue of the document..."; also see col. 7, lines 22-30, wherein "The link tag :L and its matching end tag :eL enclose a word or phrase in the body of the document that the author wants to create a link from. The LID attribute refers to one or more LDESC document link tags."], 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 [see Fig. 4a, wherein the Internal Graphics Object 110, seen in Fig. 1e, is external to the Book Text with Tags 104, seen in Fig. 1b; also see col. 8, lines 16-26, wherein "the external video object 195 which would be stored in another storage medium separate from that for the softcopy book file 100"; also see col. 15, lines 31-43, wherein The resulting invention provides a generalized link from a reference point within an organized hierarchy of text in a softcopy on-line book, to an arbitrary multimedia object within the softcopy book, or alternately from external files or external data bases."],

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 [see Fig. 1a, whereby the external executable application is specified as "DATA = 'graph.exe\GOCA FORMAT C'", seen in Fig. 1a, wherein the graph.exe program is external to the hypermedia document], and

wherein said embed text format is parsed by said browser to automatically invoke said executable application to execute on said client workstation in order to display said object and

Art Unit: 3992

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 [see steps 410-426 in Fig. 6 and steps 562-578 in Fig. 7c; also see col. 10, lines 33-60, wherein "In step 412, the link tags are located in the softcopy book text. In particular, the link tags 164, 168, and 172 in the book text of Fig. 1b are located. Then in step 414, a determination is made as to whether any link tags have a link description with the AUTOLAUNCH parameter equaling "yes" in the corresponding link descriptor tag.... In step 416 of Fig. 6, if an AUTOLAUNCH parameter is equal to "yes", then the program gets the DATA string from the link description. Reference to Fig. 1a will show that the link description tag 150 has the DATA = 'graph.exe \GOCA Format C'.... With reference to the graphic object type link descriptor 150 of Fig. 1a, the string 'graph.exe \GOCA Format C' is output by the softcopy book READ program 400 to begin execution of the specified I/O handler program, namely graph.exe, whose flow diagram handler program is shown in Fig. 7c."; also see col. 13, lines 52-67, wherein "In step 566, the data string is parsed to identify if the graphic software support specified by the author in the link descriptor 150, is present in this workstation."].

However, Cohen does not expressly state that the application run by the workstation that parses the text data and invokes the object to enable interactive processing of the object within a display area, is specifically a **browser application**. Contrarily, Cohen does state in col. 1, lines 29-40 that "The BookManager READ product can then manage, search, and show the on-line books created by BookManager Build." It is noted that this is the same functionality as a browser application. Furthermore, NCSA Mosaic discloses a method for running an application

Art Unit: 3992

program in a computer network environment, whereby the method utilizes a browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client workstation [see Fig. 1; also see page 463, col. 1, 1st paragraph, wherein “As a distributed hypermedia browser designed for information discovery and retrieval, NCSA Mosaic provides a unified interface to the diverse protocols, data formats, and information archives used on the Internet.”] Further, as read on page 463, col. 1, 3rd paragraph, NCSA Mosaic states “The NCSA Mosaic interface is based on the idea of hypermedia, where electronic links known as hyperlinks are embedded in richly formatted documents that can include full-color images and sounds. **These documents are presented to users like the pages of an interactive, scrollable, online book.**” [Emphasis added.]

Thus, at the time of the invention, one of ordinary skill in the art would consider the BookManager READ program of Cohen as a “browser application”. The NCSA Mosaic browser application, which presents hypermedia documents “that can include full color images and sounds” to users in “pages like an interactive, scrollable, on-line book”, include the same basic functions as the BookManager READ product, which also presents full color images and sounds within hypermedia documents, and manages, searches, and shows on-line books to users. Therefore, the BookManager READ product, taught by Cohen, can be considered as an equivalent to a browser application, as shown by the NCSA Mosaic reference, thus obtaining the invention as specified in claim 1.

Regarding **claim 2**, Cohen discloses the method discussed above in claim 1, and further teaches that said executable application is a controllable application [*see the discussion throughout col. 11, line 30-col. 14, line 20, whereby the executable application, being the I/O handler programs are "controllable"*] and further comprising the step of:

interactively controlling said controllable application on said client workstation via inter-process communications between said browser and said controllable application [*see Figs. 7a-7c; also see the discussion throughout col. 11, line 30-col. 14, line 20, wherein the client computer interactively controls the handler programs, whereby alternate media can be utilized if the capabilities of the handler programs are not appropriate to handle the media; also see col. 15, lines 36-43*].

Regarding **claim 3**, Cohen discloses the method discussed above in claim 2, and further teaches that 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 [*see Figs. 6, and 7a-7c; also see the discussion throughout col. 11, line 30-col. 14, line 20, wherein the client computer interactively controls the handler programs, whereby alternate media can be utilized if the capabilities of the handler programs are not appropriate to handle the media; also see col. 15, lines 36-43*].

Regarding **claim 6**, Cohen discloses 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 [*see*

Art Unit: 3992

col. 4, lines 20-55, wherein "The link tags described herein specify hypertext links which are created within on-line documents and between on-line documents....Hypertext links connect elements in one part of an on-line document to elements in another part of the same document or in a separate on-line document or in an external file or database."; also see col. 9, lines 27-30, wherein "The softcopy book file 100 can be downloaded from a host through the LAN interface 210 to the workstation 200"], the computer program product comprising:

a computer usable medium having computer readable program code physically embodied therein [see col. 9, lines 6-13, wherein "Also stored in the memory 220 is the softcopy book READ program 400 of Fig. 6, the I/O handler programs 500, 530, and 560 of Figs. 7a, 7b, and 7c, respectively, and the drivers and operating system 590. The CPU 204 of Fig. 4, executes the instructions embodied in the program 400 and in the handler programs 500, 530, and 560, in accordance with the invention."], said computer program product further comprising:

computer readable program code for causing said client workstation to execute a browser application [whereby workstation 200 executes the BookManager READ program seen in Figs. 6-7c, which can be interpreted as a "browser application", as further discussed below] 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 [see col. 2, lines 10-26; also see Figs. 1a and 1b; also see col. 6, lines 7-64];

computer readable program code for causing said client workstation to utilize said browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein the portion of said first hypermedia

Art Unit: 3992

document is displayed within a first browser-controlled window on said client workstation [see col. 10, lines 12-29, wherein "In step 410, the page of text from the softcopy book is displayed on the display 208."],

wherein said first distributed hypermedia document includes an embed text format [interpreted as the multimedia link description tags LDESC included within the document, see col. 5, lines 8-38, wherein "The BookMaster tags are improved upon, in accordance with the invention, to provide a new multimedia link description tag LDESC in the prologue of the document..."; also see col. 7, lines 22-30, wherein "The link tag :L and its matching end tag :eL enclose a word or phrase in the body of the document that the author wants to create a link from. The LID attribute refers to one or more LDESC document link tags."], 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 [see Fig. 4a, wherein the Internal Graphics Object 110, seen in Fig. 1e, is external to the Book Text with Tags 104, seen in Fig. 1b; also see col. 8, lines 16-26, wherein "the external video object 195 which would be stored in another storage medium separate from that for the softcopy book file 100"; also see col. 15, lines 31-43, wherein The resulting invention provides a generalized link from a reference point within an organized hierarchy of text in a softcopy on-line book, to an arbitrary multimedia object within the softcopy book, or alternately from external files or external data bases."],

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 [see Fig. 1a, whereby the external executable application is specified as "DATA =

Art Unit: 3992

'graph.exe\GOCA FORMAT C'', seen in Fig. 1a, wherein the *graph.exe* program is external to the hypermedia document], and

wherein said embed text format is parsed by said browser to automatically invoke said executable application to execute on said client workstation in order to display said object and 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 [see steps 410-426 in Fig. 6 and steps 562-578 in Fig. 7c; also see col. 10, lines 33-60, wherein "In step 412, the link tags are located in the softcopy book text. In particular, the link tags 164, 168, and 172 in the book text of Fig. 1b are located. Then in step 414, a determination is made as to whether any link tags have a link description with the AUTOLAUNCH parameter equaling "yes" in the corresponding link descriptor tag.... In step 416 of Fig. 6, if an AUTOLAUNCH parameter is equal to "yes", then the program gets the DATA string from the link description. Reference to Fig. 1a will show that the link description tag 150 has the DATA = 'graph.exe \GOCA Format C'.... With reference to the graphic object type link descriptor 150 of Fig. 1a, the string 'graph.exe \GOCA Format C' is output by the softcopy book READ program 400 to begin execution of the specified I/O handler program, namely *graph.exe*, whose flow diagram handler program is shown in Fig. 7c."; also see col. 13, lines 52-67, wherein "In step 566, the data string is parsed to identify if the graphic software support specified by the author in the link descriptor 150, is present in this workstation."].

However, Cohen does not expressly state that the application run by the workstation that parses the text data and invokes the object to enable interactive processing of the object within a

Art Unit: 3992

display area, is specifically a **browser application**. Contrarily, Cohen does state in col. 1, lines 29-40 that “The BookManager READ product can then manage, search, and show the on-line books created by BookManager Build.” It is noted that this is the same functionality as a browser application. Furthermore, NCSA Mosaic discloses a method for running an application program in a computer network environment, whereby the method utilizes a browser to display, on said client workstation, at least a portion of a first hypermedia document received over said network from said server, wherein the portion of said first hypermedia document is displayed within a first browser-controlled window on said client workstation [*see Fig. 1; also see page 463, col. 1, 1st paragraph, wherein “As a distributed hypermedia browser designed for information discovery and retrieval, NCSA Mosaic provides a unified interface to the diverse protocols, data formats, and information archives used on the Internet.”*] Further, as read on page 463, col. 1, 3rd paragraph, NCSA Mosaic states “The NCSA Mosaic interface is based on the idea of hypermedia, where electronic links known as hyperlinks are embedded in richly formatted documents that can include full-color images and sounds. **These documents are presented to users like the pages of an interactive, scrollable, online book.**” [Emphasis added.]

Thus, at the time of the invention, one of ordinary skill in the art would consider the BookManager READ program of Cohen as a “browser application”. The NCSA Mosaic browser application, which presents hypermedia documents “that can include full color images and sounds” to users in “pages like an interactive, scrollable, on-line book”, include the same basic functions as the BookManager READ product, which also presents full color images and sounds

Art Unit: 3992

within hypermedia documents, and manages, searches, and shows on-line books to users.

Therefore, the BookManager READ product, taught by Cohen, can be considered as an equivalent to a browser application, as shown by the NCSA Mosaic reference, thus obtaining the invention as specified in claim 6.

Regarding *claim 7*, Cohen discloses the computer program product discussed above in claim 6, and further teaches that said executable application is a controllable application [*see the discussion throughout col. 11, line 30-col. 14, line 20, whereby the executable application, being the I/O handler programs are "controllable"*] and further comprising:

computer readable program code for causing said client workstation to interactively control said controllable application on said client workstation via inter-process communications between said browser and said controllable application [*see Figs. 7a-7c; also see the discussion throughout col. 11, line 30-col. 14, line 20, wherein the client computer interactively controls the handler programs, whereby alternate media can be utilized if the capabilities of the handler programs are not appropriate to handle the media; also see col. 15, lines 36-43*].

Regarding *claim 8*, Cohen discloses the computer program product discussed above in claim 7, and further teaches that 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 [*see Figs. 6, and 7a-7c; also see the discussion throughout col. 11, line 30-col. 14, line 20, wherein the client computer interactively*

Art Unit: 3992

controls the handler programs, whereby alternate media can be utilized if the capabilities of the handler programs are not appropriate to handle the media; also see col. 15, lines 36-43].

Conclusion

24. Extensions of time under 37 CFR 1.136(a) will not be permitted in these proceedings because the provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Additionally, 35 U.S.C. 305 requires that reexamination proceedings "will be conducted with special dispatch" (37 CFR 1.550(a)). Extension of time in *ex parte* reexamination proceedings are provided for in 37 CFR 1.550(c).

25. 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.

26. 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, after final rejection and 37 CFR 41.33 after appeal, which will be strictly enforced.

Art Unit: 3992

NOTICE RE PATENT OWNER'S CORRESPONDENCE ADDRESS

Effective May 16, 2007, 37 CFR 1.33(c) has been revised to provide that:

The patent owner's correspondence address for all communications in an *ex parte* reexamination or an *inter partes* reexamination is designated as the correspondence address of the patent.

Revisions and Technical Corrections Affecting Requirements for Ex Parte and Inter Partes Reexamination, 72 FR 18892 (April 16, 2007)(Final Rule)

The correspondence address for any pending reexamination proceeding not having the same correspondence address as that of the patent is, by way of this revision to 37 CFR 1.33(c), automatically changed to that of the patent file as of the effective date.

This change is effective for any reexamination proceeding which is pending before the Office as of May 16, 2007, including the present reexamination proceeding, and to any reexamination proceeding which is filed after that date.

Parties are to take this change into account when filing papers, and direct communications accordingly.

In the event the patent owner's correspondence address listed in the papers (record) for the present proceeding is different from the correspondence address of the patent, it is strongly encouraged that the patent owner affirmatively file a Notification of Change of Correspondence Address in the reexamination proceeding and/or the patent (depending on which address patent owner desires), to conform the address of the proceeding with that of the patent and to clarify the record as to which address should be used for correspondence.

Telephone Numbers for reexamination inquiries:

Reexamination and Amendment Practice	(571) 272-7703
Central Reexam Unit (CRU)	(571) 272-7705
Reexamination Facsimile Transmission No.	(571) 273-9900

Art Unit: 3992

27. All correspondence relating to this ex parte reexamination proceeding should be directed:

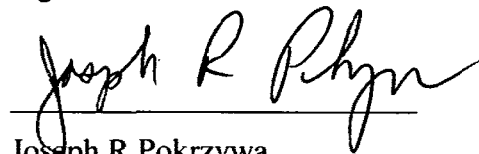
By Mail to: Mail Stop *Ex Parte* Reexam
Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to: (571) 273-9900
Central Reexamination Unit

By hand: Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication should be directed to the Central Reexamination Unit at telephone number 571-272-7705.

Signed:



Joseph R Pokrzywa

Primary Patent Examiner

Central Reexamination Unit 3992

(571) 272-7410

Conferees:

