

858 PH Ex. 1

DEC 22 2005
PATENT TRADEMARK OFFICE

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.

(Also referred to as FORM PTO-1465)

REQUEST FOR EX PARTE REEXAMINATION TRANSMITTAL FORM

Address to:
Mail Stop Ex Parte Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attorney Docket No.: 6620-66570-01
Date: December 22, 2005

64660 U.S. PTO
12/22/05

1. This is a request for *ex parte* reexamination pursuant to 37 CFR 1.510 of patent number 5,838,906 issued November 17, 1998. The request is made by:

patent owner. third party requester.

2. The name and address of the person requesting reexamination is:

Klarquist Sparkman, LLP
One World Trade Center
121 SW Salmon Street, Suite 1600
Portland, OR 97204

64660 U.S. PTO
90007858
12/22/05

3. a. A check in the amount of \$ _____ is enclosed to cover the reexamination fee, 37 CFR 1.20(c)(1);

b. The Director is hereby authorized to charge the fee as set forth in 37 CFR 1.20(c)(1) to Deposit Account No. 02-4550 (submit duplicative copy for fee processing); or

c. Payment by credit card. Form PTO-2038 is attached.

4. Any refund should be made by check or credit to Deposit Account No. _____ 37 CFR 1.26(c). If payment is made by credit card, refund must be to credit card account.

5. A copy of the patent to be reexamined having a double column format on one side of a separate paper is enclosed. 37 CFR 1.510(b)(4)

6. CD-ROM or CD-R in duplicate, Computer Program (Appendix) or large table
 Landscape Table on CD

7. Nucleotide and/or Amino Acid Sequence Submission
If applicable, items a. - c. are required.

a. Computer Readable Form (CRF)
b. Specification Sequence Listing on:

i. CD-ROM (2 copies) or CD-R (2 copies); or
ii. paper

c. Statements verifying identity of above copies

8. A copy of any disclaimer, certificate of correction or reexamination certificate issued in the patent is included.

9. Reexamination of claim(s) 1-10 is requested.

10. A copy of every patent or printed publication relied upon is submitted herewith including a listing thereof on Form PTO/SB/08, PTO-1449, or equivalent.

11. An English language translation of all necessary and pertinent non-English language patents and/or printed publications is included.

12/29/2005 MSAL/DANA 00000001 624550 90007858
01 FC:1812 2520.00 DA

(Page 1 of 2)

This collection of information is required by 37 CFR 1.510. 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 1.11 and 1.14. This collection is estimated to take 2 hours 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: Mail Stop Ex Parte Reexam, 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.

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.

12. The attached detailed request includes at least the following items:
- a. A statement identifying each substantial new question of patentability based on prior patents and printed publications. 37 CFR 1.510(b)(1)
 - b. An identification of every claim for which reexamination is requested, and a detailed explanation of the pertinency and manner of applying the cited art to every claim for which reexamination is requested. 37 CFR 1.510(b)(2)

13. A proposed amendment is included (only where the patent owner is the requester). 37 CFR 1.510(e)

14. a. It is certified that a copy of this request (if filed by other than the patent owner) has been served in its entirety on the patent owner as provided in 37 CFR 1.33(c).
The name and address of the party served and the date of service are:

The Regents of the University of California
 c/o: Law Office of Charles E. Krueger
 P.O. Box 5607
 Walnut Creek, CA 94596-1607

Date of Service: December 22, 2005; or

- b. A duplicate copy is enclosed since service on patent owner was not possible.

15. Correspondence Address: Direct all communication about the reexamination to:

The address associated with Customer Number:

OR

Firm or Individual Name Klarquist Sparkman, LLP

Address One World Trade Center
121 SW Salmon Street, Suite 1600

City Portland State OR Zip 97204

Country United States

Telephone 503-595-5300 Email stephen.wight@klarquist.com

16. The patent is currently the subject of the following concurrent proceeding(s):
- a. Copending reissue Application No. _____
 - b. Copending reexamination Control No. 90/006,831
 - c. Copending Interference No. _____
 - d. Copending litigation styled: Eolas Technologies, Inc. v. Microsoft,
No. 99-626, N.D. Ill.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Stephen A. Wight
Authorized Signature

12-22-2005
Date

Stephen A. Wight
Typed/Printed Name

37,759 For Patent Owner Requester
Registration No. For Third Party Requester

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Michael D. Doyle, David C. Martin, Cheong S. Ang
Patent No.: 5,838,906
Issued: November 17, 1998
Filed: October 17, 1994
Assignee: Regents of the University of California
For: DISTRIBUTED HYPERMEDIA METHOD FOR AUTOMATICALLY
INVOKING EXTERNAL APPLICATION PROVIDING INTER-
ACTION AND DISPLAY OF EMBEDDED OBJECTS WITHIN A
HYPERMEDIA DOCUMENT

REQUEST FOR *EX PARTE* REEXAMINATION

UNDER 35 U.S.C. § 302 AND 37 C.F.R. § 1.510

Mail Stop *Ex Parte* Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

December 22, 2005

Dear Sir:

Reexamination of U.S. Patent No. 5,838,906 ("the '906 patent"; copy attached in double column format as Exhibit A) is requested pursuant to 35 U.S.C. § 302 and 37 C.F.R. § 1.510 based on the prior art printed publications cited herein. Copies of the cited references are attached as Exhibits to this Request. Because the filing date of the '906 patent is prior to November 29, 1999, the statutory *inter partes* reexamination option is not available, and therefore *ex parte* reexamination is requested. (See MPEP § 2601.) The '906 patent is still in force. In accordance with 37 C.F.R. §§ 1.33(c) and 1.510(b)(5), this request is being served in its entirety on the attorney of record in the '906 patent.

I. INTRODUCTION

The patent owner has compromised the pending Director-instituted reexamination of the '906 patent (control no. 90/006,831, "the '831 reexamination") by failing to tell the Examiner about prior art in its possession that would have materially changed the prior art landscape in front of the Examiner. The patent owner has prior art that (1) fills in the gaps perceived by the Examiner in the art before him, (2) invalidates each of the claims of the patent, and (3) refutes contentions made by the patent owner and its expert to the Examiner. The present reexamination request places this previously unconsidered art before the Examiner for consideration, thereby presenting substantial new questions of patentability.

The Withheld Art Fills in Gaps Noted by the Examiner

The patent owner possessed but failed to tell the Examiner about a prior art reference (Exhibit B, hereinafter "Janssen," which was a public posting to the www-talk email list) that fills in the gaps the Examiner noted in Raggett II (Exhibit C, hereinafter "Raggett II"). The withheld Janssen prior art reference could hardly be more closely tied to Raggett II. The Janssen reference responded to and incorporated the Raggett II reference, quoting parts of it, and expanded upon it in ways that provide precisely what the Examiner deemed to be missing from Raggett II.

There is no doubt that the patent owner was aware of this Janssen reference that expands upon Raggett II. Exhibit B is a bates numbered version of the Janssen posting, which was provided to the patent owner long ago during its patent infringement action against Microsoft Corporation (No. 99-626, N.D. Ill.). Yet, there is no indication in the public record that the patent owner brought this key reference to the Examiner's attention.

This previously unconsidered Janssen posting is highly material because it fills each perceived gap identified by the Examiner in the Raggett II posting, and refutes positions the patent owner took with respect to the Raggett II posting. For example, the Examiner agreed with the patent owner that the art taught away from combining the Toye prior art patent with Raggett

II and the other cited art. The Janssen reference, however, completely fills this gap because it explicitly incorporates, responds to, and expands upon the Raggett II posting. In other words, the Janssen posting does not merely teach the combination with Raggett II that the Examiner found lacking; it effects and constitutes that combination.

The Withheld Prior Art Invalidates the '906 Patent Claims

In addition to filling in the gaps noted by the Examiner, the Janssen reference discloses each other limitation in the broader claims. When understood in the context of the software systems it expressly concerns (e.g., the Mosaic browser and X Window System), the Janssen reference anticipates claims 1 and 6 of the '906 patent. Further, combined with this related art and Raggett II, it establishes a prima facie case of obviousness of claims 1 and 6; and, combined with other related prior art, it renders obvious the remaining claims. As such, the Janssen posting raises substantial new questions of patentability.

The patent owner also withheld an important prior art publication co-authored by two of the patent applicants. Specifically, patent applicants Doyle and Ang (with others) co-authored a publication, dated before the critical date, describing a version of the "Visible Embryo Project" software shown in Figure 9 and discussed at length in the '906 patent. This withheld prior art publication describes the project's application software as being distributed between a client workstation and another computer connected over a network, with ongoing communication between the client workstation and the network computer as recited in at least claims 4, 5, 9, and 10 of the '906 patent. Despite being co-authored by two of the three patent applicants, being described in part in the '906 patent, and being published more than one year before the application filing date, the patent owner did not submit this prior art publication to the Patent Office.

To illustrate the importance of these withheld Janssen and Doyle-Ang references, the following features chart summarizes two approaches to mapping prior art to claims 1-10 of

the '906 patent. Detailed claim charts are included at the end of this request for *ex parte* reexamination.

Feature	Prior Art
browser (all claims)	Janssen OR Mosaic Admitted Prior Art ("APA") (as part of combination)
launching separate executable application (all claims)	Janssen OR Mosaic APA (as part of combination)
EMBED tag (all claims)	Janssen OR Raggett II (as part of combination)
automatic invocation of separate application (all claims)	Janssen OR Raggett II (as part of combination)
in-line display in browser of foreign-format data (all claims)	Janssen OR Raggett II (as part of combination)
interactive processing of foreign-format data (all claims)	Janssen OR Janssen (as part of combination)
interactive control of separate application via ongoing inter-process communication (claims 2-5 and 7-10)	X Window System art (as part of combination)
distributing separate application between client and server (claims 4, 5, 9, and 10)	Doyle-Ang reference about Visible Embryo Project software (as part of combination)

Table 1: Summary of Prior Art Mappings

The Withheld Prior Art Refutes Positions Taken by the Patent Owner

These previously unconsidered prior art references (Janssen and Doyle-Ang) relate to software for the X Window System, just like the Raggett II prior art reference and the embodiments described in the '906 patent. A person of ordinary skill in the art necessarily would have read and understood these prior art references in the context of software for the X Window System, including the Mosaic browser and other available applications for the X Window System. Therefore, to set the necessary technical background for a proper understanding of these X Window System/Mosaic prior art references, Requester submits herewith further prior art references that document the capabilities of the X Window System and of certain interactive Mosaic helper applications (also called external viewers or simply viewers) available for the X Window System in September 1993. Section IV provides citations to some

specific portions of this extensive documentation that are relevant to the claims of the '906 patent.

The patent owner gave the Examiner none of this necessary background information. Instead, it made assertions to the Examiner that could not have been made had it provided the documentation now submitted by the Requester.

For example, in the '831 reexamination, the patent owner and its expert purported to summarize the state of the art for Mosaic browser technology and Mosaic helper applications. (See, e.g., Response received October 12, 2004, pages 1, 5-7, 11-13, and 15; Response received May 11, 2004, pages 1, 4-9, 11-15 and 18; Declaration of Edward Felten signed October 6, 2004, ¶¶ 18-20; and Declaration of Edward Felten signed May 7, 2004, ¶¶ 8-11, 16, 26, 31, 33, 36-55, 60, and 61.) They urged that the Raggett materials describe a browser operating with non-interactive rendering applications, which would simply return a static image when invoked, and then terminate. They did not tell the Examiner, however, that several interactive helper applications (such as Xv, Xdvi, and Ghostview) were default helper applications for Mosaic in September 1993. These helper applications by default provide interactive, continuing processing of image data, as opposed to processing in which a viewer starts, accepts a file by standard input, and returns a static image, with no interactivity. The submitted interactive helper application documentation and other background X Window System documentation, which the patent owner and its agents failed to submit to the Patent Office, call into question the completeness and accuracy of representations of the patent owner and its agents.

In sum, by providing a more complete prior art landscape that the patent owner chose not to present, the present reexamination request fills in the perceived prior art gaps noted by the Examiner in the pending reexamination, and presents several substantial and compelling new questions of patentability.

II. IDENTIFICATION OF CLAIMS FOR WHICH REEXAMINATION IS REQUESTED

Reexamination of claims 1-10 of the '906 patent is requested. Claims 1 and 6 are the only independent claims. Claims 1-5 are method claims, and each of claims 2-5 depends consecutively from the preceding claim. Claims 6-10 are apparatus counterparts to claims 1-5, respectively. None of apparatus claims 6-10 is patentably distinct from its counterpart method claim, as admitted by the applicants during prosecution. (See Amendment mailed June 2, 1997, in the '906 patent file history, page 26 (the apparatus claims are "of the same scope as claims 1-5"); see also Office action mailed February 26, 2004, in the '831 reexamination, pages 7 and 9.)

III. PROCEDURAL BACKGROUND

The original application for the '906 patent was filed on October 17, 1994, as U.S. Patent Application Serial No. 08/324,443. The '906 patent is assigned to The Regents of the University of California ("The Regents") and is exclusively licensed to Eolas Technologies Incorporated ("Eolas"). Eolas and The Regents are pursuing a patent infringement action against Microsoft Corporation. (No. 99-626, N.D. Ill.) Certain aspects of this case have been considered by the Court of Appeals for the Federal Circuit. See *Eolas Techs. Inc. v. Microsoft Corp.*, 399 F.3d 1325 (Fed. Cir. 2005).

The Patent Office instituted the '831 reexamination on October 30, 2003. Initially, it rejected claims 1-3 and 6-8 of the '906 patent as being unpatentable over a combination of prior art including the Raggett II posting and Mosaic browser technology admitted to be prior art ("APA Mosaic"). The Patent Office added two other references to reject claims 4, 5, 9, and 10. The patent owner's admissions concerning prior art appear in the '906 patent itself as well as in various patent owner submissions to the Patent Office during the '831 reexamination.

The Patent Office considered at length the Raggett II reference in the '831 reexamination. Raggett II is a message entitled "HTML+ support for eqn & Postscript," authored by Dave Raggett and distributed to subscribers of the www-talk email list on June 14, 1993. The www-talk email list was a public, archived, and indexed discussion forum, whose contents were widely disseminated and publicly available on the Internet prior to the October 17, 1993, critical date of the '906 patent. The Raggett II posting thus qualifies as prior art against the '906 patent as a printed publication under 35 U.S.C. § 102(b). (M.P.E.P. § 2128.)

The Mosaic web browser application was itself (or "natively") able to display and process certain formats of data, but not all data formats. Raggett II describes techniques for supplementing a Mosaic browser in order to support data in a format that is "foreign" to the browser. It describes using an "EMBED" tag "to embed foreign-formats inline in the HTML+ source" (Ex. C, page 1), and gives the following example:

```
<H2>A example of an equation</H2>
<EMBED TYPE="text/eqn">zeta (s) ~~~ sum from k=1 to inf
k sup -s~~~ (Re s &gt; 1) </EMBED>
```

(*Id.*)

Raggett II describes the foreign-format data as being either internal or external to the HTML document. The above example shows the former, in which equation data is internally embedded within the EMBED tag. Or, "you can also put the foreign data in a separate file referenced by a URL." (Ex. C, pages 1 and 2.) Thus, Raggett II describes the EMBED tag in an HTML+ document including a URL that specifies the location of foreign data external to the HTML+ document.

Whether internal or external, the foreign-format data in Raggett II has type information associated with it. The Mosaic browser uses this type information to identify the format of the data. In the above quoted example set forth in Raggett II, the browser identifies the

format of the data from the “TYPE” attribute – “TYPE=text/eqn”. (Ex. C, page 1.) In this example, the type attribute is “specified as a MIME content type.” (*Id.*)

Once the foreign format is identified using the associated type information, a program needs to be identified that is capable of processing data in that format. Raggett II describes several techniques for doing this. The one of most interest here is to automatically invoke a separate program (i.e., a program other than the browser) for processing the data, which separate program is associated with the foreign-format data’s type information. As a specific example of this technique, Raggett II describes a separate application “that take[s] a sequence of bytes and return[s] a pixmap.” (*Id.*) Raggett II provides a general framework for calling such a separate program that can process the foreign-format data (“API for rendering foreign formats”) and elaborates:

Browsers can then be upgraded to display new formats without changing their code at all. All you would need is a way of binding the MIME content type to the function name for that format, e.g., via X resources or a config file. The functions could be implemented as separate programs driven via pipes and stdin/stdout or as dynamically linked library modules (Windows DLLs).

(*Id.*)

In the ‘831 reexamination, the Examiner agreed with the patent owner that Raggett II was lacking in an important respect. Specifically, the Examiner found that Raggett II was limited to invoking separate programs that translated the foreign-format data once, returned to the browser a static image, and then terminated. In other words, the Examiner determined that Raggett II did not disclose or suggest the separate program continuing to run to allow interactive processing of the foreign-format data.

What Eolas did not tell the Examiner, however, is that interactive Mosaic “helper” applications were common in the prior art. As described below, these undisclosed interactive programs, separate from the Mosaic browser, were designed to continue running and enable interactive processing and display of data. Indeed, the Janssen expansion of Raggett II describes

a technique for invoking such interactive programs. Thus, the patent owner's characterization of the separate programs available to the Mosaic browser, as contemplated in Raggett II, was at best incomplete and at worst misleading.

Without the benefit of a proper prior art landscape, on September 28, 2005, the PTO mailed a Notice of Intent to Issue Ex Parte Reexamination Certificate confirming the patentability of the '906 patent claims.

IV. SUMMARY OF THE PRIOR ART

The prior art describes what the '906 patent claims. For example, it describes what the '831 reexamination Examiner, provided with an incomplete and misleading description of the prior art, found lacking in Raggett II. Persons of skill in the art, as Mr. Janssen's posting demonstrates, knew of the availability of interactive Mosaic helper applications and therefore understood Raggett II quite differently than how the patent owner presented the prior art to the Patent Office Examiner. Below is a summary of previously unconsidered prior art demonstrating that the '906 patent claims what was already known in the art.

A. The Janssen Posting

On June 14, 1993, Bill Janssen and Dave Raggett held a public conversation over the Internet about techniques for doing what the '906 patent later claimed.

Within hours of its posting, Mr. Janssen publicly responded to Raggett II with a message he distributed to subscribers of the www-talk email distribution. For the same reasons that the Raggett II posting qualifies as prior art, the Janssen posting qualifies as prior art under 35 U.S.C. § 102(b).¹

¹ Both postings were made to the WWW-Talk email list, which is a public, archived and indexed discussion forum whose participants included those who were developing and standardizing Internet technologies at the time (e.g., Marc Andreessen, who led development of the Mosaic browser). The postings were widely disseminated and publicly available through the Internet and through other means at least from June 14, 1993, and they continue to be available on-line at

The Janssen posting (“Janssen”) is “In reply to” the message “Dave_Raggett: ‘HTML+ support for eqn & Postscript.’,” which is the Raggett II posting. (Ex. B.) Janssen quotes Raggett II:

- > *Browsers can then be upgraded to display new formats without changing their*
- > *code at all. All you would need is a way of binding the MIME content type*
- > *to the function name for that format, e.g. via X resources or a config file.*
- > *The functions could be implemented as separate programs driven via pipes and*
- > *stdin/stdout or as dynamically linked library modules (Windows DLLs).*

(*Id.*)

Janssen first affirms the quoted content of Raggett II – “Yes, that sounds good.”

Then Janssen expands upon Raggett II:

My favorite way to handle this is to have 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 that program is to handle all events and refresh on the sub-window, but the browser gets to handle configuration and window movement.

(*Id.*)

“My favorite way to handle this”: With this language, Janssen acknowledges the functionality described in the Raggett II posting and introduces an alternative mechanism for implementing such functionality.

<http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q2.index.html> and elsewhere. As such, they constitute “printed publications” within the meaning of 35 U.S.C. § 102(b) because each was a “contribution” to “electronic bulletin boards, message systems, and discussion lists” that were “accessible to persons concerned with the art to which the document relates” when they were posted to the WWW-Talk list. (See M.P.E.P. § 2128, which 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.”) The Raggett II and Janssen postings enjoy prior art effect from the date of their posting (i.e., June 14, 1993). (See M.P.E.P. § 2128, which 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.”)

“is to have the browser create and manage an X sub-window over the area where the inset is to be displayed”: With this language, Janssen describes the browser creating a new window (“X sub-window”) for display of the foreign-format (“inset format”) data. With this language, Janssen also describes the browser placing the sub-window inline with the rest of the data being displayed by the browser (“over the area where the inset is to be displayed”); i.e., the sub-window is placed at a location “inset” in the browser-controlled window.

“and pass the window ID of the sub-window to the subprogram which understands the inset format”: With this language, Janssen describes how the browser hands over information to a separate program (“the sub-program”) that is responsible for processing data in the foreign format. Specifically, the browser provides this separate program with a resource identifier (“window ID”) that the sub-program uses to identify the sub-window for two purposes, which are described next.

“with the understanding that that program is to handle all events and refresh on the sub-window”: With this language, Janssen reinforces that the separate program handles everything within the sub-window relating to interactive processing and display of the foreign-format data. Specifically, the separate program handles all keystrokes, all mouse button presses, all messages that the sub-window has become exposed, and all other events on the sub-window (“handles all events”). In other words, the separate program is not one that simply performs a translation and then terminates and disappears. On the contrary, Janssen describes this sub-program as one that takes over from the browser all event-handling throughout the life of the sub-window. Aside from event handling, the sub-program uses the window ID to identify the sub-window when drawing or redrawing (“handle all ... refresh”) to the display.

“but the browser gets to handle configuration and window movement”: With this language, Janssen further reinforces that the browser and separate program are operating side by side, with the browser handling two functions in relation to the sub-window. The browser handles the movement and the configuration of the sub-window within the browser-controlled

window, leaving foreign-format data processing and refresh operations on the sub-window to the separate program. Thus, when the browser changes the position, size, or border of the sub-window (“configuration and window movement”), the sub-window retains its place in the Web page display.

The above reading of the plain language of Janssen is confirmed by the state of the art in mid 1993. In particular, it is confirmed by the availability of interactive helper applications for Mosaic in the X Window System environment (e.g., Xv and Ghostview) as described below. In other words, the X Window System documentation and interactive helper application documentation submitted with this Request, and not previously considered by the Examiner, demonstrate conventional ways of doing what Janssen describes. As this patent owner has said, the hypothetical person of ordinary skill in the art “does things in a conventional way.” (Response received October 12, 2004, page 25.)

B. Interactive Mosaic Helper Applications / External Viewers

In September 1993, the Mosaic browser running in the X Window System environment was designed to deal with certain foreign-format data by launching a pre-designated helper application associated with the particular foreign format encountered by the browser. For example, for data in the ‘ps’ format, the browser’s default was to launch the Ghostscript application. This functionality is described in detail below.

Many of these helper applications were interactive. Once launched, they stayed running and handled the user’s interactions with the particular data type in which they specialized. This functionality is detailed below for three different prior art interactive helper applications.

Mosaic’s Launching of Helper Application Based On Particular Data Type Encountered

As noted, prior art versions of the Mosaic browser for the X Window System automatically associated particular helper applications with foreign-format data types by default.

Prior art Exhibits D – H documented this functionality for two prior art versions of Mosaic – version 1.2 and version 2.0 prerelease 4, as described below.

Mosaic 1.2: Source code and documentation for Mosaic 1.2 for the X Window System are prior art because they were available for download by anonymous file transfer protocol (“FTP”) download at least as early as June 1993. (Ex. D.) Exhibit E shows the contents of a compressed archive file entitled, “xmosaic-1.2.tar.z.” The compressed archive file includes a file entitled, “xresources.h,” which was used to provide certain settings for XMosaic 1.2. (Ex. E.) Among other things, the xresources.h file associated external viewers with the following types. (Ex. E, page 3 of the file xresources.h.)

Data Type	External Viewer
gif	Xv
jpeg	Xv
tiff	Xv
dvi	Xdvi
mpeg	mpeg_play

Table 2: Viewers Launched for Data Types in Mosaic 1.2

Mosaic 2.0, prerelease 4: Source code and documentation for Mosaic 2.0 prerelease 4 were available for download by anonymous FTP download in September 1993. (Ex. F, page 1.) In Mosaic 2.0 prerelease 4, the mechanism for associating data types with external viewers was changed to one in which MIME types map to viewers. (Ex. F, pages 1-2.) The mappings for file extensions, MIME types, and viewers were customizable. (*Id.*) Viewers were not just for images. Users could associate MIME types for other kinds of content (e.g., video animations, scientific data) with applications appropriate for the content. By default, however, certain file extensions were mapped to MIME types, as shown in the following table. (Ex. G.)

File Extension	MIME Type
.ps	application/postscript
.gif	image/gif
.tif or .tiff	image/x-tiff

.jpg or .jpeg	image/jpeg
.rgb	image/x-rgb
.mpg or .mpeg	video/mpeg
.dvi	application/x-dvi

Table 3: Default File Extension to MIME Type Mappings in Mosaic 2.0 Prerelease 4

Exhibit H shows default MIME types in Mosaic 2.0 prerelease 4 along with mappings to viewers, including the following mappings. (Ex. H.)

MIME Type	External Viewer
image/*	Xv
video/mpeg	Mpeg_play
application/postscript	Ghostview
application/x-dvi	Xdvi

Table 4: Default MIME Type to Viewer Mappings in Mosaic 2.0 Prerelease 4

In sum, Exhibits D – H establish that prior art Mosaic for X Window System browsers invoked whatever helper application was assigned to the particular foreign format the browser encountered while parsing an HTML document. As shown above, these helper applications included Xv and Ghostview, each of which was interactive as described below. (Xdvi was also interactive but is not addressed at this time.)

1. Xv: “Interactive Image Display for the X Window System.”

Xv is a prior art application that was invoked by Mosaic for interactive manipulation of GIF, JPEG, TIFF, and X11 format images. Requester submits prior art documentation for two versions of this program. (Exs. I, J, and K.)

The software Xv version 2.2 is described in the Xv 2.2 manual entitled, “Xvdocs.ps.x.” (Ex. I.) The Xv 2.2 manual is dated April 24, 1992, and included in the archive “Xv-2.21.tar.” In addition to the Xv 2.2 manual, Exhibit I includes printouts showing the contents of “Xv-2.21.tar.”

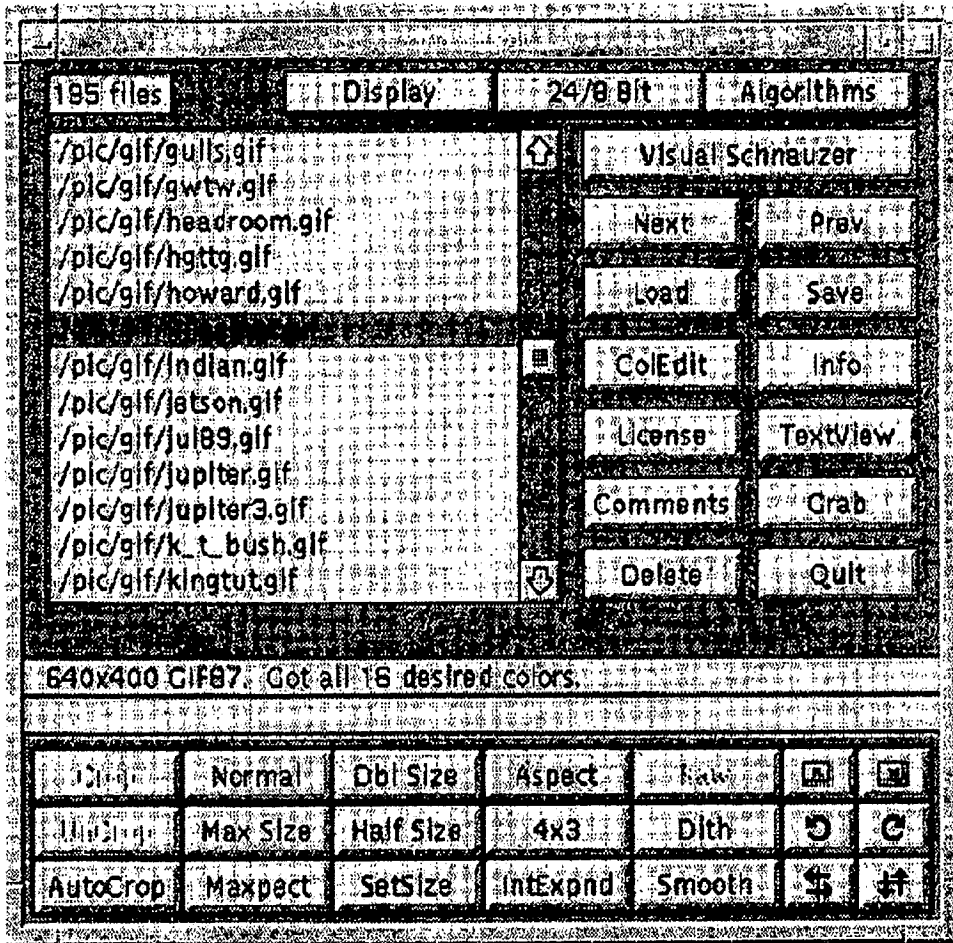
Xv version 3.0 is described in the Xv 3.0 manual entitled, “Xvdocs.ps.x.” (Ex. J.) The Xv 3.0 manual is dated April 26, 1993, and included in the archive “Xv-3.00.tar.” In

addition to the Xv 3.0 manual, Exhibit J includes printouts showing the contents of “Xv-3.00.tar.” Exhibit K is a www-talk posting describing the availability of Xv 3.0 in April 1993.

Xv versions 2.2x and 3.0 were publicly distributed before October 17, 1993, and their distributions included the Xv 2.2 and 3.0 manuals. The Xv 2.2 manual and Xv 3.0 manual qualify as prior art against the ‘906 patent as printed publications under 35 U.S.C. § 102(b). They also provide evidence of how the Janssen and Raggett II postings would have been understood prior to October 17, 1993.

According to the title pages for the manuals, Xv 2.2 and 3.0 provided “Interactive Image Display for the X Window System.” (Ex. I, title page of the Xv 2.2 manual; Ex. J, title page of the Xv 3.0 manual.) “Xv is an interactive image manipulation program for the X Window System.” (Ex. I, page 2 of the Xv 2.2 manual; Ex. J, page 3 of the Xv 3.0 manual.) The manuals describe Xv operating on images in numerous formats, including GIF, JPEG, TIFF, and X11 bitmap formats. (*Id.*) For an image displayed in a window on the screen, Xv versions 2.2 and 3.0 allowed a user to stretch, rotate, flip, or crop the image, magnify a portion of the image, or adjust colors of the image. (*Id.*) In other words, Xv versions 2.2 and 3.0 allowed ongoing real-time manipulation and control of the image, resulting in changes to the displayed image.

The Xv 2.2 and 3.0 manuals describe a user interacting with the Xv software to change what was displayed using the Xv controls window. The Xv controls window “contains controls to resize the current image, flip and rotate it, load and save different files, and bring up the other Xv windows.” (Ex. J, page 8 of the Xv 3.0 manual; *see also* Ex. I, page 6 of the Xv 2.2 manual.) The following figure shows an Xv controls window for Xv 3.0.



(Id.) Commands for resizing, rotating/flipping, smoothing, or cropping are “executed by either clicking the appropriate command button, or typing the keyboard equivalent (where given) into any open Xv window.” (Id.) Thus, Xv 2.2 and 3.0 accepted keyboard input into the Xv controls window *and* the Xv window in which the image was displayed.

Xv 2.2 and 3.0 accepted various command line options. (Ex. I, pages 38-46 of the Xv 2.2 manual; Ex. J, pages 57-71 of the Xv 3.0 manual.) For Xv 3.0, the “-viewonly” command line option is described as “For use when calling Xv from some other program. Forces all user input to be ignored.” (Ex. J, page 69.) The “-viewonly” option was not offered in Xv 2.2. In Xv 3.0, it was not the default setting. In any case, if Xv 3.0 were to handle all events in an Xv window, the calling program would not use the “-viewonly” option.

The “-” command line option is described as follows: “[s]pecifying ‘-’ all by itself tells Xv to take its input from stdin, rather than from a file. This lets you put Xv on the end of a Unix pipe.” (Ex. I, page 46; Ex. J, page 71.) Whether accepting input from a file or from stdin, however, Xv 2.2 and 3.0 by default allowed interactive manipulation of the image.

Thus, the Xv 2.2 and 3.0 manuals describe Xv accepting an image file as input and drawing/redrawing an image. Xv was interactive. The Xv 2.2 and 3.0 manuals describe ongoing interactive processing of the image data as keystrokes and button inputs were handled by the Xv software.

2. Ghostview: Previewer for Postscript Files

Ghostview is a prior art application that was invoked by Mosaic for interactive display and manipulation of postscript files. Requester submits prior art documentation for this prior art helper application. (Ex. L.)

The manual in the file entitled “Ghostview.ps” (Ex. L) describes Ghostview version 1.5. The Ghostview manual is included in the archive “Ghostview-1.5.tar.gz,” which indicates Ghostview.ps was last modified on July 25, 1993. Ghostview 1.5 was publicly distributed before October 17, 1993, and its distributions included the Ghostview manual. (See Ex. L, page 1.) The Ghostview manual qualifies as prior art against the ‘906 patent as a printed publication under 35 U.S.C. § 102(b); it also provides evidence of how the Janssen and Raggett II postings would have been understood prior to October 17, 1993. Exhibit L includes the Ghostview manual as well as printouts showing a directory listing of an FTP site and the contents of “Ghostview-1.5.tar.gz.” Exhibit L also includes a printout of the “README” file from “Ghostview-1.5.tar.gz.” The README file is dated July 25, 1993, and indicates:

Ghostview-1.5 is available via anonymous ftp from:
prep.ai.mit.edu:/pub/gnu/ghostview-1.5.tar.gz
ftp.cs.wisc.edu:/pub/X/ghostview-1.5.tar.gz

(Ex. L, page 19, which is page 1 of the Ghostview 1.5 README file.)

Ghostview 1.5 provided an X Window System interface for Ghostscript, an interpreter program for postscript files. (Ex. L, page 1 of the Ghostview 1.5 manual.) The Ghostview manual describes Ghostview 1.5 and Ghostscript functioning as two cooperating programs, with Ghostview creating a viewing window and Ghostscript drawing in it. (*Id.*)

The Ghostview manual describes Ghostview 1.5 as having a main viewport (for displaying an image) and a menu box with buttons for bringing up menus. (Ex. L, pages 1-2 of the Ghostview manual.) The menus included a “page” menu for changing pages, a “magstep” menu for changing view magnification, and an “orientation menu” for changing orientation. (*Id.*) Most of the popup menu commands could be invoked from the keyboard, and other keyboard input caused scrolling of the main viewport window up, down, left, or right. (Ex. L, pages 4-5 of the Ghostview manual.) Clicking a mouse button anywhere within the viewport window caused a zoom window to pop up. (Ex. L, page 1 of the Ghostview manual.)

The Ghostview manual describes Ghostview 1.5 accepting a postscript file as input and causing drawing/redrawing of an image (in cooperation with Ghostscript). It also describes, however, Ghostview 1.5 handling keystrokes and button inputs for ongoing interactive processing of the image.

Ghostview 1.5 was invoked with the name of the file to be previewed as a parameter. (*Id.*) Alternatively, Ghostview read from stdin if the filename was “-.” (*Id.*) Either way, Ghostview 1.5 provided tools for interactive manipulation of the image.

Ghostview 1.5 *passed the window ID* of a window to Ghostscript, for Ghostscript to draw an image to the window. The window ID passing mechanism is detailed below in the section about communication between X Clients in the X Window System.

C. X Window System

The Janssen posting must be understood in context. It relates to software for the X Window System. For this reason, Requester also submits sections of two prior art books about

the X Window System. (Exs. M and N.) Exhibit M includes chapters from Douglas Young, The X Window System, Programming and Applications with Xt (1990). Exhibit N includes chapters from Adrian Nye, Xlib Programming Manual for Version 11 (1990). Each of these books qualifies as prior art against the '906 patent as a printed publication under 35 U.S.C. § 102(b).

1. Overall Architecture

The Janssen posting uses the terms “X sub-window” and “window ID of the sub-window.” These terms of art must be understood in the context of the X Window System.

X Server software runs on a machine with a display, a keyboard, and a mouse. (Ex. M, 1.1 and 1.3; Ex. N, 1.2.2.) Programs called X Clients interact with the X Server. (*Id.*) Mosaic and Mosaic helper applications for the X Window System are X Clients.

The X Server stores data on behalf of X Clients and often shares the data between the X Clients. (*Id.*) The X Server provides identifying information for resources (e.g., window IDs for windows), and multiple X Clients can access the stored data with the identifying information. (*Id.*)

“Windows” in the X Window System are hierarchically organized according to parent-child relationships. (Ex. M, 1.5; Ex. N, 2.1.3, 2.2, and 2.4.) The root window of the hierarchy covers a whole display and has one or more child windows. (*Id.*) A child window is also called a sub-window. (*Id.*) Each window is identified by a window ID. (*Id.*)

As for communication, an X Client sends “requests” to the X Server (e.g., to request information from the X Server or request that the X Server draw something). (Ex. M, 1.4; Ex. N, 1.3.) The X Server may send “replies” in response. The X Server also sends “events” to X Clients. (Ex. M, 1.6; Ex. N, 1.3.)

2. Events

The Janssen posting uses the language “handles all events and refresh on the sub-window.” This language must be understood in the context of “events” in the X Window System.

The X Server sends “events” to an X Client when the X Server receives keystroke or mouse input directed to a window of the X Client. (Ex. M, 5.1 and 5.3; Ex. N, 2.5.1.) Other events relate to changes to the display or messages used for inter-client communication. (*Id.*) A third category of events are change notifications received by an X Client, for example, when the position, size, or border of a window of the X Client has changed. (Ex. M, 5.3.7.)

Typically, the X Server directs user input events to the window having user input focus. (Ex. M, 5.1; Ex. N, 2.5.2.) An X Client can change which windows receive which events through various mechanisms, and an X Client can change the windows for which it receives events. (Ex. M, 5.2; Ex. N, 2.5.2.) For example, for a particular window, an X Client can select which types of events are to be delivered to the X Client and which are to be ignored. (*Id.*)

For most event types, a particular event is sent to more than one X Client if each of the X Clients has selected the appropriate event type on the window in question. (*Id.*) Each of the X Clients has its own “event mask” for such a window. (*Id.*) If it has the window ID of the window, an X Client selects the event types for events it should receive and calls the function XSelectInput(), passing the window ID and the event mask. (*Id.*) The X Server routes events to interested X Client(s) according to the event mask(s). (*Id.*)

3. Communication Between X Clients

The Janssen posting describes interactions between the browser and subprogram (“pass the window ID” and “the understanding that the program is to handle ... but the browser gets to handle ...”). This language must be understood in the context of communication in the X Window System.

In the X Window System, mechanisms for inter-client communication include environment variables and properties, copy and paste using selections, buffering of data with X Server, command line options, and events. (Ex. M, 11.)

An X Client can use an environment variable to pass data to another X Client. For example, Exhibit O shows a file entitled “gs.interface” from the archive file “Ghostview-

1.5.tar.gz” for Ghostview 1.5 (see Ex. L). The “gs.interface file” describes Ghostview using an environment variable to *pass the window ID* of a window to Ghostscript. (Ex. O.)

When the GHOSTVIEW environment variable is set, ghostscript draws on an existing drawable rather than creating its own window. Ghostscript can be directed to draw on either a window or a pixmap.

Drawing on a Window

The GHOSTVIEW environment variable contains the window id of the target window. The window id is an integer. Ghostscript will use the attributes of the window to obtain the width, height, colormap, screen, and visual of the window. The remainder of the information is gotten from the GHOSTVIEW property on that window.

Drawing on a Pixmap

The GHOSTVIEW environment variable contains a window id and a pixmap id. They are integers separated by white space. Ghostscript will use the attributes of the window to obtain the colormap, screen, and visual to use. The width and height will be obtained from the pixmap. The remainder of the information, is gotten from the GHOSTVIEW property on the window. In this case, the property is deleted when read.

The GHOSTVIEW environment variable

parameters: window-id [pixmap-id]

scanf format: "%d %d"

explanation of parameters:

window-id: tells ghostscript where to
- read the GHOSTVIEW property
- send events
If pixmap-id is not present,
ghostscript will draw on this window.

pixmap-id: If present, tells ghostscript that a pixmap will be used as the final destination for drawing. The window will not be touched for drawing purposes.

(Ex. O, page 1, emphasis added.) In this way, by passing a window ID, Ghostview 1.5 told Ghostscript in which window to draw the image from a postscript file.

The file “gs.interface” describes additional communications between Ghostscript and Ghostview – events sent between Ghostscript and Ghostview.

Ghostscript sends events to the window where it read the GHOSTVIEW property. These events are of type ClientMessage. The message_type is set to either PAGE or DONE. The first long data value gives the window to be used to send replies to ghostscript. The second long data value gives the primary drawable. If rendering to a pixmap, it is the primary drawable. If rendering to a window, the backing pixmap is the primary drawable. If no backing pixmap is employed, then the window is the primary drawable. This field is necessary to distinguish multiple ghostscripts rendering to separate pixmaps where the GHOSTVIEW property was placed on the same window.

The PAGE message indicates that a "page" has completed. Ghostscript will wait until it receives a ClientMessage whose message_type is NEXT before continuing.

The DONE message indicates that ghostscript has finished processing.

(Ex. O, page 2.) This is one example of communication between X Clients using ClientMessage events. More generally, an X Client can send ClientMessage events to another X Client using calls to XSendEvent() and can handle ClientMessage events received from the other X Client.

(Ex. M, 11.3.)

For another inter-client communication mechanism, the X Server stores selections of data for copy and paste operations between X Clients. (Ex. M, 11.4.)

D. The Patent Applicants' Visible Embryo Project Software

Some of the dependent claims concern a distributed application architecture that two of the patent applicants (Doyle and Ang) described in a prior art article. The article, entitled “Processing Cross-sectional Image Data for Reconstruction of Human Developmental Anatomy from Museum Specimens,” was published in a “SIGBIO” newsletter in early 1993. (Ex. P.) The

Doyle-Ang SIGBIO article qualifies as prior art against the '906 patent as a printed publication under 35 U.S.C. § 102(b).

The Doyle-Ang SIGBIO article describes embryo visualization software developed as part of the "Visible Embryo Project" referenced in the '906 patent. (*See* Ex. P, pages 5-6.) The software operated on the X Window System and was distributed between two computers. This distributed architecture is of particular relevance to claims 4, 5, 9, and 10 of the '906 patent, according to which "additional instructions for controlling said controllable application reside on said network server."

Not coincidentally, the "application client 210" and "application server 220" in the '906 patent are software for the Visible Embryo Project. ('906 patent, column 10, lines 16-46.) "Versions and descriptions of software embodying the present invention are generally available as hyperlinked data objects from the Visible Embryo Project's World Wide Web document at the URL address 'HTTP://visembryo.ucsf.edu/'."

The Doyle-Ang SIGBIO article indicates that the Visible Embryo Project software had "already been successfully tested on workstations from Silicon Graphics, Sun, and IBM." (Ex. P, page 6.) It also reports that the embryo visualization software had been publicly demonstrated at SIGGRAPH '92, a conference in Chicago in 1992. (Ex. P, pages 5-6.) Nevertheless, in the '906 patent the patent applicants characterized the embryo visualization software as being "presently under development" ('906 patent, column 10, lines 18-19), and they failed to cite the Doyle-Ang SIGBIO article during initial prosecution or during the '831 reexamination. Even if certain aspects of the embryo visualization software were still "under development" in October 1994 as claimed, that does not change the prior art status or materiality of the Doyle-Ang SIGBIO article or the software that was "successfully tested" and demonstrated in 1992.

V. EXPLANATION OF THE PERTINENCY AND MANNER OF APPLYING CITED PRIOR ART

Reexamination of claims 1-10 of the '906 patent is requested for the following

reasons:

- Claims 1 and 6 of the '906 patent are anticipated by Janssen (as Janssen would have been understood in the context of the general knowledge in the art evidenced by Raggett II, Mosaic, and X Window System art).
- Claims 1 and 6 of the '906 patent are obvious over APA Mosaic in view of Raggett II and Janssen.
- Claims 2, 3, 7, and 8 of the '906 patent are obvious over Janssen (or alternatively APA Mosaic in view of Raggett II and Janssen) in view of Young (Ex. M) or Ghostview 1.5 (Ex. L and O).
- Claims 4, 5, 9, and 10 of the '906 patent are obvious over Janssen (or alternatively APA Mosaic in view of Raggett II and Janssen) in view of Young (or Ghostview 1.5) and the Doyle-Ang SIGBIO article (Ex. P).

The patent owner did not present to the Examiner or ask the Examiner to consider any of the Janssen, Young, Ghostview 1.5 and Doyle-Ang SIGBIO prior art, either during initial prosecution of the '906 patent or during the '831 re-examination. These references are without doubt highly relevant as they disclose the very claim requirements the Examiner found lacking in the limited prior art previously considered by the Office in the '831 re-examination.

A. The Janssen Posting Anticipates Claims 1 and 6

Claims 1 and 6 of the '906 patent recite a prior art method and product whereby a browser handles a foreign-format data object identified in a special text code and having associated type information by automatically invoking an external application to provide immediate interaction and display of that object. ('906 patent, Title.) Specifically, claims 1 and 6 each recite the limitation, "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.” In the Examiner’s Statement of Reasons for Patentability and/or Confirmation, dated September 27, 2005, the Office interpreted this claim language as follows:

Significantly, the instant claimed ‘interactive processing’ of the ‘906 patent begins at the moment the browser application parses an ‘embed text format’ detected within the hypermedia document. The web browser invokes the claimed ‘executable application’ immediately after an ‘embed’ tag is parsed and before the hypermedia document is completely displayed in the browser-controlled window. The invoked ‘executable application’ enables the claimed ‘interactive processing.’

Instant ‘906 independent claims 1 and 6 therefore require an operative coupling between the claimed ‘executable application’ and the claimed ‘interactive processing’ such that the claimed ‘interactive processing’ must be enabled by an ‘executable application’ that meets five explicitly claimed requirements:

1. The executable application must be external to the first distributed multimedia document.
2. The executable application must be automatically invoked by the browser application when the ‘embed text format’ is parsed by the browser application.
3. The executable application must execute on the client workstation.
4. The executable application must display the object within the display area created at the first location within the portion of the first distributed hypermedia document being displayed in the first browser-controlled window.
5. The executable application must enable interactive processing of the object within the display area created at the first location within the portion of the first distributed hypermedia document being displayed in the first browser-controlled window.

...

As discussed supra, a proper construction of the claimed ‘interactive processing’ necessarily requires some capability of ongoing real-time manipulation and control by the user that is applied to the object displayed within the first browser-controlled window. It is axiomatic that an executable application that terminates is incapable of providing the type of ‘interactive processing’ required by instant ‘906 independent claims 1 and 6.

(Pages 9-10, emphases in original.)

Janssen's expansion of Raggett II meets all limitations of claims 1 and 6, and it satisfies all requirements set forth in the above-quoted Examiner's Statement of Reasons for Patentability and/or Confirmation. More specifically, a person of skill in the art reading Janssen in its proper X Window System context (i.e., Raggett II, Mosaic, available interactive helper applications, and the X Window System), would have discerned each and every claim limitation of claims 1 and 6, as demonstrated in the claim chart below.

Janssen in particular discloses each of the five claim requirements identified by the Examiner in the above-quoted action:

1. The executable application must be external to the first distributed multimedia document: Janssen's executable "sub-program" is external to the HTML document. (Ex. B.) Raggett II (on which Janssen expands) describes the separate program as being external to the HTML document. (Ex. C.) In Raggett II, the browser identifies the format of the foreign-format data from a MIME type attribute in an EMBED tag parsed from the HTML document. (*Id.*) The MIME content type has been bound to a separate program, "e.g., via X resources or a config file." This description refers to a well-known MIME type mechanism by which a program relates a MIME type for a data format to an available application on the computer capable of processing that format of data. (*See* Ex. F-H.) Indeed, this is the exact same MIME type binding mechanism used with the EMBED tag described in the '906 patent at column 15, lines 9-16.

2. The executable application must be automatically invoked by the browser application when the 'embed text format' is parsed by the browser application: Janssen also incorporated this automatic invocation feature from Raggett II. Specifically, the separate

program is automatically invoked by the browser when it parses the EMBED tag. (Ex. C.) In Raggett II, the EMBED tag includes, for example, a URL that references foreign-format data in a separate file. (*Id.*) Upon parsing the EMBED tag, the Janssen browser interprets the MIME type specified in the EMBED tag, creates a sub-window where the foreign-format data will be displayed, and invokes the sub-program appropriate to the format, resulting in inline display of the foreign-format data in the sub-window of the browser window. (Ex. B and C.)

3. The executable application must execute on the client workstation: The Janssen sub-program executed, at least in part, on the client workstation. This was standard operating procedure at the time; the interactive Mosaic helper applications ran, predominately, on the client workstation on which the browser ran. (*See* Exhibits I, J, and L.)

4. The executable application must display the object within the display area created at the first location within the portion of the first distributed hypermedia document being displayed in the first browser-controlled window: Janssen describes this as well -- the foreign-format data is displayed within the “inset” area within the HTML document displayed in the Janssen browser’s window. (Ex. B.) The Janssen browser creates the X sub-window over the “area where the inset is to be displayed.” This sub-window is “refreshed” by the separate program to render the foreign-format data. (*Id.*)

5. The executable application must enable interactive processing of the object within the display area created at the first location within the portion of the first distributed hypermedia document being displayed in the first browser-controlled window: As explained above, adding this interactivity was one way in which Janssen expanded upon Raggett II. Janssen expressly disclosed that the sub-program continued to run and handled “all

events” in the sub-window. The plain meaning of “all events” is *every* event, including user input events, not just “some” events.

More specifically, the Janssen browser enables interactive processing of the embedded data object within the sub-window within the browser’s HTML document display window. Janssen states, “the browser [passes] the window ID of the sub-window to the subprogram which understands the inset format, with the understanding that that program is to handle *all events* and refresh on the sub-window.” As previously discussed, this statement would be understood in the context of the X Window System to refer to well known mechanisms of a first program passing a window ID to a second program, allowing the second program to select events (including mouse clicks, keystrokes, etc.) to receive for the window and handle such events (e.g., by setting an event mask and registering an event handler) and refresh or update the display of that window accordingly. (See Exs. M, N, and O.) The Mosaic helper applications Xv and Ghostview responded to user input in their window(s) (including the main display window) by changing how content was viewed or changing the characteristics of the content. (See Exs. I, J, and L.) The Janssen browser thus enables, through the separate program, ongoing real-time manipulation and control by the user of the embedded content in the sub-window. Because the separate program remains active in order to handle “all events and refresh” on the sub-window as described by Janssen, it does not terminate.

As summarized and shown in the following claim chart, the Janssen posting includes all recited claim elements of claims 1 and 6. Moreover, as just discussed, the Janssen posting meets all requirements of the Examiner’s interpretation of the above-discussed “interactive processing” clause of claims 1 and 6. Accordingly, reexamination of claims 1 and 6 based on Janssen is appropriate.

Claim 1 of the '906 Patent	Prior Art
<p>A method for running an application program in a computer network environment, comprising:</p>	<p>Janssen would have been understood in the context of the general knowledge in the art evidenced by the Raggett II, APA Mosaic, and X-Window System art.</p> <p>Janssen describes a “subprogram which understands the inset format.” (Ex. B.) The “subprogram” runs in a computer network environment. A Mosaic helper application is one such “subprogram” and is used as a viewer for the “inset format.” (APA Mosaic. <i>See, e.g.</i>, ‘906 patent, column 3, lines 13-16.)</p>
<p>providing at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment;</p>	<p>Janssen describes a browser. (Ex. B.) The browser executes on a client workstation and interacts with a network server coupled to the network environment, which is a distributed hypermedia environment. (APA Mosaic. <i>See, e.g.</i>, ‘906 patent, column 5, lines 34-36, and Response received May 12, 2004, page 3 (“When the browser is launched on a client workstation...”))</p>
<p>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; 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</p>	<p>Janssen describes a browser. (Ex. B.) The browser:</p> <ul style="list-style-type: none"> • executes on a client workstation; • parses a hypermedia document to identify text formats included in said distributed hypermedia document; • responds to predetermined text formats to initiate processing specified by said text formats; and • causes display, on the client workstation, of at least a portion of the hypermedia document received over the network from the network server, where the portion of the hypermedia document is displayed within a browser-controlled window on the client workstation. <p>(APA Mosaic. <i>See, e.g.</i>, ‘906 patent, column 5, lines 28-38, and Response received May 12, 2004, page 3 (“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 “The browser application then parses the local copy of the HTML document, renders the temporary local copy of the HTML document in to a Web page, and displays the rendered Web page in a browser-controlled window.”))</p>

<p>wherein said first distributed hypermedia document includes an embed text format, located at a first location in said first distributed hypermedia document that specifies the location of at least a portion of an object external to the first distributed hypermedia document</p>	<p>Janssen is a reply to Raggett II and describes processing an EMBED tag included in a hypermedia document. (Ex. B.) The EMBED tag can have a URL that references “foreign data in a separate file.” (Raggett II, Ex. C. “Well both of these will be possible with the HTML+ DTD, by using the capability to embed foreign formats inline in the HTML+ source, e.g. <code><H2>A example of an equation</H2></code> <code><EMBED TYPE=“text/eqn”>zeta (s) ~~~ sum from k=1 to inf k sup -s~~~ (Re s > 1) </EMBED>”</code> “p.s. you can also put the foreign data in a separate file referenced by a URL.”)</p>
<p>wherein said object has type information associated with it utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document</p>	<p>Janssen describes a “MIME content type.” (Ex. B.) The “MIME content type” is indicated in the EMBED tag, and the browser utilizes the MIME content type to identify and locate an executable application external to the first distributed hypermedia document. (Raggett II, Ex. C. “The browser identifies the format of the embedded data from the ‘type’ attribute, specified as a MIME content type.”)</p>
<p>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.</p>	<p>Janssen describes a browser that automatically invokes a “subprogram which understands the inset format” to execute on the client workstation, and the “subprogram” causes display of the foreign data in the “inset format.” (Ex. B. “My favorite way to handle this is to have 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 that program is to handle all events and refresh on the sub-window, but the browser gets to handle configuration and window movement.” <i>See also</i> Raggett II, Ex. C. “Browsers can then be upgraded to display new formats without changing their code at all. All you would need is a way of binding the MIME content type to the function name for that format, e.g., via X resources or a config file.”)</p> <p>In this way, Janssen describes the browser enabling interactive processing of the foreign data within a display area created at the location within the portion of the hypermedia document being displayed in the browser-controlled window. (<i>Id.</i>)</p>

B. APA Mosaic, Raggett II, and Janssen Render Obvious Claims 1 and 6

Janssen in combination with Raggett II and APA Mosaic renders the subject matter of claims 1 and 6 of the '906 patent obvious. Raggett II provides the suggestion to modify existing Mosaic browser technology inasmuch as such modifications are described in Raggett II. Janssen similarly expands upon the Raggett II/Mosaic combination.

More particularly, APA Mosaic teaches a hypermedia browser that meets all limitations of claims 1 and 6, except the limitations, “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” 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.”

Raggett II describes modifying the Mosaic browser to provide the capability to “embed foreign formats inline” in the hypermedia document, using an EMBED tag that has a “type attribute” specifying a “MIME content type” of the foreign format. (Ex. C.) Raggett II further describes binding the content type specified in this embed tag to functions implemented as separate programs for displaying the foreign format. Raggett II states, “Browsers can then be upgraded to display new formats without changing their code at all.” (*Id.*) With this statement, Raggett II illustrates one motivation to modify the Mosaic browser to support the EMBED tag and bind content types to appropriate separate programs. (*Id.*) Raggett II also describes the

EMBED tag specifying the location of foreign-format data external to the HTML document.

(Ex. C. “[Y]ou can also put the foreign data in a separate file referenced by a URL.”)

The Janssen posting describes further modifying the Raggett II/Mosaic combination to have the browser create an “X sub-window” where “the inset is to be displayed.” (Ex. B.) The browser then passes the window ID of this sub-window to the “subprogram” that handles the foreign format, “with the understanding that that program is to *handle all events* and refresh on the sub-window.” (*Id.*) Well-known mechanisms existed in the X Window System to pass a sub-window ID to a separate program and enable interactive processing of the content in the sub-window by the separate program, which handled all events, such as a user’s keyboard and mouse inputs, directed to the sub-window. (*See, e.g.,* Ex. M, 5.2; Ex. N, 2.5.2.) In fact, the Ghostview 1.5 software discussed above utilized such mechanisms. (Ex. O.)

Janssen does not directly state why this mechanism is his “favorite way to handle this.” (Ex. B.) However, Janssen does state that passing the window ID allows the separate program “to handle all events and refresh on the sub-window.” This would be readily understood by those familiar with X Window System programming (as evidenced by, e.g., Exhibits M, N, and O) as a mechanism that enables interactive processing of the foreign format content in the sub-window by the separate program responding to keyboard/mouse input events. Accordingly, apart from the fact that Janssen directly and explicitly modifies the Raggett II/Mosaic browser combination, Janssen provides a motivation for this further modification of the Raggett II/Mosaic browser – using proven techniques to enable interactive processing by the separate program.

As already fully discussed in the previous section, the combination of Janssen, Raggett II and APA Mosaic meets all requirements of the claim language as interpreted by the

Office. Accordingly, reexamination of claims 1 and 6 based on the combination of Janssen, Raggett II and APA Mosaic is appropriate.

C. Janssen and Young/Ghostview Render Obvious Claims 2, 3, 7 and 8

Claims 2, 3, 7 and 8 of the '906 patent are dependent claims, which add further limitations relating to the browser controlling the separate program via inter-process communication. Claim 2 recites the limitation of “interactively controlling said controllable application on said client workstation via inter-process communications between said browser and said controllable application.” Claim 3 recites the further limitation, “wherein the communications to interactively control said controllable application continue to be exchanged between said controllable application and the browser even after the controllable application program has been launched.” Claims 7 and 8 recite nearly identical limitations.

In the X Window System, there are various mechanisms available for inter-process communications between client programs, including environment variables (“atoms and properties”), client message events, and the X selection mechanism. (Ex. M, 11.1 to 11.4, page 280; Ex. O.) With client message events, a client program can send an event message to another client program by calling the XSendEvent() function of the X Server. (Ex. M, 11.3, page 294, “This feature can be used to forward events from one application to another, or to create and send new events”; Ex. O.)

Young describes a program called “xtalk” that allows users on two different machines to communicate with each other by text. (Ex. M, 11.3, page 296.) The xtalk program on one user’s machine sends/receives client message events to/from the xtalk program on the other user’s machine, for example, to make connection requests, including “disconnect” and

“accept” notifications, and to send/receive keyboard events that affect display on the other user’s machine. (Ex. M, 11.3, page 297.)

Ghostview 1.5 provides an example of inter-process communication in the context of a viewer application (namely, Ghostview) and separate rendering program (namely, Ghostscript). (Ex. L, page 5.) The Ghostview and Ghostscript programs communicate using an environment variable to pass a window ID, as discussed above. (Ex. O.) Ghostview and Ghostscript also communicate with each other on a continuing basis by sending client message events, such as “PAGE” (Ghostscript indicating to Ghostview that it completed drawing a page), “NEXT” (Ghostview indicating to Ghostscript to draw a next page) and “DONE” (Ghostscript indicating to Ghostscript that it has drawn up to end of the file). (Ex. O, page 1.)

Xtalk and Ghostview/Ghostscript show programs using inter-process communication to control display in a window or sub-window by another program. Xtalk illustrates using client message events for one program to forward keyboard events that affect display in a separate program’s window. Ghostview sends client message events to control paged display by Ghostscript in a window whose ID was passed by Ghostview.

Janssen teaches a browser system that enables interactive processing of inset foreign-format data using a separate program. Upon parsing an EMBED tag, the Janssen browser creates an X sub-window over an area of the HTML document where the foreign-format data is to be displayed. The Janssen browser passes a window ID for this sub-window to the separate program, which then handles “all events and refresh on the sub-window” while the browser handles “configuration and window movement.” (Ex. B.) If the browser causes a change to the position, size, or border of the sub-window, this results in a change notification being sent to the separate program, which the separate program handles. (Ex. M, 5.3.7.) Aside

from this type of event, Young's teaching (or alternatively that of Ghostview 1.5) to use client message events for inter-process communication to control display in a sub-window by another program would motivate like use of inter-process communication by the Janssen browser to control display or exchange status information. For example, the Janssen browser would directly communicate to the subprogram window configuration and movement changes (such as window re-sizing, scrolling or paging of the HTML document display) that affect the separate program's inset data display.

Note the similarities to inter-process messages sent between the browser and external program described in the '906 patent. In the table shown at column 12, lines 15-17 of the '906 patent, the set of predefined messages includes the messages: "server update done," "server ready," "server exiting," "area shown," "area hidden," and "area destroyed." Both the MEAPI inter-process communications and Ghostview's message events control paged display to effect the "understanding" expressed in the Janssen posting, i.e., that "the browser gets to handle configuration and window movement" while the subprogram "is to handle all events and refresh on the sub-window." (Ex. B.)

The Janssen browser as modified by the teaching of Young or Ghostview 1.5 meets the added claim limitations of claims 2, 3, 7 and 8. These claims therefore would have been obvious over Janssen in view of either reference.

Claim 2 of the '906 Patent	Prior Art
The method of claim 1, wherein said executable application is a controllable application and further comprising the step of: interactively controlling said controllable application via inter-process communications between said browser and said controllable	Young describes various inter-process communication mechanisms in the X Window System, including environment variables, client message events and X selection. (Ex. M, 11, pages 280-332.) More particularly, Young describes a program sending client message events to control another application, such as one user's xtalk program sending "connection requests," "disconnect" and "accept" notifications, as

application.	<p>well as forwarding keyboard input events, to a subwindow of another user's xtalk program. (Ex. M, 11.3, page 297.)</p> <p>The Ghostview 1.5 documentation describes Ghostview interactively controlling Ghostscript, including by passing a window ID via an environment variable and sending client message events to control paged display. (Ex. S.)</p>
--------------	---

Claim 3 of the '906 Patent	Prior Art
<p>The method of claim 2, wherein the communications to interactively control said controllable application continue to be exchanged between said controllable application and the browser even after the controllable application has been launched.</p>	<p>Young describes one program controlling another program (e.g., two xtalk programs) via inter-process communication mechanisms, including environment variables, client message events and X selection, which occurs after launch of the other program. (Ex. M, 11.1-11.4.)</p> <p>The Ghostview 1.5 documentation describes Ghostview interactively controlling Ghostscript, including by passing a window ID via an environment variable and sending client message events to control page display, which occur after Ghostview launches Ghostscript. (Ex. O.)</p>

**D. Janssen, Young/Ghostview, and Doyle-Ang
Render Obvious Claims 4, 5, 9 and 10**

Claims 4, 5, 9 and 10 of the '906 patent are dependent claims, which add further limitations relating to the separate program executing on both the client workstation and network server as a distributed application. Claim 4 recites limitations of "issuing... commands to the network server," "executing, on the network server," "sending information from said network server... and processing said information at the client workstation to interactively control said controllable application." Claim 5 recites the further limitation, "wherein the communications to interactively control said controllable application continue to be exchanged between said

controllable application and the browser even after the controllable application program has been launched.” Claims 9 and 10 recite nearly identical limitations.

The Doyle-Ang SIGBIO article describes software developed as part of the “Visible Embryo Project.” (Ex. P.) The article reads, in part:

Software tools were developed to allow the interactive three-dimensional visualization of the embryo reconstruction in real time. Figure 3 shows the display of the application as it appeared at the SIGGRAPH '92 conference in Chicago (Doyle, et al., 1992). The left of the screen shows a surface-based model of the embryo's exterior. This model was built from data which was derived, through three-dimensional interpolation, from the original embryo dataset. Two-hundred volume slices of the embryo (stored as texture maps) can be interactively displayed at this lower resolution while the model is rotated freely in three dimensions. A cutting-plane can be seen to intersect the surface-based model. This cutting plane can be interactively controlled to intersect with the embryo model at any arbitrary angle and position. To the right of the screen, one can see a window that displays a high-resolution image of the oblique section through the embryo as indicated by the interactive cutting plane. *In order to maintain the quick response needed for effective real-time interaction, the computational load of this application was distributed so that the interface panel, seen at the bottom of the screen, and the 3-D surface model were running on the CPU of the Silicon Graphics workstation. Computation of the high-resolution oblique section image displayed in the right window took place on the Convex supercomputer.* Both of these operations occurred simultaneously, communicating through a high-speed fiber optic network.

Current efforts are being directed towards the development of a very portable tool for viewing arbitrary oblique slices through such data. This program allows interactive display of orthogonal and oblique slices through volumetric data without using any machine-specific functions. The application is written in pure ANSI-standard C and *uses the X Window (Motif) toolkit for its interface. It has already been successfully tested on workstations from Silicon Graphics, Sun, and IBM.*

(Ex. P, pages 5-6.)

According to the article, the embryo visualization software is distributed between two computers separated by a network – the “interface panel” and “3-D surface model” run on a workstation while an “oblique section image” is computed on a supercomputer (network server).

The operation of such a distributed software application necessarily involves issuing of commands to the network server and sending of response information to the workstation.

Although the Doyle-Ang SIGBIO article presents little technical detail about communication across the network, certain features are inherent to communication between a workstation and network server in distributed software applications. These inherent features include: issuing of commands from the workstation to the network server, execution of instructions on the network server, sending of information from the network server back to the workstation, and processing of the information at the workstation.

In the article, the authors indicate that this split of the computational load between supercomputer and workstation was made to “maintain the quick response needed for effective real-time interaction.” This teaching to split the computational load between a workstation and other computer so as to maintain real-time interactivity would have motivated distributing the load of computationally intensive subprograms for the Janssen (in combination with Young or Ghostview) browser when required for real-time interactivity.

The combination Janssen and Young (or Ghostview) as modified by the teaching of the Doyle-Ang SIGBIO article meets the added claim limitations of claims 4, 5, 9 and 10. These claims therefore would have been obvious.

Claim 4 of the '906 Patent	Prior Art
The method of claim 3, wherein additional instructions for controlling said controllable application reside on said network server, wherein said step of interactively controlling said controllable application includes the following sub-steps:	The Doyle-Ang SIGBIO article describes instructions controlling the three-dimensional embryo visualization application residing on the supercomputer (network server). (Ex. P, page 6: “Computation of the high-resolution oblique section image displayed in the right window took place on the Convex supercomputer.”)
issuing, from the client workstation,	The Doyle-Ang SIGBIO article describes the interface

one or more commands to the network server;	panel running on the workstation and communicating with the network server. (Ex. P, page 6: "Both of these operations [interface panel and sectional computation] occurred simultaneously, communicating through a high-speed fiber optic network.")
executing, on the network server, one or more instructions in response to said commands;	The Doyle-Ang SIGBIO article describes the oblique sectional image computation being performed on the network server under interactive control of the interface panel on the workstation. (Ex. P, page 6: "This cutting plane can be interactively controlled to intersect with the embryo model at any arbitrary angle and position. To the right of the screen, one can see a window that displays a high-resolution image of the oblique section through the embryo as indicated by the interactive cutting plane. ... Computation of the high-resolution oblique section image displayed in the right window took place on the Convex supercomputer.")
sending information from said network server to said client workstation in response to said executed instructions; and	The Doyle-Ang SIGBIO article describes the oblique sectional image computed at the network server being communicated to the workstation for display. (Ex. P, page 7: "This program allows interactive display of orthogonal and oblique slices through volumetric data without using any machine-specific functions"; Figure 3)
processing said information at the client workstation to interactively control said controllable application.	The Doyle-Ang SIGBIO article describes the interface panel running on the workstation interactively controlling the display of the oblique sectional image data. (Ex. P, page 6: "This cutting plane can be interactively controlled to intersect with the embryo model at any arbitrary angle and position. To the right of the screen, one can see a window that displays a high-resolution image of the oblique section through the embryo as indicated by the interactive cutting plane.")


Claim 5 of the '906 Patent	Prior Art
The method of claim 4, wherein said additional instructions for controlling said controllable application reside on said client workstation.	The Doyle-Ang SIGBIO article describes the interface panel and 3-D surface model program as residing on the workstation. (Ex. P, page 6: "the computational load of this application was distributed so that the interface panel, seen at the bottom of the screen, and the 3-D surface model were running on the CPU of the Silicon Graphics workstation.")

VI. CONCLUSION

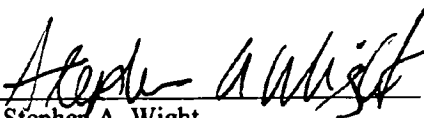
For the above reasons, reexamination of claims 1-10 of the '906 patent is hereby respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

By 

Kyle B. Rinehart
Registration No. 47,027

By 

Stephen A. Wight
Registration No. 37,759


One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

CERTIFICATE OF SERVICE

I certify that a true and correct copy of the foregoing document (with exhibits attached) was served, as indicated below on The Regents of the University of California, on December 22, 2005.

The Regents of the University of California
c/o: Law Office of Charles E. Krueger
P.O. Box 5607
Walnut Creek, CA 94596-1607

Via First Class Mail



Stephen A. Wight

858 PH Ex. 2



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 02/09/2006

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

EXAMINER

ART UNIT PAPER NUMBER

DATE MAILED: 02/09/2006

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)

Klarquist Sparkman, LLP
One World Trade Center
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)).

Order Granting / Denying Request For Ex Parte Reexamination	Control No. 90/007,858	Patent Under Reexamination 5838906	
	Examiner St. John Courtenay III	Art Unit 3992	

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

The request for *ex parte* reexamination filed 22 December 2005 has been considered and a determination has been made. An identification of the claims, the references relied upon, and the rationale supporting the determination are attached.

Attachments: a) PTO-892, b) PTO-1449, c) Other: _____

1. The request for *ex parte* reexamination is GRANTED.

RESPONSE TIMES ARE SET AS FOLLOWS:

For Patent Owner's Statement (Optional): TWO MONTHS from the mailing date of this communication (37 CFR 1.530 (b)). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).**

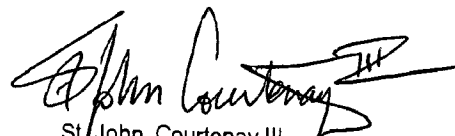
For Requester's Reply (optional): TWO MONTHS from the **date of service** of any timely filed Patent Owner's Statement (37 CFR 1.535). **NO EXTENSION OF THIS TIME PERIOD IS PERMITTED.** If Patent Owner does not file a timely statement under 37 CFR 1.530(b), then no reply by requester is permitted.

2. The request for *ex parte* reexamination is DENIED.

This decision is not appealable (35 U.S.C. 303(c)). Requester may seek review by petition to the Commissioner under 37 CFR 1.181 within ONE MONTH from the mailing date of this communication (37 CFR 1.515(c)). **EXTENSION OF TIME TO FILE SUCH A PETITION UNDER 37 CFR 1.181 ARE AVAILABLE ONLY BY PETITION TO SUSPEND OR WAIVE THE REGULATIONS UNDER 37 CFR 1.183.**

In due course, a refund under 37 CFR 1.26 (c) will be made to requester:

- a) by Treasury check or,
- b) by credit to Deposit Account No. _____, or
- c) by credit to a credit card account, unless otherwise notified (35 U.S.C. 303(c)).


 St. John Courtenay III
 Primary Examiner
 Art Unit: 3992

cc: Requester (if third party requester)

Response to Request for *ex parte* Reexamination

1. Reexamination has been requested for claims 1-10 of U.S. Patent number 5,838,906 ('906 patent).
2. A substantial new question of patentability affecting claims 1-10 of United States Patent number 5,838,906 is raised by the request for *ex parte* reexamination.
3. A prior art patent or printed publication raises a substantial question of patentability where there is a substantial likelihood that a reasonable Examiner would consider the prior art patent or printed publication **important** in deciding whether or not the claim is patentable. See MPEP §2242 (I), 3rd paragraph.
4. The '906 patent is currently assigned to The University of California, 1320 Harbor Bay Parkway, Suite 150 Alameda, CA 94502.
5. The '906 patent application filing date was Oct. 17, 1994 and the patent issued on Nov. 17, 1998. There is no claim to priority on record.
6. In the instant request for reexamination, the requester argues that '906 patent claims 1-10 are either anticipated or obvious in light of the following references:

(Exhibit B) Bill Janssen, "Re: HTML+ support for eqn & Postscript," www-talk email list, 1 page (June 14, 1993).

Art Unit: 3992

(Exhibit C) Dave Raggett, "HTML+ support for eqn & Postscript," www-talk email list, 2 pages (June 14, 1993).

(Exhibit D) Marc Andreessen, "NCSA Mosaic for X 1.2 available," www-talk email list, 3 pages (June 30, 1993).

(Exhibit E) Anonymous, "xresources.h," 15 pages (1993). (Downloaded from "Index of /pub/mosaic/Unix/source/old" as part of the file "xmosaic-1.2.tar.z"; Exhibit E has 15 pages, including printouts for: (1) the web page for downloading "xmosaic-1.2.tar.z," (2) the contents of "xmosaic-1.2.tar.z," and (3) the file "xresources.h" in "xmosaic-1.2.tar.z.")

(Exhibit F) Marc Andreessen, "NCSA Mosaic for X 2.0 prerelease 4 available," www-talk email list, 4 pages (September 29, 1993).

(Exhibit G) Anonymous, "Default File Extensions in Mosaic 2.0pre4," 1 page.

(Exhibit H) Anonymous, "Default MIME Types in Mosaic 2.0pre4," 1 page.

(Exhibit I) John Bradley, "xv, Interactive Image Display for the X Window System," 72 pages (1992). (Downloaded from "Index of /root/usr.local-src/xv-2.21" as part of the file "xv-2.21.tar.z"; Exhibit I has 77 pages, including printouts for: (1) the web page for downloading "xv-2.21.tar.z," (2) the contents of "xv-2.21.tar.z," and (3) the file "xvdocs.ps.z" in "xv-2.21.tar.z").

(Exhibit J) John Bradley, "xv, Interactive Image Display for the X Window System," 105 pages (April 26, 1993). (Downloaded from "Index of /pub/net/infosys/NCSA/Web/Mosaic/Unix/view" as part of the file "xv-3.00.tar.z"; Exhibit J has 110 pages, including printouts for: (1) the web page

Art Unit: 3992

for downloading "xv-3.00.tar.z," (2) the contents of "xv-3.00.tar.z," and (3) the file "xvdocs.ps.z" in "xv-3.00.tar.z").

(Exhibit K) Marc Andreessen, "xv 3.0' is out....," www-talk email list, 1 page (April 30, 1993).

(Exhibit L) Timothy Thiesen, "Ghostview(1) Unix Programmer's Manual," 14 pages (July 1993). (Downloaded from "ftp://mirror.cs.wesc.edu/pub/mirrors/ghost/gnu/ghostview/" as part of the file "ghostview-1.5.tar.gz"; Exhibit L has 21 pages, including printouts for: (1) the FTP site for downloading "ghostview-1.5.tar.gz," (2) the contents of "ghostview-1.5.tar.gz," (3) the file "ghostview.ps" in "ghostview-1.5.tar.gz," and (4) the file "README" in "ghostview-1.5.tar.gz.")

(Exhibit M) Douglas Young, The X-Window System. Programming and Applications with Xt, Prentice Hall, title page, copyright page, pages i-x, 1-13, 123-166, 280-332, 520-533 (1990).

(Exhibit N) Adrian Nye, Xlib Programming Manual for Version 11, O'Reilly & Associates, Inc., title page, copyright page, pages i-xxxiii, 1-46, and index (1988).

(Exhibit O) Anonymous, "gs.interface," 2 pages (July 1993). (Downloaded from "ftp://mirror-cs.wesc.edu/pub/mirrors/ghost/gnu/ghostview/" as part of the file "ghostview-1.5.tar-gz.")

(Exhibit P) Doyle et al., "Processing Cross-sectional Image Data for Reconstruction of Human Developmental Anatomy from Museum Specimens," Newsletter of the Association for Computing Machinery Special Interest Group

Art Unit: 3992

on Biomedical Computing, vol. 13, no. 1, ACM Press, cover page, table of contents, pages 9-15 (February 1993).

7. Consideration of "old art"

Exhibit C (Dave Raggett) was considered by the Office as a prior art publication during the course of the previous *ex parte* reexamination proceeding. See *ex parte* reexamination No. 90/006,831, where the instant **Exhibit C** publication is cited as: *Raggett, D., Posting to WWW-Talk Public Mailing List, (Posted June 14, 1993)*. This publication was referenced as "Raggett II" within the 90/006,831 office actions of record.

The requester also relies upon the **Mosaic browser** that was previously considered by the Office as **Admitted Prior Art (APA)**.

For the sake of expediency, such art is referred to as "old art" throughout, since the term "old art" was coined by the Federal Circuit in its decision of *In re Hiniker*, 150 F.3d 1362,1365-66, 47 USPQ2d 1523, 1526 (Fed. Cir. 1998). The request for reexamination is therefore based at least in part on patents and/or printed publications already cited/considered in an earlier concluded examination of the patent being reexamined.

On November 2, 2002, Public Law 107-273 was enacted. Title III, Subtitle A, Section 13105, part (a) of the Act revised the reexamination statute by adding the following new last sentence to 35 U.S.C. 303(a) and 312(a):

"The existence of a substantial new question of patentability is not precluded by the fact that a patent or printed publication was previously cited by or to the Office or considered by the Office."

Art Unit: 3992

For any reexamination ordered on or after November 2, 2002, the effective date of the statutory revision, reliance on previously cited/considered art, i.e., "old art," does not necessarily preclude the existence of a substantial new question of patentability (SNQ) that is based exclusively on that old art. Rather, determinations on whether a SNQ exists in such an instance shall be based upon a fact-specific inquiry done on a case-by-case basis.

8. **Exhibit B** ("Janssen") raises a SNQ of patentability.

It is agreed that at least the "old art" of **Mosaic (APA)** and **Exhibit C** (AKA "Raggett II") in combination with the cited **Exhibit B** publication ("Janssen") would have been considered important by a reasonable Examiner in deciding whether at least claims 1-3 and 6-8 are patentable.

In particular, Janssen's June 14, 1993 publication (**Exhibit B**) discloses using the browser to "... *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 that program is to handle all events and refresh on the sub-window, but the browser gets to handle configuration and window movement."*

There is a substantial likelihood that this teaching by Janssen would have been considered important to a reasonable Examiner in deciding whether at least claims 1-3 and 6-8 are patentable.

9. **Exhibit P** (Doyle et al.) raises a SNQ of patentability.

There is a substantial likelihood that the cited **Exhibit P** publication (Doyle et al.) would have been considered important by a reasonable Examiner (in combination with **Mosaic (APA)**, and **Exhibits B** and **C**) in deciding whether at least dependent claims 4, 5, 9, and 10 are patentable.

In particular, **Exhibit P** discloses a distributed computer system where the computational load for the application is distributed "*in order to maintain the quick response needed for effective real-time interaction ...*" [see page 13, col. 2]. The "*interface panel*" and "*3-D surface model*" run on the CPU of a local Silicon Graphics workstation. The computations for the "*high resolution oblique section image*" are performed on a networked Convex supercomputer [see page 13, col. 2, see also fig. 3].

Exhibit P further discloses: "*Two-hundred volume slices of the embryo (stored as texture maps) can be interactively displayed at this lower resolution while the model is rotated freely in three dimensions. A cutting-plane can be seen to intersect the surface-based model. This cutting plane can be interactively controlled to intersect with the embryo model at any arbitrary angle and position*" [see page 13, col. 2, see also fig. 3].

10. Exhibits D through O

Exhibits D-O are newly cited publications that are not considered cumulative to the prior art of record as these references reveal additional information and details regarding the Mosaic browser (and related Mosaic "helper" applications), the X-Window System, and MIME content types.

11. Issue not within scope of *ex parte* reexamination

It is noted that an issue not within the scope of reexamination proceedings has been raised. The requester contends that the Patent Owner failed to disclose two references that the requester alleges are material to patentability (See **Exhibit B** and **Exhibit P**, see also pages 2 and 3 of the request for reexamination).

This issue will not be considered in a reexamination proceeding. The scope of reexamination proceedings is limited to consideration of prior art patents and publications applied under the appropriate parts of 35 U.S.C. §§ 102 and 103. See 37 C.F.R. §1.552(c).

Conclusion

12. A substantial new question of patentability affecting claims 1-10 of United States Patent number 5,838,906 is raised by the request for *ex parte* reexamination.

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

14. The Patent Owner is reminded of the continuing responsibility under 37 C.F.R. § 1.565(a) to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving Patent number 5,838,906 throughout the course of this reexamination proceeding. The third party requester is also reminded of the ability to similarly apprise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP §§ 2207, 2282 and 2286.

15. All claims are subject to reexamination.

Art Unit: 3992

How to Communicate with the USPTO

ALL correspondence relating to this *ex partes* reexamination proceeding should be directed as follows:

Please mail any communications to:

Attn: Mail Stop "Ex Parte Reexam"
Central Reexamination Unit
Commissioner for Patents
P. O. Box 1450
Alexandria VA 22313-1450

Please FAX any communications to:

(571) 273-9900
Central Reexamination Unit

Please hand-deliver any communications to:


Customer Service Window
Attn: Central Reexamination Unit
Randolph Building, Lobby Level
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the Reexamination Legal Advisor or Examiner, or as to the status of this proceeding, should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

Signed:

St. John Courtenay III (Monday - Friday 9:00 AM - 5:30 PM)
Primary Examiner
Central Reexamination Unit 3992
(571) 272-3761

Conferees:
MR. Reinhart SPRE-CRU-3992
Eric Kohler 1/24/06



ST. JOHN COURTENAY III
PRIMARY EXAMINER

858 PH Ex. 3



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)

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://berkely.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:

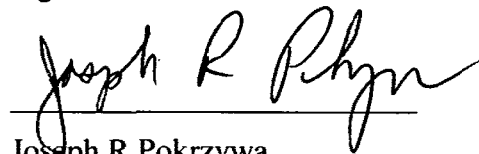
By Mail to: Mail Stop *Ex Parte* Reexam
Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

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:



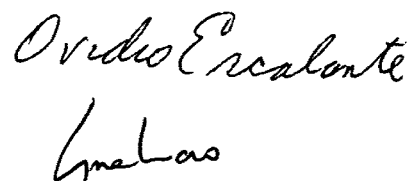
Joseph R Pokrzywa

Primary Patent Examiner

Central Reexamination Unit 3992

(571) 272-7410

Conferees:



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

30080 7590 09/06/2007

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

EXAMINER

ART UNIT PAPER NUMBER

DATE MAILED: 09/06/2007

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



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

9/19/07

THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS

STEPHEN A. WIGHT
KLARQUIST SPARKMAN LLP
121 SW SALMON STREET, SUITE 1600
PORTLAND, OR 97204

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO 90/007858
PATENT NO. 5,838,906
ART UNI 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 Interview Summary	Control No.	Patent Under Reexamination	
	90/007,858	5838906	
	Examiner	Art Unit	
	Joseph R. Pokrzywa	3992	

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

- (1) Joseph R. Pokrzywa (USPTO) (3) Charles E. Krueger
(2) Sue Lao, Roland Foster (USPTO) (4) Michael D. Doyle

Date of Interview: 06 September 2007

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: Patent Owner presented a PowerPoint presentation and an animation of the language of independent claim 6.

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 and 6.

Identification of prior art discussed: the "Viola" reference, the "Cohen" reference (U.S. Patent 5,367,621), and the "NCSA Mosaic" reference.

Description of the general nature of what was agreed to if an agreement was reached, or any other comments:
See Continuation Sheet.

(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).


JOSEPH R. POKRZYWA
PRIMARY EXAMINER

cc: Requester (if third party requester)

Examiner's signature, if required

Continuation of Description of the general nature of what was agreed to if an agreement was reached, or any other comments:

The Patent Owner first noted that the claimed invention of the '906 Patent was reduced to practice prior to August 16, 1994, being the date of the Viola reference, and the Patent Owner intends to submit documents under 37 CFR 1.131 to establish invention of the claimed subject matter prior to the effective date of the Viola reference.

With respect to the Cohen reference, the Patent Owner discussed differences in the claimed invention and the teachings of Cohen. Particularly, the Patent Owner stated that Cohen does not teach of the specific claimed features of: interactive processing, an embed text format, a display within the browser window, and type information used by the browser to identify and locate. Each of these features were further individually discussed in detail.

The examiner noted that upon submission of a response to the previous Office action, which incorporates these arguments, the examiner would review and reconsider the current art rejections.



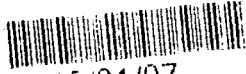
JOSEPH R. POKRZYWA
PRIMARY EXAMINER



ROLAND G. FOSTER
CRU EXAMINER-AU 3992

4x2
OPQA

858 PH Ex. 5



10/01/07

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.

<h2>TRANSMITTAL FORM</h2> <p><i>(to be used for all correspondence after initial filing)</i></p>	Application Number	REX Control No. 90/007,858
	Filing Date	12/22/2005
	First Named Inventor	Patent No. 5,838,906
	Art Unit	3992
	Examiner Name	Pokrzywa, Joseph R.
Total Number of Pages in This Submission 245	Attorney Docket Number	006-1-5

ENCLOSURES (Check all that apply)					
<input type="checkbox"/> Fee Transmittal Form	<input type="checkbox"/> Drawing(s)	<input type="checkbox"/> After Allowance Communication to TC			
<input type="checkbox"/> Fee Attached	<input type="checkbox"/> Licensing-related Papers	<input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences			
<input checked="" type="checkbox"/> Amendment/Reply	<input type="checkbox"/> Petition	<input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)			
<input type="checkbox"/> After Final	<input type="checkbox"/> Petition to Convert to a Provisional Application	<input type="checkbox"/> Proprietary Information			
<input type="checkbox"/> Affidavits/declaration(s)	<input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address	<input type="checkbox"/> Status Letter			
<input type="checkbox"/> Extension of Time Request	<input type="checkbox"/> Terminal Disclaimer	<input checked="" type="checkbox"/> Other Enclosure(s) (please identify below):			
<input type="checkbox"/> Express Abandonment Request	<input type="checkbox"/> Request for Refund	(1) Patent Owner's Statement of the Substance of the Interview			
<input type="checkbox"/> Information Disclosure Statement	<input type="checkbox"/> CD, Number of CD(s) _____	(2) Return Postcard			
<input type="checkbox"/> Certified Copy of Priority Document(s)	<input type="checkbox"/> Landscape Table on CD				
<input type="checkbox"/> Reply to Missing Parts/ Incomplete Application	<table border="1"> <tr> <td>Remarks</td> </tr> <tr> <td>(A) EACH OF THE TWO DOCUMENTS TO BE FILED HAS AN ATTACHED CERTIFICATE OF SERVICE.</td> </tr> <tr> <td>(B) THE "RESPONSE AFTER NON-FINAL REJECTION" HAS 4 ATTACHMENTS</td> </tr> </table>		Remarks	(A) EACH OF THE TWO DOCUMENTS TO BE FILED HAS AN ATTACHED CERTIFICATE OF SERVICE.	(B) THE "RESPONSE AFTER NON-FINAL REJECTION" HAS 4 ATTACHMENTS
Remarks					
(A) EACH OF THE TWO DOCUMENTS TO BE FILED HAS AN ATTACHED CERTIFICATE OF SERVICE.					
(B) THE "RESPONSE AFTER NON-FINAL REJECTION" HAS 4 ATTACHMENTS					
<input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53					

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT			
Firm Name	Law Office of Charles E. Krueger		
Signature			
Printed name	Charles E. Krueger		
Date	September 27, 2007	Reg. No.	30,077

CERTIFICATE OF TRANSMISSION/MAILING			
I hereby certify that this correspondence is being facsimile transmitted to the USPTO or 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 the date shown below:			
Signature			
Typed or printed name	Sharon D. Krueger	Date	September 27, 2007

This collection of information is required by 37 CFR 1.5. 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 1.11 and 1.14. This collection is estimated to 2 hours 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.

36155 U.S. PTO



10/01/07

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re reexamination control number:

90/007,858

Filed: 12/22/2005

Patent No: 5,838,906

Examiner: POKRZYWA, JOSEPH R

Art Unit: 3992

PATENT OWNER'S STATEMENT OF
THE SUBSTANCE OF THE INTERVIEW

Commissioner for Patents
Alexandria, VA 22313-1450

DATE: 2007-SEPT-06

ATTENDEES: Examiners Pokrzywa, Lao, and Foster; Inventor Dr. Michael D. Doyle;
Attorney of Record Charles E. Krueger.

EXHIBITS AND DEMONSTRATIONS: PowerPoint Slide presentation (Attached) and an
animation of the language of independent claim 6.

CLAIMS DISCUSSED: Independent claims 1 and 6.

REFERENCES DISCUSSED: The Viola reference, Mosaic and the Cohen patent.

SUMMARY: Dr. Doyle displayed an animation of claim 6 and the elements and limitations of
that claim were discussed with the examiners. A PowerPoint slide presentation was given by Dr.
Doyle demonstrating a reduction to practice of the subject matter of claim 6 prior to the alleged
effective date of the Viola reference and pointing out elements and limitations of claim 6 not
expressly found or inherently described in the Cohen reference.

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

56155 U.S. PTO
10/01/07

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re reexamination control number:

90/007,858

Filed: 12/22/2005

Patent No: 5,838,906

Examiner: POKRZYWA, JOSEPH R.

Art Unit: 3992

Response after Non-Final Rejection

Commissioner for Patents
Alexandria, VA 22313-1450

Sir:

In response to the Office Action mailed 07/30/2007, please consider the following remarks:

REMARKS

This is the second reexamination of U.S. Patent No. 5,838,906 ("the '906 patent"). The first reexamination was a Director Ordered Reexamination, Control No. 90/006,831 ("the first reexamination") which resulted in issuance of a Reexamination Certificate without amending the claims. Shortly after the NIRC was posted on the PAIR page the present reexamination ("the second reexamination") was requested.

The '906 patent has been involved in a related litigation, EOLAS TECHNOLOGIES, INC. and THE REGENTS OF THE UNIVERSITY OF CALIFORNIA, Plaintiffs, v. MICROSOFT CORPORATION, Defendant., Northern Dist. of Ill, No. 99 C 626, Judge Rebecca R. Pallmeyer presiding. This litigation has been settled, and Microsoft has licensed the '906 patent.

The '906 patent was also involved in Interference 105,563 McK declared 05/24/2007. This interference has been resolved in favor of the '906 patent.

The claim construction set forth in the Markman ruling in the related litigation, affirmed by the United States Court of Appeals for the Federal Circuit, is utilized in the following arguments and 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 1-10 are rejected under 35 U.S.C. §102 based on two different references. These rejections will be referred to below as the Viola rejection and the Cohen rejection.

In the Viola rejection, claims 1-10 are rejected under 35 U.S.C. §102 as being anticipated by "A Brief Overview of the VIOLA Engine, and its applications" written by Pei

Wei, pages TT05441-TT05600, which include the “Viola in a Nutshell: the Viola World Wide Web Toolkit”, from <http://scam.xcf.berkeley.edu/~wei/viola/book>. (Hereinafter “the Viola reference”)

In the Cohen rejection, claims 1-10 are rejected under 35 U.S.C. §102 as being anticipated by Cohen et al. (U.S. Patent No. 5,367,621, hereinafter “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 Number 226 (hereinafter “NCSA Mosaic”).

Included with this response is a Rule 132 Declaration (Attachment 1) by Professor Edward W. Felten, Professor of Computer Science at Princeton University (hereinafter “Felten”), traversing the rejections of claims 1 and 6 of U.S. Patent No. 5,838,906 (“the ‘906 patent). References to this declaration will be made in the following arguments.

Outline of the Argument for Claim 1 and 6

I. The Viola Rejection

A. The subject matter recited in claims 1 and 6 was reduced to practice prior to August 16, 1994.

II. The Cohen Rejection

B. Description of the References

1. Cohen
2. NCSA Mosaic

C. The Examiner’s Reasoning

D. Traverse

1. The claimed element of an executable application, external to a hypermedia document being displayed by a browser, to execute on a client workstation in order to display an object within a display area created in the hypermedia document and enable interactive processing of the object is not explicitly found or inherently described in Cohen.

2. The claimed element of an embed text format, located at a first location in said first distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document, wherein said object has type information associated with it utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document in order to display said object within a display area created at said first location within the portion of the first distributed hypermedia document is not explicitly found or inherently described in Cohen.

3. The claimed element of 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 is not explicitly found or inherently described in Cohen.

4. The claimed element of type information, associated with the object, utilized by the browser to identify and locate an executable application external to the first distributed hypermedia document is not explicitly found or inherently described in Cohen.

DETAILED ARGUMENT

I. The Viola Rejection

A. Prior Invention

A Rule 131 Declaration is attached hereto (Attachment 2) to establish reduction to practice of the subject matter of claims 1 and 6 prior to the asserted effective date of the Viola reference.

The District Court, in the related litigation, found that the invention recited in claim 6 was reduced to practice no later than January 27, 1994, when it was presented at a Web conference. (United States Court of Appeals of the Federal Circuit Opinion, Attachment 3, page 3). This date was applied by United States Court of Appeals of the Federal Circuit when determining whether asserted references were prior art.

Additionally, during the prosecution of the '906 patent two Rule 131 declarations were filed that established a reduction to practice prior to the asserted effective date of the Viola reference.

II. The Cohen Rejection

Because claims 1 and 6 contain many of the same elements and limitations, the following detailed argument is directed only to claim 6. The same arguments are equally applicable to claim 1.

A. The Claimed Invention.

The invention, as recited for example in claims 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 and interactive processing of the object within a display window created at the first location of the portion of the hypermedia document being displayed is enabled.

B . Description of the References

1. Cohen

Cohen describes a method, program and data processing system for providing a generalized link from a reference point within an organized hierarchy of a formatted text stream in an on-line book to an arbitrary type multimedia object. (2: 11-16). Hypertext link tags are described as specifying hypertext links which are created within on-line documents and between on-line documents.

In Cohen, an author-defined link is created by the book's author to establish a relationship between a source location within the softcopy text and a target location within the same text or the text in another softcopy book. The author will place a link tag in the location of the softcopy book which is the source or referencing location. Then the author will include a link description tag at the beginning of the softcopy book, where the link description tag describes the information necessary to create a link from the source link tag to the target location.

Cohen refers to two publications entitled "IBM BookMaster" which describe the BookManager BUILD and BookManager READ program products which use on-line, softcopy books which are formatted using the Generalized Markup Language (GML). (4:20-31) Using the GML described in the above referenced BookMaster publications, new tags and concepts described herein enable the creation of hypertext links within and between on-line documents. (4:34-40)

According to the Cohen invention, the BookMaster link is improved upon to provide a new multimedia link description tag LDESC in the prologue of the document. (5:8-10) The format of the LDESC is set forth at lines 13-30 of column 5 and definitions of the various attributes of the LDESC are defined from line 34 of column 5 to line 64 of column 6. Particular attribute definitions will be referenced below. Examples of LDESCs are depicted in Fig. 1a.

Cohen describes a BookMaster link tag designated :L. The link tag is used to specify a word or phrase that the author wishes to create a source link from. The link tag and the LID attribute identifies the link descriptor LDESC tag that specifies a link. The link tag L does not generate any text for on-line documents. The link L tag and its attributes are: :L LID=name, where LID=name refers to the link descriptor tag LDESC that is to be associated with the word or phrase associated with the link tag :L and its end tag :eL. 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. (7:27-30). An example of a document having link tags is depicted in Fig. 1b.

If a link is activated, either automatically or by user action, then the softcopy book reading program gets the DATA string from the LDESC identified by the LID attribute of the activated link tag. The softcopy book reading program then outputs the DATA string to start the execution of the I/O handler specified in the string. (10:49-60; Fig. 6, step 418; 11:13-15; Fig. 6, step 426) Cohen describes the function of the DATA "string" as: Lets the author pass data to multimedia object handler programs for the first, primary element. For example, string may be parameters to create a link to an animation sequence. Values for string depend on the capabilities of the user's installation. (6:8-12).

Cohen describes the operation of various I/O handler programs. A video handler program displays a CD video on the display (12:22-23; Fig. 7a, step 526), a graphics handler program generates graphics which are displayed on the display 208 (14:5-10).

After the completion of the I/O handler program whose execution was started as a result of step 426 of the flow diagram of FIG. 6, the flow proceeds back to step 420 to wait for another mouse pointer to activate a link tag or alternately to wait until a new page is requested by the user. In step 428, if there is no pointer activation or new tag requested, then the flow continues to loop back to step 420. (11:22-29, 14:9-10)

Cohen also describes a technique for updating internal objects stored in a book file. (14:21 to 15:30)

2. NCSA Mosaic

Mosaic 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 at paragraphs 15, 29; NCSA Mosaic page 463, col. 1, 3rd paragraph]. The browser then retrieves a selected published source HTML document from a network server by 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 at paragraph 15; NCSA Mosaic page 463, Fig. 1 and caption]. 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.

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. [Felten at paragraph 15; NCSA Mosaic page 463 Fig. 1 and caption] Mosaic has the capability of allowing the user to invoke an external application to open a new window to display file types that cannot be displayed by Mosaic (helper applications). [Felten at paragraph 16; NCSA Mosaic page 464, col. 1, topic 5]

For data formats that can not be rendered by the browser application itself, i.e., data in a foreign or non-native format such as ".TIF," Mosaic launches helper applications, in response to a user's command, in a separate window to view certain types of file types. 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). [Felten at paragraph 16; NCSA Mosaic page 464, col. 1, topic 5]

Many viewers exist that handle various file formats such as TIF. 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. [Felten at paragraph 16; NCSA Mosaic page 464, col. 1, topic 5]

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. [Felten at paragraph 16; NCSA Mosaic page 464, col. 1, topic 5]

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.

There is no disclosure in Mosaic of automatically invoking an external application to enable interactive processing of an object in a display area of a hypermedia document being displayed by the browser.

C. THE EXAMINER'S REASONING

The examiner states that claims 1-3 and 6-8 are rejected as being anticipated by Cohen in view of NCSA Mosaic. NCSA Mosaic is cited as a secondary reference utilized to show that the BookMaster READ product of the primary reference can be considered as a "browser application", therein proving that the Cohen reference has an "enabled disclosure".

Parts of the examiner's reasoning relating to the disclosure of particular claimed elements will be set forth below in the part of the traverse relating to the particular claimed element.

D. TRAVERSE

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.

The basic requirements anticipation are set forth in MPEP §2131:

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.

The requirements for showing that a characteristic not disclosed is the reference is inherent are set forth in MPEP §2131.III:

To serve as an anticipation when the reference is silent about the asserted inherent characteristic, such gap in the reference may be filled with recourse to extrinsic evidence. Such evidence must make it clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.

The level of skill in the relevant art is set forth in Felten as:

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 at paragraphs 10-11].

1. **The claimed element of an executable application, external to a hypermedia document being displayed by a browser, to execute on a client workstation in order to display an object within a display area created in the hypermedia document and enable interactive processing of the object is not explicitly found or inherently described in Cohen.**

a. **The claimed interactive processing is not explicitly found in Cohen.**

Claim 6 recites:

[an] executable application [that executes] .. 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.

With regard to this claimed feature, at page 27 of the office action the examiner quotes the part of claim 6 set forth above followed by citations to the Cohen reference.

The examiner then acknowledges that Cohen does not expressly state that the softcopy book reading program is a browser. It is then stated that the functionality described is the same as a browser application.

It is then stated on page 28 of the office action that, 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 of the pages as an interactive, scrollable, online book**". [emphasis added by examiner]

This rejection is respectfully traversed for the following reasons.

Claim 6 recites that the executable application executes on the client workstation in order to "enable interactive processing". Interactive processing means that the user, by using the mouse or keyboard or similar device, can change the structure or presentation of the object. [Felten at paragraph 40] As set forth above, this "interactive processing" is enabled by the execution of the executable application.

The specification describes interactive processing that allows a user to manipulate and control the object being displayed in the display area created in the browser controlled window. For example, at col. 6, line 63 to col. 7, line 6 it is stated that:

The present invention 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. The user is able to use a vast amount of computing power beyond that which is contained in the user's client computer.

At col. 7, lines 12-15 it is stated that:

Also, the user is able to rotate, scale and otherwise reposition the viewpoint with respect to these images without exiting the hypermedia browser software.

At col. 10, lines 8-10 it is stated that:

The user is then able to interactively operate controls to recompute different views for the image data.

At col. 10, lines 47-55 it is stated that:

In the present example where a multidimensional image object representing medical data for an embryo is being viewed, application server 220 could perform much of the viewing transformation and volume rendering calculations to allow a user to interactively view the embryo data at their client computer display screen. In a preferred embodiment, application client 210 receives signals from a user input device at the user's client computer 200. An example of such input would be to rotate the embryo image from a current position to a new position from the user's point of view.

At col. 15, lines 57-60 it is stated that:

The present invention allows a user to have interactive control over application objects such as three dimensional image objects and video objects

At col. 16, lines 18-22 it is stated that:

By using the controls in panel window 354 the user is able to manipulate the image within image window 352 in real time do perform such operations as scaling, rotation, translation, color map selection, etc.

Claim 6 has consistently been construed to require enablement of user interaction with the displayed object.

For example, in the Reasons for Patentability/Confirmation mailed 09/27/05 in the first reexamination of claim 6, at page 4, the examiner stated: "To be consistent with the specification, the claimed "interactive processing" necessarily requires some capability of ongoing real-time manipulation and control by the user of the object displayed within the browser-controlled window." [emphasis in original].

The Markman order of Judge Zagel (Attachment 4) in the related Microsoft Litigation, which was affirmed by the Court of Appeals for the Federal Circuit, states the following at page 4:

Second, the inventors noted the limited capability of the state-of-the-art browser (Mosaic) to provide interaction with data objects. Generally, the user needed to go outside the browser to interact with the data. “Users are limited to traditional hypertext and hypermedia forms of selecting linked data objects for retrieval and launching viewers or other forms of external software to have the data objects presented in a comprehensible way.” ’906 Patent col. 6, ll. 35- 39. At the time, these external software programs were commonly called helper applications. When Mosaic encountered web pages that required helper applications to deal with objects, it forced the user to interact with a separate display area (a pop-up window). There was no communication between helper and the browser, i.e., the browser was inactive while the helper was active. ’906 File History, Paper # 19, pp. 7-8.

The inventors envisioned “a system that allows a user at a small client computer connected to the Internet to locate, retrieve and manipulate data objects when the data objects are bandwidth-intensive and compute-intensive;” and that allows “a user to manipulate data objects in an interactive way to provide the user with a better understanding of information presented.” ’906 Patent, col. 6, ll. 40-47. An example of this idea is a browser that is capable of displaying a web page that retrieves complex 3D medical images (e.g., an image of an embryo). [emphasis added]

The language of claim 6 requires that the object having interactive processing enabled is displayed in a display area created in the portion of the hypermedia document displayed. Thus, the claim language requires that the browser is displaying the hypermedia document while the external application is enabling interactive processing with the object displayed in the display area.

None of the citations to Cohen at page 27 of the office action relate to interactive processing of a displayed object as recited in claim 6. Steps 410 -426 in Fig.6 and steps 562-579 in Fig. 7c describe locating links and outputting the data string to start the execution of the I/O handler specified in the string. Col. 10, lines 33-60 describe locating the link tags, determining whether to autolaunch the I/O handler, and outputting the data string to start the execution of the I/O handler specified in the string. Col. 13, lines 52-67 describe parsing of the data string by the I/O handler to identify if the graphic software support specified by the author in the link descriptor is present in the workstation.

In Cohen the I/O handlers are invoked to display objects. A video handler program displays a CD video on the display (12:22-23; Fig. 7a, step 526), a graphics handler program generates graphics which are displayed on the display 208 (14:5-10).

To display the object is to make it visible, e.g. by sending it to an output device. Interactive processing means that the user, by using the mouse or keyboard or similar input device, can change the structure or presentation of the object. [Felten at paragraph 40]. Thus, Cohen's disclosure relating to displaying a video or graphics object does not fairly teach or suggest interactive processing.

Accordingly, the feature of interactively controlling an object displayed by an invoked I/O handler is not expressly found in Cohen. [Felten at paragraph 41]. The examiner does not point to any support in Cohen for the element of interactive processing. [Felten at paragraph 42]

b. The claimed interactive processing is not inherent in Cohen.

As set forth above, Cohen is silent regarding the claimed interactive processing. No extrinsic evidence exists that makes it clear that the missing interactive processing element is necessarily present in the Cohen system and would be so recognized by persons of ordinary skill in the art.

The softcopy book reading program described in Cohen has been equated to a browser, such as NCSA Mosaic, by the examiner. Mosaic used a technology known as "helper applications". This technology allowed the browser to link to an external program, in cases where the browser encountered a file whose format the browser did not understand. For example, if the user clicked on a hyperlink that pointed to a file in .mpeg format (i.e., a movie in MPEG format), then the browser would launch an external MPEG-viewer program and pass the .mpeg file to that program. The result would be that the MPEG program ran in a separate window from the browser. [Felten at paragraph 16]. Helper applications allowed the browser to link to an external program, but that program could not provide interactivity within the browser window. The helper application was just an external program that ran on the same computer, in a separate window. [Felten at paragraph 17]

Further, Regarding NCSA Mosaic, both examiners in the first reexamination acknowledged that the four-way combination including Applicant's Admitted Prior Art (Mosaic) "does not explicitly teach 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". [emphasis added by examiner]. Reasons for Patentability/Confirmation mailed 09/27/05 in the first reexamination of claim 6, at page 3.

Accordingly, there is no teaching in Mosaic that makes it clear that the claimed interactive processing is necessarily present in Cohen.

Additionally, the design of Cohen was incompatible with interactive processing. In Cohen, when the main book reader program invokes an external program, it waits for the external program to finish before resuming the main book reader program. While the external program is running, the main book reader program is frozen and cannot accept input. This is evident from Fig. 6 of Cohen and the accompanying description in (e.g.) Columns 10 and 11. For example:

After the completion of the I/O handler program whose execution was started as a result of step 426 of the flow diagram of FIG. 6, the flow proceeds back to step 420 to wait for another mouse pointer to activate a link tag or

alternatively to wait until a new page is requested by the user. (Cohen at 11:22-27)

Notably, it is only after the completion of the external program that the main book reader program can accept input from the user. If the external program had tried to provide interactive processing, this would have required the external program to continue running (waiting for user interaction) which would have frozen the main book reader program, giving the user no way to switch to a different page of the e-book. The role of the external program in Cohen is to display the object and then stop running. [Felten at paragraph 44]

Finally, the fact that an object is displayed does not necessarily require that the user can interact with the object. [Felten at paragraph 43]

c. The claimed interactive processing is not enabled by Mosaic.

As set forth above, Mosaic does not fairly teach or suggest the claimed interactive processing. Accordingly, Mosaic does not show that the claimed interactive processing was in the public's possession before applicant's invention.

2. **The claimed element of an embed text format, located at a first location in said first distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document, wherein said object has type information associated with it utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document in order to display said object within a display area created at said first location within the portion of the first distributed hypermedia document is not explicitly found or inherently described in Cohen.**

a. **The claimed embed text format is not explicitly found in Cohen.**

Claim 6 recites:

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, ..., and wherein said embed text format is parsed by said browser to automatically invoke said executable application .. 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 the first distributed hypermedia document ...

With regard to this claimed feature, at page 26 of the office action the examiner quotes the part of claim 6 set forth above followed by citations to the Cohen reference.

It is stated at page 26 of the office action that the embed text format is interpreted as the link description tags LDESC included within the document. Also cited is Col. 7, lines 22-30 describing the link tag :L and its matching end tag :eL to enclose a word or phrase in the body of the document that the author wants to create a link from, and the LID attribute that refers to one or more LDESC document link tags.

This rejection is respectfully traversed for the following reasons.

The language of claim 6 recites several limitations relevant to the embed text format. First, the embed text format is located at a first location in a hypermedia document. Secondly, the embed text format specifies the location of at least a portion of an object external to the hypermedia document. Thirdly, the external object is displayed in a display area created at the first location, i.e., the location of the embed text format within the hypermedia document.

Turning first to the LDESC tag of Cohen, the LDESC tag is not located at a first location in the document where a display window is created. Instead, the LDESC (link description) tag appears in the document file's prologue. "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 ..." (Cohen at 5:8-11, emphasis added) The figures in Cohen also depict the LDESC tags as being in the file's prologue rather than in the book text. For example, Figure 1 depicts the structure of the document file. The "link description tags 102" (also depicted in Figure 1a) are

distinct from the “book text with tags 104” (which is also depicted in Figure 1b). (See also Figs. 4, 8a, and 8b.) [Felten at paragraph 52]

Turning next to the link tag :L of Cohen, the link tag :L does appear in the book text but it lacks the claimed feature that the embed text format specifies the location of at least a portion of an object external to the first distributed hypermedia document. The link tag of Cohen does not specify the location of an object, nor does it specify the location of anything that is external to the first distributed hypermedia document. [Felten at paragraph 54]

Turning finally to the requirement that the external object is displayed in a display area created at the first location, as is discussed in more detail in section 3 below, none of the citations in the office action point to a part of Cohen where the claimed display area is expressly found. In Cohen the I/O handlers are invoked to display objects, however there is no teaching relating the location of a display area.

Accordingly, the claimed embed text format is not expressly found in Cohen. [Felten at paragraph 49].

b. The claimed embed text format is not inherent in Cohen.

As set forth above, the claimed embed text format is not expressly found in Cohen. No extrinsic evidence exists that makes it clear that the missing embed text format is necessarily present in the Cohen system and would be so recognized by persons of ordinary skill in the art.

In fact the person of ordinary skill would recognize that the Cohen strategy of having a small, simple link tag that refers to a larger, more detailed link description in the document prologue, makes sense given the problem that Cohen was trying to solve. Cohen was designed for use with electronic books. These books, unlike Web pages, are large, multi-page files that often repeat graphic elements on different pages. By separating the link tag and link description, Cohen allowed an element to be repeated without having to repeat the full link description each time. Instead, there could be a single link description in the document prologue, and one small link tag at each place in the document where the object was to be used. [Felten at paragraph 55]

c. The claimed embed text format is not enabled by Mosaic.

Mosaic does not fairly teach nor suggest the embed text format but instead uses an ordinary hyperlink to link to any data that is to be displayed with a helper application. [Felten at paragraph 21]. Further, as acknowledged by the examiner at paragraph 14: “the NCSA Mosaic references ... 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...**” (emphasis in original).

Accordingly, Mosaic does not show that the claimed embed text format was in the public’s possession before applicant’s invention.

3. **The claimed element of 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 is not explicitly found or inherently described in Cohen.**

a. The claimed display area is not explicitly found in Cohen.

Claim 6 recites:

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.

With regard to this claimed feature, at page 27 of the office action the examiner quotes the part of claim 6 set forth above followed by citations to the Cohen reference.

This rejection is respectfully traversed for the following reasons.

The language of claim 6 recites several limitations relevant to the display area. First, the display area is located at a first location in a hypermedia document. Secondly, that the external object is displayed in a display area created at the first location, i.e., the location of the embed text format within the hypermedia document, of the portion of the hypermedia document being viewed.

None of the citations at page 27 of the office action point to a part of Cohen where the claimed display area is expressly found. Steps 410 -426 in Fig.6 and steps 562-579 in Fig. 7c describe locating links and outputting the data string to start the execution of the I/O handler specified in the string. Col. 10, lines 33-60 describe locating the link tags, determining whether to autolaunch the I/O handler, and outputting the data string to start the execution of the I/O handler specified in the string. Col. 13, lines 52-67 describe parsing of the data string by the I/O handler to identify if the graphic software support specified by the author in the link descriptor is present in the workstation.

In Cohen the I/O handlers are invoked to display objects. A video handler program displays a CD video on the display (12:22-23; Fig. 7a, step 526), a graphics handler program generates graphics which are displayed on the display 208 (14:5-10). There is no teaching relating the location of a display area.

Accordingly, the claimed display area is not expressly found in Cohen. [Felten at paragraph 57].

b. The claimed display area is not inherent in Cohen.

As set forth above, the claimed display area is not expressly found in Cohen. No extrinsic evidence exists that makes it clear that the missing display area is necessarily present in the Cohen system and would be so recognized by persons of ordinary skill in the art.

Displaying within the browser-controlled window is not inherent in Cohen either. In Cohen, displaying does not necessarily take place within the browser-controlled window. The disclosure in Cohen is consistent with the display taking place in a separate window, as with Mosaic helper applications, or on a separate device. [Felten at paragraph 60]

c. The claimed display window is not enabled by Mosaic.

Mosaic helper applications are displayed in a separate window. When a helper application is launched, a new window is created, and the helper application uses this new window exclusively. The new window used by the helper application is entirely separate from any window used or controlled by the browser. [Felten at paragraph 33]

Accordingly, Mosaic does not show that the claimed display window was in the public's possession before applicant's invention.

4. **The claimed element of type information, associated with the object, utilized by the browser to identify and locate an executable application external to the first distributed hypermedia document is not explicitly found or inherently described in Cohen.**

a. The claimed type information is not explicitly found in Cohen.

Claim 6 recites:

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.

With regard to this claimed feature, at page 26 of the office action the examiner quotes the part of claim 6 set forth above followed by citations to the Cohen reference.

With regard to the claimed type information, the examiner cites Fig. 1a of Cohen, whereby the external executable application is specified as "DATA = 'graph.exe\GOCA FORMA C'", seen in Fig. 1a, wherein the graph.exe program is external to the hypermedia document.

This rejection is respectfully traversed for the following reasons.

None of the citations at page 26 of the office action point to a part of Cohen where the claimed type information is expressly found.

If the DATA = feature of Cohen is equated to the claimed type information and the graph.exe is equated to the executable application then the claimed feature that the type information is utilized by the browser to identify and locate the executable application is not explicitly found in Cohen. [Felten at paragraph 63]

In Cohen, the book reader (which is equated to the browser) does not utilize the "type information" to identify and locate anything. All the book reader does with this information is to pass it on, unexamined, to the operating system, which invokes the application. The book reader does not have any kind of algorithm or procedure that it follows to identify and locate an application to be used. [Felten at paragraph 64]

Accordingly, the claimed type information is not expressly found in Cohen. [Felten at paragraph 63].

b. The claimed type information is not inherent in Cohen.

As set forth above, the claimed type information is not expressly found in Cohen. No extrinsic evidence exists that makes it clear that the missing type information is necessarily present in the Cohen system and would be so recognized by persons of ordinary skill in the art.

The person of ordinary skill would recognize that the Cohen technique is consistent with the operation a softcopy book reading program. The program design is consistent with allowing authors to cause a program of choice to be invoked on the users computer. [Felten at paragraph

65] Accordingly, the feature of the type information utilized by the browser to identify and locate an external application is not necessarily present in Cohen.

Additionally, the design of Cohen is inconsistent with the use of type information recited in claim 6. The identify and locate step, and the fact that that step is done by the browser, is an important aspect of the claimed '906 invention. For example, this step provides an important security protection. Users often want to display distributed hypermedia documents that come from untrusted sources, such as Web pages that come from arbitrary sites. If the author of such a site can cause an executable application of his choice to be invoked on the user's system, then the site author can use that application to gain access to the user's private files or modify the state of the user's computer, for example to install spyware or a virus. [Felten at paragraph 65]

Having the browser – a program trusted by the user – identify and locate the executable application lets the browser protect the user from this danger. A properly written browser will only allow trusted applications to be run, thereby protecting the user against security problems. A hostile site author cannot run a malicious application on the user's computer, because it is the browser, not the site author, that is identifying and locating the application that will be run. References in which the browser does not utilize type information to identify and locate the executable application lack this protection. [Felten at paragraph 66]

c. The claimed type information is not enabled by Mosaic.

The claimed feature of type information utilized by the browser to identify and locate an external application is not enabled by Mosaic. Cohen explicitly discloses an alternative technique of identifying and locating an external application.

Claim 1

Claim 1 recites a method including the same limitations argued above with respect to the patentability of claim 6. Accordingly, claim 1 is not anticipated by Cohen as enabled by Mosaic.

Dependent Claims

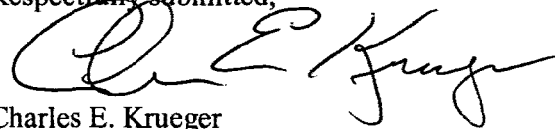
Claims 2-5 depend on claim 1 and are thus allowable for the same reasons as claim 1. Claims 7-10 depend on claim 6 and are thus allowable for the same reasons as claim 6.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Reexamination are not anticipated by the cited references. The issuance of a formal Notice of Intent to Issue Reexamination Certificate (NIRC) at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this Reexamination, 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

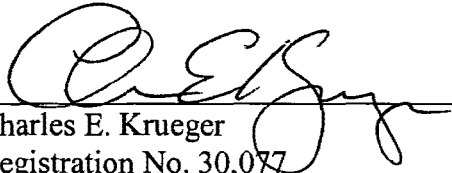
REX Control No. 90/007,858
Attorney Docket No. 006-1-5

CERTIFICATE OF SERVICE

I hereby certify that on this date I caused a true and correct copy of the attached Response After Non-Final Rejection to be served via first class mail on the following:

Stephen A. Wight
Klarquist Sparkman, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon St.
Portland, OR 97204

9/27/2007
Date


Charles E. Krueger
Registration No. 30,077

Law Office of Charles E. Krueger
P.O. Box 5607
Walnut Creek, CA 94549
(925) 944-3320

858 PH Ex. 6

Application of: Doyle et al.

Application Num.: 90/007,858

Filed: December 22, 2005

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

Examiner: Pokrzywa, J.R.

Art unit: 3992

Declaration of Edward W. Felten

I, Edward W. Felten, declare as follows:

1. I have been retained by Eolas and the Regents of the University of California to serve as an expert in the field of computer science and Internet software. My Curriculum Vitae, which recites my technical expertise, is attached hereto to as Exhibit A.

I. Qualifications

2. I graduated with Honors from the California Institute of Technology in 1985, with a B.S. degree in Physics. I received an M.S. in Computer Science in 1991, and a Ph.D. in Computer Science in 1993, both from the University of Washington.
3. I am currently a Professor of Computer Science and Public Affairs at Princeton University, where I have taught since 1993. I was originally hired at Princeton as an Assistant Professor of Computer Science, in 1993. I was promoted to Associate Professor of Computer Science in 1999, and to Professor of Computer Science in 2003. I was appointed as Professor of Computer Science and Public Affairs in 2006.
4. I am the author or co-author of numerous publications relating to computer science and Internet software. These publications are listed in my CV.
5. I have been asked to address the arguments presented in the Office Action mailed July 30, 2007 ("the Office Action") in connection with the reexamination of United States Patent No. 5,838,906 ("the '906 Patent") that the claims of the '906 Patent are unpatentable as being "anticipated" by U.S. Patent No. 5,367,621 ("Cohen"), issued to Cohen. For the reasons described in this declaration, I respectfully disagree with the arguments presented in the Office Action and, instead, believe that the claims of the '906 Patent fully meet the requirements for patentability over the cited references, as those patentability arguments have been described to me.

6. To familiarize myself with the issues involved in the rejection of the claims, I have reviewed numerous documents, including the following: the '906 Patent and its file history, the documents cited in the Office Action, and all other documents referenced or cited in this declaration.
7. My discussion is also based on my experience as a computer science researcher and teacher, and as a Web user and network software developer. From this experience, I have gained an independent understanding of how the browser art developed.

II. Response to the Unpatentability Arguments Raised in the Office Action

8. I have been told by patent counsel for Eolas and the Regents that a patent may not be obtained if the patent's subject matter is anticipated by a prior art reference. I have further been told that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. I have further been told that to serve as an anticipation when a reference is silent about the asserted inherent characteristic, such gap in the reference may be filled with recourse to extrinsic evidence, and that such evidence must make it clear that the missing descriptive matter is necessarily present in the thing described in the reference and that it would be so recognized by persons of ordinary skill in the art.
9. In reading the claims of the '906 Patent, I am using the claim construction established by Judge Zagel and upheld by the Federal Circuit in the *Eolas v. Microsoft* litigation.

1. The Level of Skill in the Art

10. My benchmark for what ordinary skill in the art means 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 an average student – or a person who has gained an equivalent level of knowledge through experience in the industry. This person knows how to do things in conventional ways but does not exhibit an unusual level of innovative thinking.
11. In 1994, those of ordinary skill in the art were just becoming familiar with the Web and Web browsers. One of ordinary skill would have had a general idea of how the Mosaic browser worked, and would have been familiar with hyperlinks, forms, and helper applications.

2. The '906 Patent

12. The claims of the '906 Patent describe a technology that allows web page authors to include, within the boundaries of a web page, interactive objects. This is done (briefly stated) by including in the web page's HTML text an embed text format, that provides information about where to get the object's data, along with information to identify and locate an executable application that will be invoked on the client computer to display the data and to provide interactivity with it, and by providing a web browser that knows how to parse the HTML to extract the embed text format, how to use type information to identify and locate the

executable application, how to invoke the executable application, to execute on the client computer, and how to interface to the executable application so as to allow the user to interact with it within the boundaries of the browser window.

13. My analysis of patentability will focus on Claim 6 of the '906 Patent. Claim 1 is very similar to Claim 6, the only material difference being that Claim 6 covers a computer program product and Claim 1 covers a method for using such a product. Accordingly, all of my analysis of Claim 6 applies to Claim 1 as well. Patentability of the other dependent claims discussed in the Office Action follows from the patentability of the underlying independent Claims 1 and 6.

3. Prior Art Browsers, Including Mosaic

14. The Office Action cites the applicants' admitted prior art. I have reviewed all prior art references referenced in the '906 Patent's file history. The Office Action's discussion of this prior art focuses on the Mosaic browser, which was the most advanced prior art browser.
15. In 1994, the Web was young, and browsers were a relatively new technology. Browsers offered only a very limited form of interactivity. A page could contain hyperlinks, on which the user could click to view another page. A page could be a form to be filled out by the user, with a "submit" button which, when clicked, caused the user to see another page.
16. Another technology, known as "helper applications," was implemented in the Mosaic browser. This technology allowed the browser to link to an external program, in cases where the browser encountered a file whose format the browser did not understand. For example, if the user clicked on a hyperlink that pointed to a file in .mpeg format (i.e., a movie in MPEG format), then the browser would launch an external MPEG-viewer program and pass the .mpeg file to that program. The result would be that the MPEG program ran, in a separate window from the browser.
17. Helper applications allowed the browser to link to an external program, but that program could not provide interactivity within the browser window. The helper application was just an external program that ran on the same computer, in a separate window.
18. None of these methods allowed a Web page author to place fully interactive objects within the confines of a Web page's display.
19. These methods are all implemented in today's browsers, and they are all in use on the Web today.

A. '906 Claim Elements Lacking in Mosaic

20. Mosaic lacked at least three elements required by the claims of the '906 Patent. Among the required elements lacking in Mosaic were an embed text format, automatic invocation, and display of the object within the browser window.

i. Mosaic Lacked the Required Embed Text Format

21. Mosaic lacked the required embed text format of the '906 claims. Instead, Mosaic used an ordinary hyperlink to link to any data that was to be displayed with a helper application.
22. Claim 6 requires that "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..." ('906 Patent at 18:14-18)
23. Hyperlinks in Mosaic were specified using an "A" (short for "Anchor") tag. For example, the HTML element

```
<a href="http://example.com/page.html">link</a>
```

would cause the text "link" to be displayed, typically in underlined blue type. If the user clicked on the underlined word "link", the browser would follow the hyperlink and navigate to the URL "http://example.com/page.html".

24. Similarly, the HTML element

```
<a href="http://example.com/video.mpg">video</a>
```

would cause the text "video" to be displayed, in underlined blue type. If the user clicked on the underlined word "photo", the browser would download the file at the URL "http://example.com/video.mpg" and launch a helper application to display it.

25. In the claimed '906 system, the browser instead used a special tag, the "embed text format", to specify that an embedded object should be included. Mosaic lacked the embed text format. The use of an embed text format was a significant improvement over the prior art Mosaic browser, as it allowed the browser to recognize immediately that an embedded object was present and special processing was needed.
26. The Examiner states that Mosaic lacks the embed text format: "the NCSA Mosaic references ... 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...**" (Office Action at paragraph 14, emphasis in original)

ii. Mosaic Lacked the Required Automatic Invocation

27. Mosaic further lacked the required automatic invocation of the '906 claims.
28. Claim 6 includes the limitation "wherein said embed text format is parsed by said browser to automatically invoke said executable application" ('906 Patent at 18:23-25), which requires that the executable application be invoked automatically, that is, without requiring any action such as a mouse click from the user.

29. In Mosaic, as stated above, a helper application was launched via an ordinary hyperlink. All hyperlinks required a mouse click by the user to be activated. To launch a helper application, the user had to make such a mouse click. Therefore, the launching of helper applications was not automatic.
30. Automatic invocation was an important improvement in the claimed '906 technology over the prior art Mosaic browser. Automatic invocation allowed the object to appear immediately when the user visited the enclosing web page, thus helping to make the object appear to be an integral part of the web page.

iii. Mosaic Lacked the Required Display within the Browser Window

31. Mosaic further lacked the required '906 claim element of displaying the object within the browser window.
32. Claim 6 of the '906 Patent includes the limitation "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." ('906 Patent at 18:26-30) This requires the object to be displayed, and interactive processing of the object to be enabled, within an area in the browser window.
33. Mosaic helper applications displayed in a separate window. When a helper application was launched, a new window was created, and the helper application used this new window exclusively. The new window used by the helper application was entirely separate from any window used or controlled by the browser.
34. The ability of the claimed '906 technology to display and enable interactive processing within the browser window was a significant advance over the prior art Mosaic browser. Enabling display and interactivity within the browser window allowed the object to appear, seamlessly, as an integral part of the web page's display. This was a more natural model for users, allowing the object to be placed in the context of any descriptive text appearing on the web page.

4. The Cohen Reference

35. Cohen describes a system that allows the author of a softcopy book to specify specific multimedia presentations accompanying the text of the book.

B. Cohen Does Not Anticipate the '906 Claims

36. The Examiner argues that the claims 1-3 and 6-8 of the '906 Patent are anticipated by Cohen. I respectfully disagree with this conclusion, for the reasons given below.
37. As discussed above, my analysis will focus on Claim 6 of the '906 Patent. My analysis of Claim 6 also applies to Claim 1. Non-anticipation of the dependent Claims 2, 3, 7, and 8 follows from non-anticipation of the underlying independent Claims 1 and 6.

38. Several elements of Claim 6 are absent in Cohen. For example, Cohen lacks the elements of interactive processing, an embed text format, display of the object within a browser window, and the use of type information by a browser to identify and locate an executable application.

C. Cohen Lacks the Required Interactive Processing

39. Claim 6 of the '906 Patent requires that the browser "automatically invoke said executable application to execute on said client workstation in order to display said object and **enable interactive processing of said object ...**" ('906 Patent at 18:24-27, emphasis added) Cohen lacks the element of interactive processing.
40. Claim 6 requires that the system not only "display said object" but also "enable interactive processing" of the object. To display the object is to make it visible, e.g. by sending it to an output device. Interactive processing means that the user, by using the mouse or keyboard or similar input device, can change the structure or presentation of the object. The interactive processing must occur within the "display area" where the object is displayed.
41. Cohen does not expressly disclose interactive processing of the object. Cohen repeatedly discusses displaying the object, but nowhere is this display coupled to interactive processing of the object. See, e.g. Cohen at 2:44-47 ("transferring ... to the multimedia output device, the multimedia data from the multimedia object"), 2:59-60 ("to support the presentation of the multimedia object"), 8:31-35 (describing how an animation object is "displayed"), 8:46-56 ("displayed graphics"), 9:46-10:4 ("to transfer the audio data from the audio object to the stereo high fi to produce the audio presentation to output audio data from the audio object stored thereon to the stereo high fi for the audio presentation"), 12:4-23 ("playing of the video information from the compact disk player ... and the presentation of the resulting motion picture on the display 208 at the workstation"), 13:7-9 ("the animation for the ALT_OBJECT is displayed"), 13:21-51 ("Audio data form [sic] the object is output to the audio support."), 13:52-14:10 ("The data in the object 110 is output to the appropriate graphic support software and the graphics is generated ... as the graphics 190' which is displayed on the display 208.") Nowhere does Cohen discuss **interactive processing** of the object within a browser window.
42. The Examiner does not point to any support in Cohen for the element of interactive processing. The Examiner discusses, as a single unit, a longer portion of the claim (Office Action at p. 32), and the Examiner cites support in Cohen for some aspects of this longer portion of the claim. But none of the material cited by the Examiner supports the notion that Cohen discloses **interactive processing** of an object within a browser-controlled window.
43. Nor is interactive processing inherent in Cohen. Displaying an object does not necessarily require that the user can interact with the object.
44. The design of Cohen was incompatible with interactive processing. In Cohen, when the main book reader program invokes an external program, it waits for the external program to finish before resuming the main book reader program. While

the external program is running, the main book reader program is frozen and cannot accept input. This is evident from Fig. 6 of Cohen and the accompanying description in (e.g.) Columns 10 and 11. For example:

After the completion of the I/O handler program whose execution was started as a result of step 426 of the flow diagram of FIG. 6, the flow proceeds back to step 420 to wait for another mouse pointer to activate a link tag or alternatively to wait until a new page is requested by the user.

(Cohen at 11:22-27) Notably, it is only after the completion of the external program that the main book reader program can accept input from the user. If the external program had tried to provide interactive processing, this would have required the external program to continue running (waiting for user interaction) which would have frozen the main book reader program, giving the user no way to switch to a different page of the e-book. The role of the external program in Cohen is to display the object and then stop running.

45. By contrast, in the invention defined in Claim 6 of the '906 Patent, the browser and the external program can both be active and accepting input at the same time, so the user has the option of seamlessly interacting with the object or navigating the browser.
46. One of the main achievements of the '906 inventors was to provide interaction with an object, seamlessly within a Web page. A system like Cohen, which can display objects but cannot provide interactive processing, is missing an important element of the invention defined in Claim 6 of the '906 Patent.

D. Cohen Lacks the Required Embed Text Format

47. Claim 6 of the '906 Patent requires that "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, ..., and wherein said embed text format is parsed by said browser to automatically invoke said executable application .. 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 the first distributed hypermedia document ..." ('906 Patent at 18:14-29)
48. The use of an embed text format is an important element of the invention defined in Claim 6 of the '906 Patent. One drawback of many prior art browsers, such as Mosaic, is that they lacked an embed text format. For example, the examiner states that "the NCSA Mosaic references ... 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...**" (Office Action at paragraph 14, emphasis in original)
49. Cohen does not disclose the use of an embed text format.

50. The Examiner states that the “link description tags LDESC” of Cohen are the embed text format (Office Action at p. 31). For the reasons described below, I respectfully disagree with this conclusion.
51. The LDESC tags cannot be the embed text format, because they do not satisfy the required claim element “wherein said first distributed hypermedia document includes an **embed text format, located at a first location** in said first distributed hypermedia document ... to display said object and enable interactive processing of said object **within a display area created at said first location ...**” (‘906 Patent at 18:14-28, emphasis added) This claim element requires that the embedded object be displayed at a location in the distributed hypermedia document (e.g., the Web page) that corresponds to the location of the embed text format within the document.
52. The LDESC tag does not appear in the document at the required location. Instead, the LDESC (link description) tag appears in the document file’s prologue. “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 ...**” (Cohen at 5:8-11, emphasis added) The figures in Cohen also depict the LDESC tags as being in the file’s prologue rather than in the book text. For example, Figure 1 depicts the structure of the document file. The “link description tags 102” (also depicted in Figure 1a) are distinct from the “book text with tags 104” (which is also depicted in Figure 1b). (See also Figs. 4, 8a, and 8b.)
53. The fact that the LDESC tag does not appear at a location in the book text is one reason why the LDESC tag cannot be the embed text format of the ‘906 claims.
54. The “link tag :L” of Cohen does appear in the book text, but it cannot be the embed text format either, because (e.g.) it lacks the required claim element of an “embed text format ... that specifies the location of at least a portion of an object external to the first distributed hypermedia document ...” (‘906 Patent at 18:15-18). The link tag of Cohen does not specify the location of an object, nor does it specify the location of anything that is external to the first distributed hypermedia document. This is one reason why the “link tag :L” of Cohen cannot be the embed text format.
55. Cohen’s design strategy, of having a small, simple link tag that refers to a larger, more detailed link description in the document prologue, makes sense given the problem that Cohen was trying to solve. Cohen was designed for use with electronic books. These books, unlike Web pages, are large, multi-page files that often repeat graphic elements on different pages. By separating the link tag and link description, Cohen allowed an element to be repeated without having to repeat the full link description each time. Instead, there could be a single link description in the document prologue, and one small link tag at each place in the document where the object was to be used. The claimed ‘906 design, by contrast, is better suited for use on the Web where individual pages are provided separately. For at least this reason, the use of a single tag is not expressly found or inherently described in Cohen.

56. Another reason that Cohen and Mosaic do not anticipate Claim 6 is that neither Cohen nor Mosaic had an embed text format.

E. Cohen Lacks the Required Display within the Browser Window

57. Claim 6 of the '906 Patent requires that the display of the object take place "within a display area ... within said first browser-controlled window" ('906 Patent at 27-30). Cohen lacks the required element of displaying the object within the browser-controlled window.
58. Cohen discloses displaying the object on a screen, but there is no disclosure of the display happening within an area of the browser-controlled window.
59. The Examiner does not point to any support in Cohen for the element of displaying within the browser-controlled window. The Examiner discusses, as a single unit, a longer portion of the claim (Office Action at p. 32), and the Examiner cites support in Cohen for some aspects of this longer portion of the claim. But none of the material cited by the Examiner supports the notion that Cohen discloses displaying of the object within a display area in the browser-controlled window.
60. Displaying within the browser-controlled window is not inherent in Cohen either. In Cohen, displaying does not necessarily take place within the browser-controlled window. The disclosure in Cohen is consistent with the display taking place in a separate window, as with Mosaic helper applications, or on a separate device.
61. One reason that Claim 6 is not anticipated by Cohen and Mosaic is that both Cohen and Mosaic lack the require element of displaying the object within a display area in the browser-controlled window.

F. Cohen Lacks the Required Use of Type Information by the Browser to Identify and Locate an Executable Application

62. Claim 6 of the '906 Patent requires, "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..." ('906 Patent at 18:19-22). Cohen lacks this element.
63. The Examiner's only discussion of this element of Claim 6 in relation to Cohen is as follows:

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

(Office Action at pp. 31-32) In this discussion, the Examiner is apparently reading the "DATA = ..." line as being the type information, and graph.exe as being the executable application. However, the Examiner does not point to any disclosure of a browser utilizing the type information to identify and locate the executable application. Nor does Fig. 1a of Cohen depict any steps performed by the browser.

64. In Cohen, the book reader (which the Examiner equates to a browser) does not utilize the "type information" to identify and locate anything. All the book reader does with this information is to pass it on, unexamined, to the operating system, which invokes the application. The book reader does not have any kind of algorithm or procedure that it follows to identify and locate an application to be used.
65. The identify and locate step, and the fact that that step is done by the browser, is an important aspect of the invention defined in Claim 6 of the '906 Patent. For example, this step provides an important security protection. Users often want to display distributed hypermedia documents that come from untrusted sources, such as Web pages that come from arbitrary sites. If the author of such a site can cause an executable application of his choice to be invoked on the user's system, then the site author can use that application to gain access to the user's private files or modify the state of the user's computer, for example to install spyware or a virus.
66. Having the browser – a program trusted by the user – identify and locate the executable application lets the browser protect the user from this danger. A properly written browser will only allow trusted applications to be run, thereby protecting the user against security problems. A hostile site author cannot run a malicious application on the user's computer, because it is the browser, not the site author, that is identifying and locating the application that will be run. References in which the browser does not utilize type information to identify and locate the executable application lack this protection.
67. The importance of this claim element was reinforced by the *Eolas v. Microsoft* litigation, in which the question of which references had this claim element was an important one.

III. Conclusion

68. For the reasons given above, I conclude that the rejection of claims 1 and 6 as being unpatentable is incorrect. The claims of the '906 patent were not anticipated by the references cited in the Office Action.

I declare that all statements made herein of my knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both under 18 U.S.C. Section 1001, and that such willful false statements may jeopardize the validity of the patent.

Dated: September 27, 2007



Edward W. Felten

Edward W. Felten

Dept. of Computer Science
Princeton University
35 Olden Street
Princeton NJ 08540
(609) 258-5906
(609) 258-1771 fax
felten@cs.princeton.edu

Education

Ph.D. in Computer Science and Engineering, University of Washington, 1993.
Dissertation title: "Protocol Compilation: High-Performance Communication for Parallel Programs." Advisors: Edward D. Lazowska and John Zahorjan.
M.S. in Computer Science and Engineering, University of Washington, 1991.
B.S. in Physics, with Honors, California Institute of Technology, 1985.

Employment

Professor of Computer Science, Princeton University, 2003-present.
Associate Professor of Computer Science, Princeton University, 1999-2003.
Assistant Professor of Computer Science, Princeton University, 1993-99.
Senior Computing Analyst, Caltech Concurrent Computing Project, California Institute of Technology, 1986-1989.

Director, Secure Internet Programming Laboratory, Dept. of Computer Science,
Princeton University, 1996-present.

U.S. Dept. of Justice, Antitrust Division: consulting and testimony in Microsoft antitrust case, 1998-2002.

Robins Kaplan, Miller & Ciresi. Consulting and testimony in patent lawsuit, 1998-2003.

Keker & Van Nest. Consulting in intellectual property / free speech lawsuit, 2002.

Electronic Frontier Foundation. Consulting in intellectual property / free speech lawsuits, 2001-present.

Certus Ltd.: consultant in product design and analysis, 2000-2002.

Cigital Inc.: Technical Advisory Board member, 2000-present.

Cloakware Ltd.: Technical Advisory Board member, 2000-present.

Propel.com: Technical Advisory Board member, 2000-2002.

NetCertainty.com: Technical Advisory Board member, 1999-2002.

FullComm LLC: Scientific Advisory Board member, 1999-2001.

Sun Microsystems: Java Security Advisory Board member, 1997-present.

Finjan Software: Technical Advisory Board member, 1997-2002.

International Creative Technologies: consultant in product design and analysis, 1997-98.
Bell Communications Research: consultant in computer security research, 1996-97.

Honors and Awards

Scientific America 50 Award, 2003.
Alfred P. Sloan Fellowship, 1997.
Emerson Electric, E. Lawrence Keyes Faculty Advancement Award, Princeton University School of Engineering, 1996.
NSF National Young Investigator award, 1994.
Outstanding Paper award, 1997 Symposium on Operating Systems Principles.
Best Paper award, 1995 ACM SIGMETRICS Conference.
AT&T Ph.D. Fellowship, 1991-93.
Mercury Seven Foundation Fellowship, 1991-93.

Research Interests

Computer security, especially relating to consumer products. Technology law and policy. Internet software. Operating systems. Interaction of security with programming languages and operating systems. Distributed computing. Parallel computing architecture and software.

Professional Service

Professional Societies and Advisory Groups

ACM Advisory Committee on Security and Privacy, 2002-2003.
DARPA Information Science and Technology (ISAT) advisory group, 2002-present.
Co-chair, ISAT study committee on "Reconciling Security with Privacy," 2001-2002.
National Academy study committee on Foundations of Computer Science, 2001-present.

Program Committees

USENIX General Conference, 2004.
Workshop on Foundations of Computer Security, 2003.
ACM Workshop on Digital Rights Management, 2001.
ACM Conference on Computer and Communications Security, 2001.
ACM Conference on Electronic Commerce, 2001.
Workshop on Security and Privacy in Digital Rights Management, 2001.
Internet Society Symposium on Network and Distributed System Security, 2001.
IEEE Symposium on Security and Privacy, 2000.
USENIX Technical Conference, 2000.
USENIX Windows Systems Conference, 2000.
Internet Society Symposium on Network and Distributed System Security, 2000.

IEEE Symposium on Security and Privacy, 1998.
ACM Conference on Computer and Communications Security, 1998.
USENIX Security Symposium, 1998.
USENIX Technical Conference, 1998.
Symposium on Operating Systems Design and Implementation, 1996.

Corporate Advisory Boards

Sun Microsystems, Java Security Advisory Council.
Digital Inc.: Technical Advisory Board.
Cloakware Ltd.: Technical Advisory Board.
Propel.com: Technical Advisory Board.
Finjan Software: Technical Advisory Board.
Netcertainty: Technical Advisory Board.
FullComm LLC: Scientific Advisory Board.

University and Departmental Service

Faculty Advisory Committee on Policy, 2002-present.
Council of the Princeton University Community, 2002-present (Executive Committee)
Faculty Advisory Committee on Athletics, 1998-2000.
Computer Science Academic Advisor, B.S.E. program, class of 1998 (approx. 25 students)
Faculty-Student Committee on Discipline, 1996-98.
Faculty-Student Committee on Discipline, Subcommittee on Sexual Assault and Harrassment, 1996-98.

Students Advised

Ph.D. Advisees:

Minwen Ji (Ph.D. 2001). Dissertation: Data Distribution for Dynamic Web Content.
Researcher at Compaq Systems Research Center.
Dirk Balfanz (Ph.D. 2000). Dissertation: Access Control for Ad Hoc Collaboration.
Researcher at Xerox Palo Alto Research Center.
Dan S. Wallach (Ph.D. 1998). Dissertation: A New Approach to Mobile Code Security.
Assistant Professor of Computer Science, Rice University.
Robert A. Shillner (Ph.D. expected 2004). Tentative dissertation title: Improving Distributed File Systems using a Shared Logical Disk. Technical staff member at Google.
Michael Schneider (Ph.D. expected 2003). Dissertation topic: Network Defenses against Denial of Service Attacks.

Significant Advisory Role:

Drew Dean (Ph.D. 1998). Advisor: Andrew Appel. Researcher at SRI International.
Stefanos Damianakis, Ph.D. 1998. Advisor: Kai Li. President, Netrics, Inc.
Pei Cao, Ph.D. 1996. Advisor: Kai Li. Assistant Professor of Computer Sciences, University of Wisconsin. On leave at Cisco Systems.

Lujo Bauer, Ph.D. 2003. Advisor: Andrew Appel. Postdoctoral researcher at Carnegie-Mellon University.

Publications

Books and Book Chapters

- [1] Freedom to Tinker. Edward W. Felten. Publication expected, 2004.
- [2] Securing Java: Getting Down to Business with Mobile Code. Gary McGraw and Edward W. Felten. John Wiley and Sons, New York 1999.
- [3] Java Security: Web Browsers and Beyond. Drew Dean, Edward W. Felten, Dan S. Wallach, and Dirk Balfanz. In "Internet Besieged: Countering Cyberspace Scofflaws," Dorothy E. Denning and Peter J. Denning, eds. ACM Press, New York, 1997.
- [4] Java Security: Hostile Applets, Holes and Antidotes. Gary McGraw and Edward Felten. John Wiley and Sons, New York, 1996.
- [5] Dynamic Tree Searching. Steve W. Otto and Edward W. Felten. In "High Performance Computing", Gary W. Sabot, ed., Addison Wesley, 1995.

Journal Articles

- [6] Mechanisms for Secure Modular Programming in Java. Software – Practice and Experience, 33:461-480, 2003.
- [7] The Digital Millennium Copyright Act and its Legacy: A View from the Trenches. Illinois Journal of Law, Technology and Policy, Fall 2002.
- [8] DRM and Fair Use: A Skeptical View. Edward W. Felten. Communications of the ACM. April, 2003.
- [9] The Security Architecture Formerly Known as Stack Inspection: A Security Mechanism for Language-based Systems. Dan S. Wallach, Edward W. Felten, and Andrew W. Appel. ACM Transactions on Software Engineering and Methodology, 9:4, October 2000.
- [10] Statically Scanning Java Code: Finding Security Vulnerabilities. John Viega, Tom Mutdosch, Gary McGraw, and Edward W. Felten. IEEE Software, 17(5), Sept./Oct. 2000.
- [11] Client-Server Computing on the SHRIMP Multicomputer. Stefanos N. Damianakis, Angelos Bilas, Cezary Dubnicki, and Edward W. Felten. IEEE Micro 17(1):8-18, February 1997.
- [12] Fast RPC on the SHRIMP Virtual Memory Mapped Network Interface. Angelos Bilas and Edward W. Felten. IEEE Transactions on Parallel and Distributed Computing, February 1997.

- [13] Implementation and Performance of Integrated Application-Controlled File Caching, Prefetching and Disk Scheduling. Pei Cao, Edward W. Felten, Anna R. Karlin, and Kai Li. ACM Transactions on Computer Systems, Nov 1996.
- [14] Virtual Memory Mapped Network Interface Designs. Matthias A. Blumrich, Cezary Dubnicki, Edward W. Felten, Kai Li, and Malena Mesarina. IEEE Micro, 15(1):21-28, February 1995.

Symposium Articles

- [15] Receiver Anonymity via Incomparable Public Keys. Brent R. Waters and Edward W. Felten. ACM Conference on Computer and Communications Security. November 2003.
- [16] Attacking an Obfuscated Cipher by Injecting Faults. Matthias Jacob, Dan Boneh, and Edward W. Felten. ACM Workshop on Digital Rights Management, November 2002.
- [17] A General and Flexible Access-Control System for the Web. Lujo Bauer, Michael A. Schneider, and Edward W. Felten. 11th USENIX Security Symposium, August 2002.
- [18] Informed Consent in the Mozilla Browser: Implementing Value-Sensitive Design. Batya Friedman, Daniel C. Howe, and Edward W. Felten. Hawaii International Conference on System Sciences, January 2002. (Best Paper award, organizational systems track.)
- [19] Reading Between the Lines: Lessons from the SDMI Challenge. Scott A. Craver, John P. McGregor, Min Wu, Bede Liu, Adam Stubblefield, Ben Swartzlander, Dan S. Wallach, Drew Dean, and Edward W. Felten. USENIX Security Symposium, August 2001.
- [20] Cookies and Web Browser Design: Toward Realizing Informed Consent Online. Lynette I. Millett, Batya Friedman, and Edward W. Felten. Proc. of CHI 2001 Conference on Human Factors in Computing Systems, April 2001.
- [21] Timing Attacks on Web Privacy. Edward W. Felten and Michael A. Schneider. Proc. of 7th ACM Conference on Computer and Communications Security, Nov. 2000.
- [22] Archipelago: An Island-Based File System for Highly Available and Scalable Internet Services. USENIX Windows Systems Symposium, August 2000.
- [23] Proof-Carrying Authentication. Andrew W. Appel and Edward W. Felten. Proc. of 6th ACM Conference on Computer and Communications Security, Nov. 1999.
- [24] An Empirical Study of the SHRIMP System. Matthias A. Blumrich, Richard D. Alpert, Yuqun Chen, Douglas W. Clark, Stefanos, N. Damianakis, Cezary Dubnicki, Edward W. Felten, Liviu Iftode, Margaret Martonosi, Robert A. Shillner, and Kai Li. Proc. of 25th International Symposium on Computer Architecture, June 1998.

- [25] Performance Measurements for Multithreaded Programs. Minwen Ji, Edward W. Felten, and Kai Li. Proc. of 1998 SIGMETRICS Conference, June 1998.
- [26] Understanding Java Stack Inspection. Dan S. Wallach and Edward W. Felten. Proc. of 1998 IEEE Symposium on Security and Privacy, May 1998.
- [27] Extensible Security Architectures for Java. Dan S. Wallach, Dirk Balfanz, Drew Dean, and Edward W. Felten. Proc. of 16th ACM Symposium on Operating Systems Principles, Oct. 1997. Outstanding Paper Award.
- [28] Web Spoofing: An Internet Con Game. Edward W. Felten, Dirk Balfanz, Drew Dean, and Dan S. Wallach. Proc. of 20th National Information Systems Security Conference, Oct. 1997.
- [29] Reducing Waiting Costs in User-Level Communication. Stefanos N. Damianakis, Yuqun Chen, and Edward W. Felten. Proc. of 11th Intl. Parallel Processing Symposium, April 1997.
- [30] Stream Sockets on SHRIMP. Stefanos N. Damianakis, Cezary Dubnicki, and Edward W. Felten. Proc. of 1st Intl. Workshop on Communication and Architectural Support for Network-Based Parallel Computing, February 1997. (Proceedings available as Lecture Notes in Computer Science #1199.)
- [31] Early Experience with Message-Passing on the SHRIMP Multicomputer. Richard D. Alpert, Angelos Bilas, Matthias A. Blumrich, Douglas W. Clark, Stefanos Damianakis, Cezary Dubnicki, Edward W. Felten, Liviu Iftode, and Kai Li. Proc. of 23rd Intl. Symposium on Computer Architecture, 1996.
- [32] A Trace-Driven Comparison of Algorithms for Parallel Prefetching and Caching. Tracy Kimbrel, Andrew Tomkins, R. Hugo Patterson, Brian N. Bershad, Pei Cao, Edward W. Felten, Garth A. Gibson, Anna R. Karlin, and Kai Li. Proc. of 1996 Symposium on Operating Systems Design and Implementation.
- [33] Java Security: From HotJava to Netscape and Beyond. Drew Dean, Edward W. Felten, and Dan S. Wallach. Proc. of 1996 IEEE Symposium on Security and Privacy.
- [34] Integrated Parallel Prefetching and Caching. Tracy Kimbrel, Pei Cao, Edward W. Felten, Anna R. Karlin, and Kai Li. Proc. of 1996 SIGMETRICS Conference.
- [35] Software Support for Virtual Memory-Mapped Communication. Cezary Dubnicki, Liviu Iftode, Edward W. Felten, and Kai Li. Proc. of Intl. Parallel Processing Symposium, April 1996.
- [36] Protected, User-Level DMA for the SHRIMP Network Interface. Matthias A. Blumrich, Cezary Dubnicki, Edward W. Felten, and Kai Li. Proc. of 2nd Intl. Symposium on High-Performance Computer Architecture, Feb. 1996
- [37] Improving Release-Consistent Shared Virtual Memory using Automatic Update. Liviu Iftode, Cezary Dubnicki, Edward W. Felten, and Kai Li. Proc. of 2nd Intl. Symposium on High-Performance Computer Architecture, Feb. 1996

- [38] Synchronization for a Multi-Port Frame Buffer on a Mesh-Connected Multicomputer. Bin Wei, Gordon Stoll, Douglas W. Clark, Edward W. Felten, and Kai Li. Parallel Rendering Symposium, Oct. 1995.
- [39] A Study of Integrated Prefetching and Caching Strategies. Pei Cao, Edward W. Felten, Anna R. Karlin, and Kai Li. Proc. of 1995 ACM SIGMETRICS Conference. Best Paper award.
- [40] Evaluating Multi-Port Frame Buffer Designs for a Mesh-Connected Multicomputer. Gordon Stoll, Bin Wei, Douglas W. Clark, Edward W. Felten, Kai Li, and Patrick Hanrahan. Proc. of 22nd Intl. Symposium on Computer Architecture.
- [41] Implementation and Performance of Application-Controlled File Caching. Pei Cao, Edward W. Felten, and Kai Li. Proc. of 1st Symposium on Operating Systems Design and Implementation, pages 165-178, November 1994.
- [42] Application-Controlled File Caching Policies. Pei Cao, Edward W. Felten, and Kai Li. Proc. of USENIX Summer 1994 Technical Conference, pages 171-182, 1994.
- [43] Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer. Matthias A. Blumrich, Kai Li, Richard D. Alpert, Cezary Dubnicki, Edward W. Felten, and Jonathan S. Sandberg. Proc. of Intl. Symposium on Computer Architecture, 1994.
- [44] Performance Issues in Non-Blocking Synchronization on Shared-Memory Multiprocessors. Juan Alemany and Edward W. Felten. Proceedings of Symposium on Principles of Distributed Computing, 1992.
- [45] Improving the Performance of Message-Passing Applications by Multithreading. Edward W. Felten and Dylan McNamee. Proceedings of Scalable High-Performance Computing Conference (SHPCC), 1992.
- [46] A Highly Parallel Chess Program. Edward W. Felten and Steve W. Otto. 1988 Conference on Fifth Generation Computer Systems.

Other Publications

- [47] Freedom to Tinker weblog, at <http://www.freedom-to-tinker.com>. Commentary on technology law and policy. Approximately 4000 readers per day.
- [48] Secure, Private Proofs of Location. Brent Waters and Edward W. Felten. Submitted for publication, January 2003.
- [49] An Efficient Heuristic for Defense Against Distributed Denial of Service Attacks using Route-Based Distributed Packet Filtering. Michael A. Schneider and Edward W. Felten. Submitted for publication, January 2003.
- [50] Written testimony to House Commerce Committee, Subcommittee on Courts, the Internet, and Intellectual Property, oversight hearing on "Piracy of Intellectual Property on Peer to Peer Networks." September 2002.

- [51] Written testimony to Senate Judiciary Committee hearings on "Competition, Innovation, and Public Policy in the Digital Age: Is the Marketplace Working to Protect Digital Creativity?" March 2002.
- [52] Informed Consent Online: A Conceptual Model and Design Principles. Batya Friedman, Edward W. Felten, and Lynette I. Millett. Technical Report 2000-12-2, Dept. of Computer Science and Engineering, University of Washington, Dec. 2000.
- [53] Mechanisms for Secure Modular Programming in Java. Lujjo Bauer, Andrew W. Appel, and Edward W. Felten. Technical Report CS-TR-603-99, Department of Computer Science, Princeton University, July 1999.
- [54] A Java Filter. Dirk Balfanz and Edward W. Felten. Technical Report 567-97, Dept. of Computer Science, Princeton University, October 1997.
- [55] Inside RISKS: Webware Security. Edward W. Felten. Communications of the ACM, 40(4):130, 1997.
- [56] Simplifying Distributed File Systems Using a Shared Logical Disk. Robert A. Shillner and Edward W. Felten. Princeton University technical report TR-524-96.
- [57] Contention and Queueing in an Experimental Multicomputer: Analytical and Simulation-based Results. Wenjia Fang, Edward W. Felten, and Margaret Martonosi. Princeton University technical report TR-508-96.
- [58] Design and Implementation of NX Message Passing Using SHRIMP Virtual Memory Mapped Communication. Richard D. Alpert, Cezary Dubnicki, Edward W. Felten, and Kai Li. Princeton University technical report TR-507-96.
- [59] Protocol Compilation: High-Performance Communication for Parallel Programs. Edward W. Felten. Ph.D. dissertation, Dept. of Computer Science and Engineering, University of Washington, August 1993.
- [60] Building Counting Networks from Larger Balancers. Edward W. Felten, Anthony LaMarca, and Richard Ladner. Univ. of Washington technical report UW-CSE-93-04-09.
- [61] The Case for Application-Specific Communication Protocols. Edward W. Felten. Univ. of Washington technical report TR-92-03-11.
- [62] A Centralized Token-Based Algorithm for Distributed Mutual Exclusion. Edward W. Felten and Michael Rabinovich. Univ. of Washington technical report TR-92-02-02.
- [63] Issues in the Implementation of a Remote Memory Paging System. Edward W. Felten and John Zahorjan. Univ. of Washington technical report TR-91-03-09.

858 PH Ex. 7

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Reexamination Control No:

90/007,858

Filing Date: 12/22/2005

Patent No: 5,838,906

Examiner: POKRZYWA, JOSEPH R

Art Unit: 3992

DECLARATION OF MICHAEL D.
DOYLE UNDER 37 CFR 1.131

Commissioner for Patents
Alexandria, VA 22313-1450

I, MICHAEL D. DOYLE, hereby declare that:

1. I am a co-inventor of the subject matter disclosed and claimed in U.S. Patent No. 5,838,906.
2. Photocopies of drawings and records showing facts such, in character and weight, establish reduction to practice prior to the effective date of the Viola reference form part of this declaration and are attached hereto. These attachments include:
 - A. Digital Libraries Proposal
 - B. Medicine Meets Virtual Reality II Abstract and Transcript
 - C. Paper Submission for Visualization 94 "Integrated Control of Distributed Volume Distribution Through the World-Wide-Web"
 - D. Hardin/Smarr emails
 - E. Hardin Letter
 - F. Stereo-Pair Email
 - G. SIGWEB Meeting Announcement

3. As set forth below, the subject matter claimed in the above patent was reduced to practice in this country prior to August 16, 1994, the effective date of the reference entitled "A Brief Overview of the VIOLA Engine, and its applications, which include the "viola in a Nutshell: the Viola World Wide Web Toolkit" (hereinafter "the Viola reference") cited by the examiner.

DEMONSTRATIONS OF SOFTWARE EMBODIMENTS WHICH ARE REDUCTIONS TO PRACTICE OF THE SUBJECT MATTER RECITED IN CLAIM 6

4. The earliest demonstrations of reductions to practice of the invention described in claims 1 and 6 of the 5,838,906 patent (referred to hereafter as 906) to individuals outside of UCSF Center for Knowledge Management staff were given during November of 1993. These

ATTACHMENT 2

Doyle et al.

Control No. 90/007,858

Page 2

demonstrations included private presentations to influential experts in the field, as well as public demonstrations to large technical and scientific audiences. The software used for these demonstrations incorporated all of the elements of claims 1 and 6 of the 906 patent. A detailed chronology these software demonstrations follows.

Dr. Donald Lindberg

5. One of the earliest demonstrations we made was to Dr. Donald Lindberg, the Director of the National Library of Medicine (NLM) and the Director of the National Coordination Office for High Performance Computing and Communications (HPCC). The HPCC program was an important source of funding for many of the early Web innovators, including the original developers of the Mosaic browser, who worked in the software development group at the National Center for Supercomputing Applications, in Illinois. Therefore, Dr. Lindberg was keenly interested in the state of the art in Web technologies at the time, and in innovations that could push that state of the art forward.

6. Dr. Lindberg was visiting UCSF on November 17, 1993, for a conference regarding the Red Sage Project, which I was directing at the time. We pulled him aside during the conference to come down to my Center in order to view and interact with a demonstration of the 3-dimensional visualization of Visible Embryo Project data via our 906-enhanced web browser. This demonstration involved using our 906-enhanced Web browser, from a workstation in one of the Library's conference rooms, to access the Center for Knowledge Management's (CKM) Web server in order to demonstrate the features of the 906 invention. This demonstration involved showing him the Vis embedded 3-D visualization application, as well as an embedded video player and an embedded molecular modeling application, all embedded in Web pages served up by the Center's Web server over a TCP/IP Internet connection.

NIH/AFIP

7. During the November demonstration, Dr. Lindberg invited me to go to the NLM Lister Hill Center three weeks later in order to do a live demonstration of the system during a platform presentation at an NLM technical conference, in early December, 1993. At that time, I showed the use of our 906-enhanced Web browser, from the stage at Lister Hill, to access the UCSF Web server in order to perform a demonstration similar to the one shown to Dr. Lindberg three weeks earlier at UCSF.

8. During that same trip, I also visited with Dr. Adrienne Noe, currently Director of the National Museum of Health and Medicine, at the Armed Forces Institute of Technology (AFIP), to show her a similar demonstration of the 906 technology. Using a Silicon Graphics workstation in her Human Developmental Anatomy Center at AFIP, I installed and demonstrated our 906-enhanced Web browser, accessing the UCSF CKM Web server to show her the embedded visualization examples, as discussed above.

SIGWEB

9. Less than one week after the first Lindberg demo, also in mid-November, 1993, my UCSF group presented an on-stage live demonstration of our 906-enhanced browser system to a large group of early Web developers at the second monthly meeting of SIGWEB, held at Xerox PARC on November 19, 1993. (Attachment G) SIGWEB was a special interest group for the World Wide Web that was founded, at my direction, by several members of my UCSF staff. Prominent members included Xerox PARC, Sun Microsystems, O'Reilly Publishers, Pacific Bell, NASA Ames, SCO, Lockheed, Amdahl, Netcom, Silicon Graphics, as well as researchers from eight different University of California institutions. (Later SIGWEB conferences featured presentations by Web luminaries such as Marc Andreessen, the creator of the Mosaic browser, and Tony Johnson, the creator of the Midas browser.)

10. Among those in the audience at that SIGWEB demonstration was Sun's Jim Gosling, who later went on to develop the Java applet system at Sun, releasing it in May of 1995. Several individuals from Silicon Graphics Corp (SGI) were at that November '93 SIGWEB meeting, as well, and saw the 906 demonstration. As a result, John Flynn, a key individual at SGI responsible for health care technology applications, invited us to do a demonstration of the 906 system for Silicon Graphics technical management, in mid-January, at the SGI corporate headquarters.

NCSA and the UCSF Digital Libraries grant proposal

11. On October 6th, 1993, I had sent an email to Dr. Larry Smarr, Director of the National Center for Supercomputer Applications (NCSA), the home of Mosaic, asking him if NCSA would be interested in collaborating with my group on a response to a recently-announced request for proposals for an NSF grant program called the Digital Libraries Program. (Attachment D, page 4) My email mentioned that we were "about 3 weeks away from demonstrating dynamic control of 3-D medical volume visualization through Mosaic," referring to our work on the 906 invention. Dr. Smarr passed my email on to Joseph Hardin, who Dr. Smarr stated was the "NCSA Associate Director for Software Development, whose group invented Mosaic." Marc Andreessen and Eric Bina, the co-creators of Mosaic, worked for Mr. Hardin in that group. On October 30, 1993, Mr. Hardin replied in an email stating that he was interested in discussing such a collaboration. I then replied to him in an email on November 18, 1993, sending him general details of the project concept, and mentioning that Dr. Lindberg had "just asked me to use our Mosaic/visualization technology to create demonstration applications using the National Library of Medicine's Visible Human Project data." (Attachment D, page 1)

12. We then had several discussions via telephone, involving myself, Mr. Hardin, and on at least one call, members of each of our teams. During December of 1993 and January of 1994, I sent Mr. Hardin's group various materials illustrating our accomplishments on the 906 invention and my plans for using that technology as the basis for the grant proposal. These materials included a project outline with a diagram of how the 906 technology would fit into a larger digital library system (a copy of this outline is in the 906 patent prosecution file history and is attached hereto: **Attachment E, pages 2-7**), a draft of a grant proposal providing a description of implementation details of an embodiment of the 906 invention that had been completed in preparation for the project (Attachment A), and a videotape demonstration of the 906 invention software in operation. The videotape demonstration was created in December of 1993, and was also the basis for most of the video footage used in the MMVR demo described below. (Attachment B, pages 8-16)

Silicon Graphics

13. As mentioned above, SGI's John Flynn had invited us to do a demonstration of the 906 system. In preparation for that demonstration, SGI loaned my development team an SGI Reality Engine workstation to see if we could adapt it for use as a computational resource for integrating with our VIS volume visualization system, so that the Reality Engine hardware graphics acceleration capabilities could be used to provide a speed-enhanced capability to display and interact with our 3-dimensional embryo data via a Web page being viewed through our 906-enhanced browser. We successfully completed that adaptation and proceeded to give an on-site demonstration of the 906 system at the SGI Corporate Briefing Center in mid-January, 1994. That demonstration was videotaped by my staff. A portion of that demonstration videotape is included in the MMVR video transcript included in the 906 patent file history and is attached hereto: Attachment B at pages 13-14).

MMVR II

14. On November 30, 1993, I submitted an abstract for a presentation to be given at a conference called Medicine Meets Virtual Reality II, which was being sponsored by the UCSD Medical School, the IEEE, and the ACM. Earlier on that day, I had asked Cheong Ang to run the 906 software and to do a screen-capture demonstrating the embedded volume visualization application in action. (Attachment F) This screen-grab image was included in the abstract that was submitted on the same day to the conference organizers. On December 13th, I received a mailing from the conference organizers confirming that they had received my abstract submission. A copy of this abstract is in the 906 patent prosecution file history and attached hereto: Attachment B pages 4-7.

15. Video clips of a 906 demonstration had been captured by my UCSF staff in December of '93. These were dubbed together with clips from the January '94 SGI demonstration, and then combined with a voice-over to create a 906-demonstration videotape that was shown during my platform presentation at the MMVR conference, which started on January 27, 1994, in San Diego, CA. (Attachment B, pages 8-16)

16. A major topic of the conference was the possibility of using the Internet to provide "telepresence" for virtual reality style applications for a variety of specialized medial applications. Since the World Wide Web was beginning to become very well known, there was tremendous interest among the attendees concerning the World Wide Web and whether it might have any relevance to the virtual reality community. As the audience was viewing the video demonstration, they also had in their possession, in their conference proceedings books, the published abstract referred to above, which showed the highly-detailed screen-grab images that Mr. Ang had prepared for me during the previous November. (Attachment B, page 6, and Attachment F) After making my platform presentation and showing the 906-demo videotape, I received immediate expressions of interest from a variety of researchers present in the conference hall at the time. A transcript of the videotape shown at this conference is in the 906 patent prosecution file history and is attached hereto: Attachment B pages 8-16.

AAAS

17. One of the conference chairs at the MMVR conference, the NLM's Dr. Michael Ackermann, was also to be chairing a session on scientific visualization the following month at the annual meeting of the AAAS in San Francisco, in February '94. He requested that I show the same videotape at the AAAS conference. I did a presentation at AAAS that was very similar to the one that I had given at the MMVR meeting to the large audience present at that AAAS presentation. (Attachment B, pages 19-21)

UMich

18. Another person who was present at the MMVR conference was Dr. Brian Athey, from the University of Michigan (UMich) Digital Libraries Project. He invited me to travel a few weeks later to Ann Arbor to present a seminar at UMich on our new browser innovations and their applicability to the Visible Embryo project and other large-scale digital-library-like informatics projects. I gave that presentation to an audience of 10-20 digital library researchers. Shortly afterward, Dr. Athey attempted to recruit me to join the faculty at UMich.

IEEE Visualization '94

19. In April of 1994, Cheong Ang, David Martin and I submitted a paper that we had co-authored to the conference organizers of the IEEE Visualization '94 conference. This paper described an embodiment of the 906 invention. The paper was accepted by the conference committee, and a platform presentation was given at the conference in October 1994. A copy of this paper is included in the 906 patent prosecution file history and is attached hereto: Attachment C.

UPenn

20. Also as a result of the MMVR conference, I was invited in the Spring of 1994 to give a seminar at the University of Pennsylvania (UPenn). That invitation was extended by Dr. Jayram Udupa, the Director of the UPenn Medical Image Processing Group. I showed the MMVR 906-demo videotape to a group of approximately 20 medical visualization and informatics researchers. One of the attendees of that demonstration, Dr. Scott Baldwin, then proceeded to spend several months trying to recruit me to join the UPenn's faculty, as a part of their prestigious Wistar Institute. Also during this time period, I did a live 906 software demonstration at George Mason University to a group of students of Dr. George Michaels, who is currently Associate Director of the Pacific Northwest National Laboratory.

21. Shortly after my group gave two very well-received private 906 demonstrations at UCSF, in August of 1994, to very technically sophisticated and well-funded commercial entities, Continuum Productions, which was personally owned by Bill Gates, and Booklink, a company which was later acquired by AOL, the 906 co-inventors decided to co-found Eolas Technologies Inc., and so I declined the various academic recruiting efforts I was receiving from institutions such as UPenn, and my team went on to launch Eolas.

CONCLUSION

22. With regard to claim 6, a computer program product including a computer usable medium having computer readable program code physically embodied therein to cause a client workstation to perform all the elements and limitations recited in claim 6 existed and was operable in November of 1993. The screen shot at page 6 of Attachment B shows a portion of a hypermedia document, having a URL beginning with <http://www.library.ucsf.edu> which is displayed in a browser controlled window. An object is displayed 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. Interactive processing of the displayed object is enabled by an executable application, external to the distributed hypermedia document displayed, when an embed text format is parsed by the browser. The browser uses the object's type information to identify and locate the executable application as referenced in Attachment A, pages 13 and 14. Different types of objects are thereby handled, as shown in Attachment A, page 28, and as referred to in Attachment A, pages 10, 11, 13, 14, 23, Attachment B, Page 7, Attachment C, page 8, Attachment D, page 2, and Attachment E, pages 2, 3, and 4.

23. With regard to claim 1, all the method steps of claim 1 were performed when the computer program product of claim 6 was executed by a workstation in November of 1993.

Doyle et al.
Control No. 90/007,858
Page 7

24. I declare that all statements made herein of my knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both under 18 U.S.C. Section 1001, and that such willful false statements may jeopardize the validity of the patent.

DATED: September 22, 2007

A handwritten signature in black ink, appearing to read "Michael D. Doyle", written over a horizontal line.

Michael D. Doyle

NSF/ARPA/NASA Digital Libraries RFP Response (draft)

Title: Embedded Visualization Objects for Knowledge Access, Creation and Management through the World Wide Web

Principal Investigator: Michael D. Doyle, Ph.D., UCSF Library and Center for Knowledge Management

Table of Contents:

Project Summary:

Executive Summary:

**Plaintiffs' Trial
Exhibit 690**

ATTACHMENT A

E 006776

Project Description:

Objectives of Proposed Work:

1) To develop a prototype knowledge management environment for the biomedical sciences which integrates access to online representations of the scientific literature, bibliographic databases, high-performance visualization technologies, large-scale scientific databases, and tools for authoring new-generation scientific publications;

1.a) To explore and evaluate the applicability of these tools in the areas of radiology and developmental & molecular biology;

1.b) To provide a means for relating digital forms of spatial, functional, and conceptual information as a basis for linking the biomedical scientific literature, through the Red Sage electronic journals project, to data resources provided through the Visible Human Project, The Human Brain Project, The Visible Embryo Project, The Human Genome Project, The Protein Database, and other large-scale biomolecular and biostructural databases;

1.b.1) To exploit these linking strategies in the creation of a set of integrated semi-automatic front ends to varied scientific databases accessible through the Internet;

1.b.2) To incorporate these linking methodologies into interactive authoring and editorial tools, allowing the creation of online publications that can embed visualizations and simulations which draw data from these Internet-accessible scientific databases;

1.c) To develop tools which provide access to interactive visualization and analysis of massive biomedical datasets through the Internet's World Wide Web distributed hypermedia network;

1.c.1) To refine and extend our existing algorithms enabling distributed visualization and analysis software "engines" which can be efficiently accessed by remote users through the Internet;

1.c.2) To refine and extend our existing algorithms to allow the display and real-time interactive control of three- and four-dimensional data visualization and analysis tools within hypermedia documents viewed using NCSA's MOSAIC graphical browser to the World Wide Web;

1.c.3) To develop algorithms which use novel compression technologies for the optimized interactive remote control of computationally-intensive graphical applications through the Internet;

1.c.4) To integrate a, b & c into a system which allows real-time remote access to distributed parallel computational applications for visualization and analysis resources within a distributed hypermedia environment;

1.d) To explore extensions of the paradigm of scientific publishing which are made possible through use of current multimedia technologies in a networked environment, including:

1.d.1) publishing multidimensional datasets integrated with articles, eg: MRI and molecular data, preferred views, animations, interactive visualizations, interactive mathematical models, and

1.d.2) development of scientific authoring tools for publications which exist only in the networked environment.

1.d.2.a) This will include integration of HTML+ WYSIWYG authorial and editorial tools, multidimensional data visualization applications, molecular modelling and database management tools into an interactive scientific publishing environment.

Background and Significance

Knowledge Management

Biomedical professionals today face an imposing problem. The amount of information which they must assimilate just to keep current in their fields has created a situation for them which has often been described as being like "trying to drink from a firehose." This results in an environment where improvements in an individual professional's information management tools can result in drastic improvements in their effectiveness in such areas as quality of health care, significance of research discoveries, or commercial potential for new inventions. This problem is especially significant for industrial researchers, who often have problems even in accessing the information they need, which is often found in scientific literature collections at major academic institutions. Computer-based information technology shows promise for solving these problems eventually, but limitations in computer capacity, cost and networking speed pose formidable barriers to reaching solutions to these problems. The presently-proposed project will attempt to exploit state-of-the-art computer, software, and networking technology to begin to tackle these problems.

In 1990, Richard Lucier, University Librarian at the University of California, San Francisco, charted out his view of the future of academic libraries in an information technology environment. The following is adapted from his paper published in EDUCOM Review (Nov.-Dec., 1992, vol 27, (6):24-31).

Since the 1980s, network access to knowledge through online catalogs and indices, along with a new emphasis on service and education, has assumed major importance at academic institutions across the nation. However, during that time, the respective roles and functions of scientists, publishers, and librarians have remained fundamentally the same.

Scientists routinely discover new knowledge and then communicate it through both writing and teaching. Publishers then disseminate that information through a variety of primary and secondary information products. In their traditional storage-and-retrieval role, research libraries use these resources to build large collections and thereby make available to users the world's published literature.

The limitations of this prevailing model for scientific communication were becoming apparent by the late 1980s. The length of the hard-copy publishing process makes it increasingly difficult for scholars and scientists to communicate their findings in a timely fashion. Faced with a long-term financial crisis evidenced in libraries by the rising cost of information, universities could no longer afford to maintain comprehensive collections. As a result, the presentation of knowledge in static form, whether in print or as part of the emerging electronic library, has begun to appear grossly inadequate.

Due in part to these factors, scientists, often on their own and with inadequate support, are beginning to augment the passive presentation of knowledge by adding interactive, discipline-based knowledge bases. These are routinely developed, maintained, and shared across computer networks. Problems resulting from the heterogeneity of data formats and user interfaces can often be handled only in an ad hoc manner, however, because there is usually a lack of widespread and organized support for information exchange activities. Knowledge management has emerged from this environment as a creative response to managing the world's knowledge base.

What is remarkably different about the knowledge management model is that it insinuates the library at the beginning of the information transfer cycle, rather than at the end, and focuses on information capture rather than access and use.

Encompassing the entire information life cycle, from creation of new knowledge to its dissemination and use, knowledge management represents a collaborative enterprise in which scientists and the library work together to develop and maintain knowledge bases and derivative information products. Knowledge management constitutes a new model for scientific communication in which faculty and a multidisciplinary team of computer scientists, software engineers, and research librarians share responsibility for the collection, structuring, representation, dissemination, and use of knowledge through the use of electronic information technologies. Knowledge bases are developed and maintained through knowledge management processes, which ensure content integrity and usefulness.

The collaborative nature of knowledge management, embodied in techniques of shared development of functional specifications, rapid prototyping, and user acceptance testing, fosters an interdependency among the collaborators. The ongoing management of the knowledge base also requires funding and administrative strategies that cut across traditional departmental, disciplinary, and institutional boundaries.

The long-term implications for building and maintaining a portion of a library's collection in this manner are enormous. Knowledge management transforms the various roles in the scientific communications process and potentially places ownership and control back into the hands of the scientific community. It also has tremendous potential for closing the gap between research faculty and students and for integrating the library into research and education programs in a significant way.

The knowledge management view that Mr. Lucier has proposed outlines many of the fundamental principals behind UCSF's Center for Knowledge Management.

...more on CKM...

The Red Sage Project

The Red Sage project was created as a first step towards achieving a functional knowledge management environment at UCSF. This project arose from the belief that university libraries must thoroughly investigate the applications and consequences of state-of-the-art information technology to their mission of serving the information needs of their campuses. The university library must become less of a storage facility and more of a center for information management, for learning, and for consultation.

It's important to note that this kind of work cannot be successful in a vacuum. Many factors in the areas of both publishing and information technologies are of critical importance. So about one and one half years ago The UCSF University Librarian, Richard Lucier, met with a major scientific publisher, Springer-Verlag, and a major player in the information technology industry, AT&T Bell Labs, to decide that it would be feasible to begin a project looking at the problem of designing, implementing and evaluating a fully digital academic library. The system would use AT&T's innovative RightPages™ electronic journal software as the archiving and delivery platform. This agreement was struck over dinner in the Red Sage restaurant in Washington D.C., and so the Red Sage project was born. Over the last year, many additional publishers have joined the team. These include John Wiley & Sons, the American Medical Association, the Massachusetts Medical Society, publishers of the New England Journal of Medicine, and the Nature Publishing Company, publishers of Nature, as well as many others.

The primary goal of the Red Sage Project is to explore the technical, legal, economic, business, and social issues surrounding the delivery of scientific, technical, and medical information in a

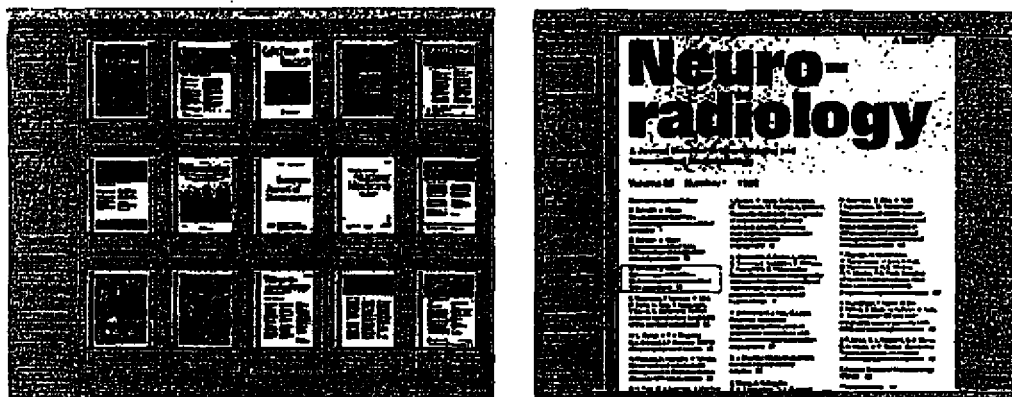


Figure 1: AT&T's Right Pages client-server system presents users with a traditional library metaphor.

networked environment. An additional objective is to establish a critical mass of content that will make this system a "must-use" resource for the faculty, staff and students on the UCSF campus. Therefore it was decided to limit the initial database to a few key content areas, so that we could achieve good coverage of content critical to a large number of research and clinical faculty on campus. These areas included both radiology and molecular biology, as well as an area called "general high impact", which included a few publications which appeal to a general audience, but which have a very high readership on campus, such as the New England Journal of Medicine. The project will have over 40 titles available online when the system is first opened to the public in the first week of January, 1994.

There are some main features of this system which are of note: First of all the project team decided that, in order to get the system up as soon as possible, they would try to use an existing software package that was stable enough to be used as a "production system" in the Library. It needed to allow the user to follow a traditional library metaphor (Fig. 1), organizing the information in the form of stacks, journals, years, issues, and articles, just as these things are organized in the Library. It required a graphical user interface that allowed onscreen display of journal pages, providing the same "look and feel" as their paper equivalents. There had to be a useable search engine, and it was desirable for that engine to be able to record an individual's preferences and to alert the user to pertinent articles as soon as they were loaded onto the system. To provide this functionality, and to insure reliability and system integrity across a wide variety of client platforms, a client-server architecture was preferable. To allow facile analysis of human factors issues, the system needed to allow retroactive analysis of the users' practices and patterns of use. And, finally, the team was aware that, at least initially, most users will use the system for finding and screening the articles they need, but will want to print out the papers before studying them. Therefore, high quality printer output to a local machine was required. The X Window-based RightPages electronic journal system from AT&T met all of these needs well.

Image quality was a primary consideration in this project. The Red Sage team realized that, in order for the users to accept the system, they must not be asked to sacrifice legibility of either text or images in the online versions of the articles. The onscreen display first shows the article so that the entire page fits on the screen. Depending upon the size of the screen, this may be adequate for easy viewing or, alternatively, the image can be zoomed to allow easier reading, as well as detailed examination of tables and figures. Photographs are shown on the pages in a binary, or black and white, mode, but if the user wants to see a higher quality view, a button on the control panel will bring up a grayscale, or eventually color, display of that photo (Fig. 2).

In addition to the bitmapped images of the pages, the system provides an underlying full text ASCII database that can be searched with conventional approaches. This searching capability can be automated so that the system stores user profiles of search terms, and new content is

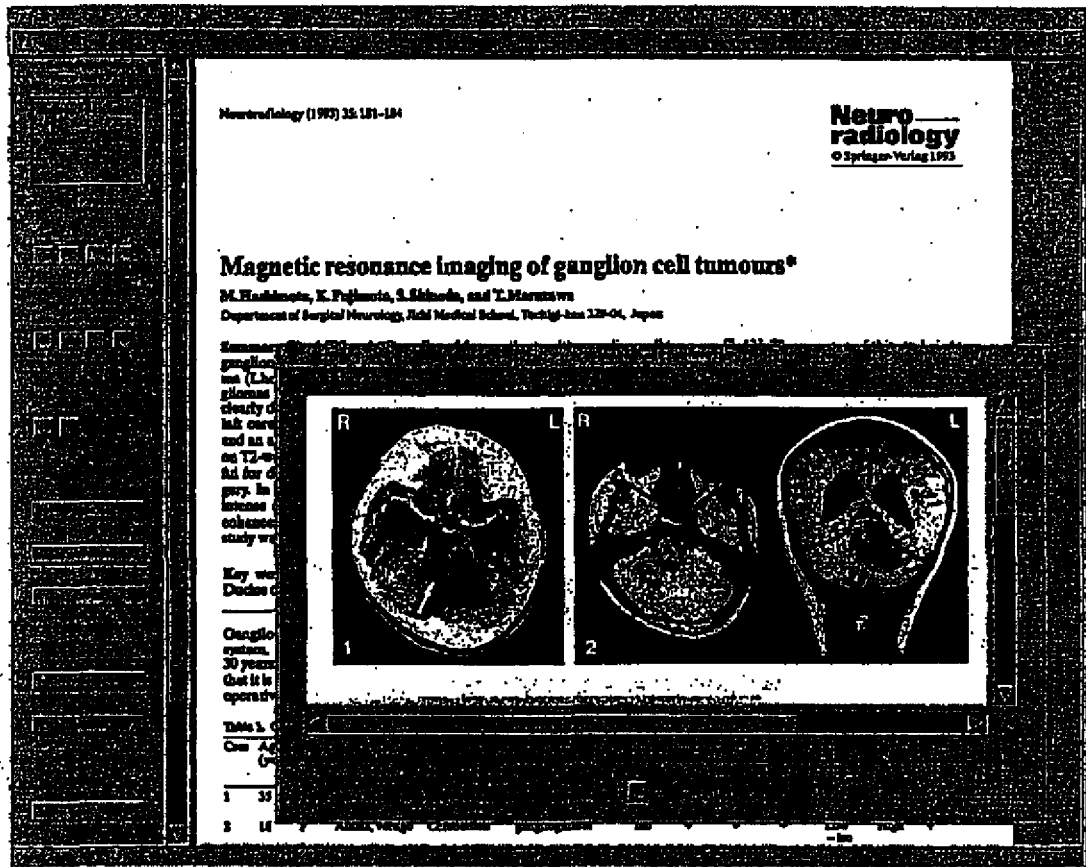


Figure 2: The Red Sage system provides high-quality grayscale images for viewing at the click of a button.

searched automatically as it is being loaded onto the system. If newly-loaded content matches those parameters, the next time the user logs on he or she will see the relevant issues highlighted on the stacks display.

This alerting feature represents perhaps the most important aspect of online literature systems in comparison to conventional printed journals. With the deluge of scientific and technical information that is printed every day, it is both physically and economically impossible for researchers to keep up to date without the aid of computer-based indexing and retrieval systems. MEDLINE, from the National Library of Medicine, provides this capability for much of the biomedical research community. The articles covered by MEDLINE, however, routinely appear 2-3 months after the date of publication. The articles included in the Red Sage database routinely appear from 1 to 8 weeks before their printed equivalents are available. This is very important to both academic and industrial researchers, where a piece of key information in a published article can drastically influence the course of an important experiment or clinical care protocol.

The World Wide Web

The delivery of information via electronic means has grown over the past several years to include text, images, audio and video [REF]. A significant obstacle to overcome in delivering this information in a ubiquitous manner has been a combination of format and accessibility. Although significant steps have been taken to alleviate these problems, it was not until the acceptance of the World-Wide-Web (W³) and the availability of a portable and redistributable client, in the form of Mosaic from the National Center for Supercomputer Applications (NCSA) at the University of Illinois at Urbana-Champaign (UIUC), that a method has become available to distribute a wide variety of multi-media information in an inter-networked environment.

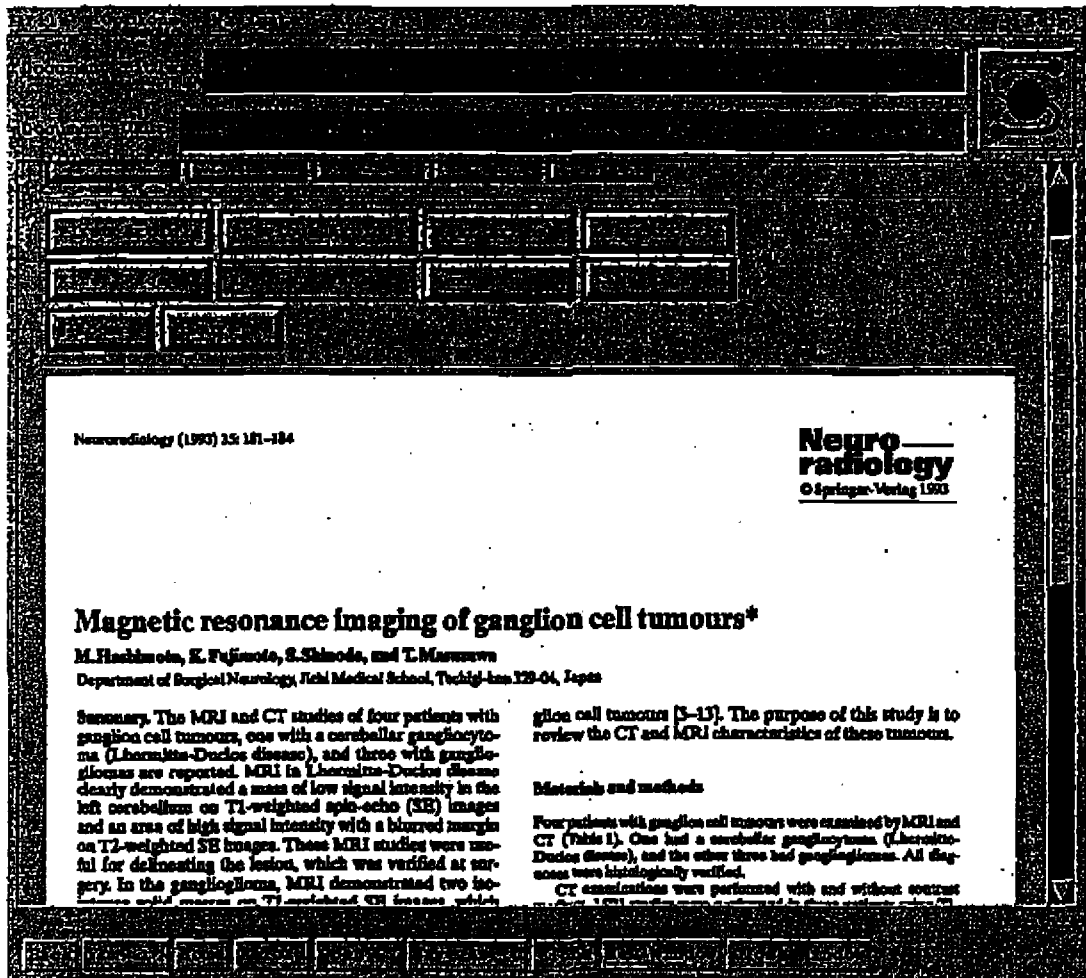
1.1 The World-Wide-Web

The World-Wide-Web (W³) provides a means for accomplishing information access, retrieval and structure support over the Internet through the definition of a network communications protocol for transmitting hyper-text documents (Hyper-Text Transfer Protocol - HTTP) and a SGML document type definition (Hyper-Text Mark-up Language - HTML). These two pieces of technology, combined with a client capable of presenting multi-media data, and a server maintaining a repository of information, provide the basic mechanisms for delivering multi-media information in a client/server fashion over the Internet. In addition to the protocol and DTD, the W³ also provides connectivity to other inter-network services, such as Telnet, FTP, Archie, WAIS, electronic mail (SMTP/MIME) and network news (USENET). This additional connectivity allows a WWW client to be used as an inter-network information integration agent (INIIA).

MOSAIC-based Red Sage Browser

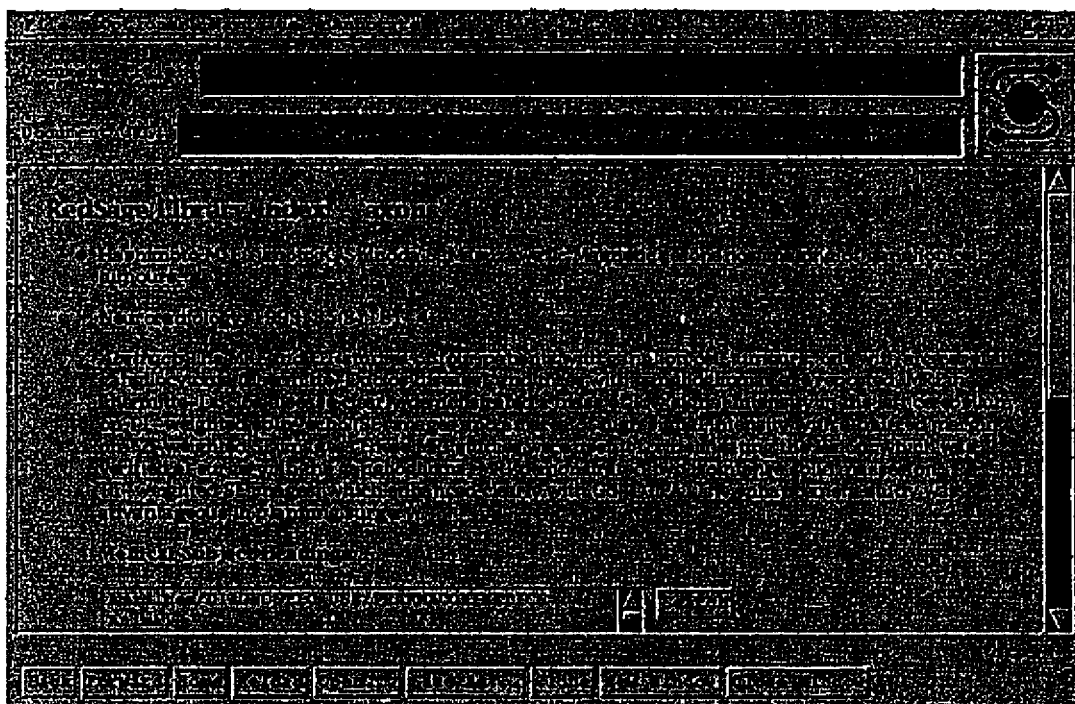
We have utilized content from the RedSage Project [REF], technologies from NCSA/UIUC, the Center for Network Information Discovery and Retrieval (CNIDR), the European Center for Particle Physics (CERN), the University of California at Berkeley (UCB), and other individuals and institutions to provide an on-line digital library of medical journals in the areas of radiology and molecular biology. This repository is not only accessible via the Internet, but may be utilized as both a retrospective and prospective information source, assisting users in the identification and dissemination of relevant articles. Our project is an experimental evolution of the RightPages™ software, utilizing the same content base of journal titles, but extending the scope of information discovery and retrieval to include not only the journal literature, but also the bibliographic information available from on-line indices and catalogs as well as the breadth of services offered by other Internet providers (e.g. GDB, GenBank, OMIM).

...DIAGRAM...



Medline

...DIAGRAM...



Integrated Vis

Advance scanning devices such as Magnetic Resonance Imaging (MRI), Computer Tomography (CT), and Satellite Imaging have been widely used in areas like medicine, defect detection on the production lines, and meteorology. The need to visualize the sampled data quickly spurred the development of various volume visualization techniques, which in turn, provided the 3D imaging related research groups a revolutionary tool that allowed noninvasive examination of the object of interest. As a result, visualization systems such as AVS, ApE, Sunvision Voxel, and 3D Viewnix have been subsequently introduced to the user communities. These high-end commercial software systems, however, are not cost-effective solutions for the majority of the users who are interested in visualization of the published data and results. Accessibility of the tremendous datasets, and availability of a suitable graphics workstation further hinder most people's possibility of experiencing the authors' findings.

Our project attempted to integrate volume visualization with W^3 , and thus take advantage of the client-server architecture of W^3 to allow centralized data storage and on-demand dissemination. We enhanced the capability of the W^3 server, PLEXUS, from Knowledge Systems and the Computer Science Department of Carnegie Mellon University, and the W^3 client, Mosaic, from the National Center for Supercomputer Applications (NCSA) to support the 3D data object. In addition, we united our distributed volume visualization system, VIS, with Mosaic for 3D volume manipulation and image computation. The distributed nature of VIS allows the users to utilize remote computation servers, and therefore, enables visualization on low-cost workstations.

The data storage sites run W^3 servers which communicate with W^3 clients such as Mosaic via Hyper-Text Transfer Protocol (HTTP), the same way the banal FTP servers serve a file transfer session. Upon receiving a service request, the W^3 server will establish a connection with the client, and transmit the requested digital objects (documents, sound clips, images etc.) with their respective tags in the headers. The client will read the headers, and based on the types of data, decide the appropriate actions. For example, a W^3 server will reply to an image transfer request from a client by sending a header

containing /img? as the data type followed by the image data. The client will peek into the header and invoke its image handling function(s).

1.2 About the W³ client Mosaic

Mosaic is one of the major W³ clients. It has versions for different platforms, which include X-Window Systems, Microsoft Windows 3.1, and Macintosh. Because of its fast development and wide availability, it is one of the most prevalent web browsers. The current version (2.2) supports documents written in Hyper-Text Mark-up Language (HTML). It enables not only page formatting, type-face formatting, and image displaying, but also specification of hypertext links and creation of a variety of Graphical User Interface (GUI) widgets.

1.3 About the volume visualization tool VIS

VIS is a simple but complete volume visualizer that reads the Hierarchical Data Format (HDF) volume data files, and performs fast texture mapping of the volume data onto the geometry formed by six axial clipping planes (cube), and one arbitrary clipping plane. It also has a distributed volume rendering routine that supports run-time GUI based selection and deselection of computation servers, and an isosurfacing function with software Gouraud shading (thus special graphics hardware is not required) for surface extraction and rendering.

1.4 Goal

Through the integration, the W³ client (Mosaic) supports 3D image data visualization by spawning VIS when the data type description passed from the W³ server is a 3D volume image.

The rest of the paper is divided into five parts. Section 2 will describe VIS and how it fulfills the role as a W³ visualization tool. Section 3 is devoted to Mosaic, and will cover the implementation of its interface with VIS. The last section is about our conclusion and possible future development.

E 006786

2. VIS as a W³ visualization tool

VIS was first developed to be a portable visualization tool among X-Window system workstations. It is designed to carry out most of its functions as fast as possible without using any graphics hardware. We traded the possibility of real time rendering with the ability to run on most workstations: SGI's, DEC Alpha's, IBM RS 6000's, Sun SPARCStations etc. In order to provide the best interactive performance, we modified VIS in two different directions: (1) utilization of graphics hardware, and (2) distributed rendering on powerful workstations. The results of rendering are sent into Mosaic with a set of predefined network protocols.

2.1 Real time VIS

With the help of an SGI Onyx with RealityEngine² (RE²) graphics, and the OpenGL implementation for the RE², we are able to implement volumetric rendering in real time. The rendering routine defines the input volume dataset as a 3D texture map. This setup allows not only interactive transformations (rotations, translations, and scaling), but also provides the GUI for interactively defining the axial and arbitrary clipping planes, and specifying the intensity-to-density and density-to-opacity functions.

2.2 Distributed VIS

The VIS implementation has great potentials to run in parallel. We distributed the loads of volume rendering among workstations with a greedy algorithm that chooses to give bigger portions of the picture to faster machines. The client that VIS is running on will break the load into a relatively small portions, which are still big enough such that the overhead for network transmission of the data is substantially smaller than the time to transmit the actual data. Each of the computation servers will then fetch a portion for rendering via Remote Procedure Calls (RPC). A fast server will return the results earlier, and fetch another portion. The servers will compete with each other for data, and this ensures that most servers are busy most of the time. A report of analysis of this algorithm from Giertsen and Petersen (Giertsen, 93) shows that the performance improvement is a function of the number of load portions, and the number of computation servers. The test results suggest that the performance improvement max out when the number of sections of 512x512 and 1024x1024 pictures are around 10-20 on several volume datasets of various sizes on up to four servers. They also show that a larger number of workstations servers will subsequently enlarge the optimal number of sections. This approach may not be the best distributing algorithm, but it achieves the close-to-optimal results most of the time.

2.3 Interface with Mosaic

The VIS/Mosaic software system consists of 3 elements: VIS, Mosaic, and Panel (which provides GUI for data manipulation). Currently, both of the above VIS implementation communicate with Mosaic through the standard X interclient communication protocol (ICP), but the same algorithm will theoretically work in various windowing systems that support ICP. Upon receiving the notification to load a 3D image dataset, Mosaic creates a DrawingArea widget in its document page, and a pixmap as a storage of rendering results. It then starts a child process, which in turn, launches Panel. The fact that both Mosaic and Panel are running on the client machine, and thus sharing the same windowing system server, provides a convenient communication bridge between the two. The pixmap created by Mosaic will serve as a chunk of shared memory (for images) accessible both by Mosaic and Panel. In addition, Panel and Mosaic may send messages through the windowing server to coordinate the image displaying.

Beside coordinating the displaying of images, Panel functions as a load distributor. It delivers the rendering parameters to the selected VIS server(s), and fetches the resulting image portion(s). Thus the scenario following a user's action on the Panel will be (1) Panel requests VIS server(s) to render the image with the parameters from its GUI, (2) Panel fetches the returned image data, then writes it to the pixmap, (3) Panel sends a message to notify Mosaic upon completion, and (4) Mosaic bit-blots the image

in the pixmap into its DrawingArea widget. This will be addressed in more details under section 3.2. The configuration of this software system is depicted in figure 1 below.

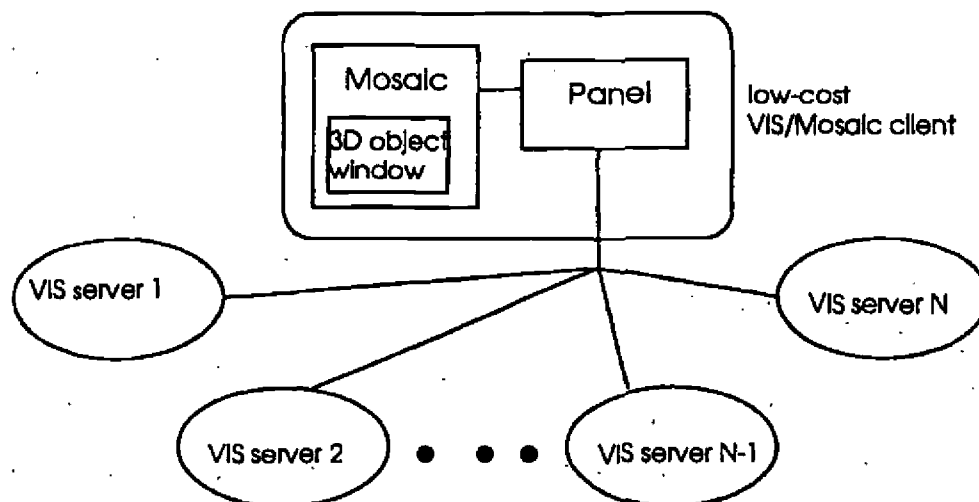


Figure 1: VIS/Mosaic software system.

3. Mosaic and the W³ server

Supporting of the 3D volume data object requires both Mosaic and the W³ server (PLEXUS, in our case) to recognize this data type. We also defined the roles of each of VIS, Mosaic, and PLEXUS.

3.1 Startup coordination

Since the administrator of a W³ site will declare to its W³ server the type of each W³ file, we added the 3D volume data object to the PLEXUS server, and declared the HDF volume datasets to be such type. The W³ client (Mosaic) is modified to handle this type of data when it sees the tag in the HTTP header. As mentioned in the last section, Mosaic creates a DrawingArea widget and a pixmap, and forks a process to start Panel, which in turn, starts the user specified VIS server(s). Once started, VIS server(s) download(s) the HDF data from the same PLEXUS server with the Universal Resource Locator (URL) from Mosaic. The modifications involve addition of a new tag into the PLEXUS code, enhancement of the HTML library on the Mosaic side, and implementation of URL loading function for VIS.

3.2 Interclient communication

We recognized the minimum set of communication protocols between Mosaic and Panel:

- (a) Mosaic notifies Panel in events when the window id is no longer valid. This happens when the user navigates from the current page to the previous (with BACK button), or to another document through a link.
- (b) Panel notifies Mosaic when the scene needs to be updated. This occurs when the user manipulates the data through the Panel's GUI, which may be any actions (rotations, scaling, changes in clipping planes, mode switching, for instance, from texture mapping to volume rendering) that result in re-rendering of the picture.

The first case of (a) does not impose any problem since Mosaic can simply terminate VIS. The second case is harder to handle because Mosaic caches the page, and in principle, we expect VIS to remain running because it should not be loading the HDF data, which may be quite a few megabytes, again. In the previous case, we simply have Mosaic kill the child process that launched Panel, which may subsequently send a terminate message to the VIS servers. In the latter case, Mosaic will send a message to Panel requesting it to unrealize/unmap its window. Then when the user comes back to the 3D object page, Mosaic may send another message to Panel ordering it to realize the GUI window, and update the window id of the DrawingArea that Panel has been sending message to (ironically Mosaic does not cache the widgets/windows on the cached page).

In (b), Panel will send an update message directly to the DrawingArea window id provided by Mosaic. Upon receiving the redrawing message, the DrawingArea widget calls its registered refreshing function, which simply bit-blots the pixmap Panel updated into the window.

4. Conclusion and future work

Our accomplishment takes the technology of networked multimedia system (especially the W³) a step further by proving the possibility of adding new data types to the W³ servers and clients, and coordinating the execution of the applications that handle them with the W³ clients. The addition of the 3D volume data object in the form of HDF to the W³ is welcomed by many medical researchers, for it is now possible for them to view volume datasets without a high-cost workstation, and accessing datasets in the W³ fashion, through hypertext and hypergraphics links within an HTML page. As for the researchers who would like to share their findings with the world, they merely have to run a W³ server that serves the HDF files, which in turn, can be easily created with the various import facilities available freely from NCSA.

A very useful future development will be to define a clearer set of communication protocols, probably built on top of a popular network communication tool (ToolTalk) licensed by many major workstation/PC operating system vendors, that can be easily utilized with minimum modification in external application to communicate with Mosaic. Another possible solution is to use the Data Transfer Mechanism supported by Mosaic to handle the communication.

MetaMAP

Volumeic data--huge datasets--isn't practical to interact with entire dataset--multi-tiered system needed--Navigation is a problem--point-and-click selection of anatomical regions desirable

To effectively exploit the advantages of new mechanisms of information and retrieval, new approaches must be made towards incorporating existing programs as well as developing entirely new applications. There exists a need to integrate more sophisticated graphics into applications and to take a wider view of how that integration can take form. Nowhere is this more evident than in the design of the human-computer interface. According to James Foley in a 1987 Scientific American article entitled "Interfaces for Advanced Computing", "The interface between the user and the computer may be the last frontier in computer design (Foley, 1987).

Palette segmentation indexing (PSI) provides a new method for the creation of interactive computer applications which allows the user to interact directly with digital video images. The types of images that are most appropriate for this process are those which are made up of a number of discrete image elements, or objects, each of which has its own identity. An image of the anatomy of the arm would be a good example. The MetaMap process would allow the creation of a program where a user could interact with such an image merely by pointing to different areas of the arm picture with a mouse and clicking a button. The system could then identify the anatomical structure being pointed to, and automatically find and

display text or additional images which explain the important concepts concerning that anatomical structure.

MetaMapping deals with images as object-oriented entities. Single images are typically made up of multiple discrete elements, each of which has an identity in its own right. This is accomplished through a sophisticated technique, called palette segmentation indexing, for reprogramming of both the image and the color map so that the discrete image elements comprise unique sets of color indices. This results in a highly simplified mechanism for the identification of any particular image element or "object". It also produces a very efficient form of object-identity data storage in a spatial-resolution-independent system.

A MetaMap-based interface is most appropriate for hypermedia-style applications. "Hypermedia" describes a particular type of computer interface and database structure where any particular data item (or object) can be related, or linked, to any other item. This concept was first proposed by Vannevar Bush (Bush, 1945), and later popularized by Ted Nelson (Nelson, 1988). The object orientation of MetaMapped images, the space efficiency of the indexing data, and the simplicity of the identification process combine to extend the concept of hypertext applications to include hyperimages as well.

The MetaMap process is device independent. This means that programs which use it can be made to operate on many different kinds of computers. This process is protected by a United States patent (Doyle, #4,847,604), however a royalty free license will be granted to this project for implementation of the MetaMAP process in the browsing system.

A MetaMap-based interface falls within the category of "direct manipulation" interfaces. According to Ben Schneiderman (Schneiderman, 1982, 1983), direct manipulation interfaces can be identified by the three following "essential" characteristics: 1) Continuous representation of the objects of interest, 2) The interactive activity comprises physical actions instead of complex syntax (especially those actions which mimic the desired real world counterpart's actions or changes), and 3) Rapid incremental reversible operations whose impact on the object of interest is immediately visible, so that the objects upon which the actions are taken are exactly the same as those upon which the evaluation is made. Hutchins, et al. (1987), state that "a good Direct Manipulation interface is invisible - The user feels as if operations are actually done directly in the task domain. And if the computer novice is already knowledgeable in the task domain, then much of what is needed to use the interface is already known." Hutchins, et al., go on to state that a problem with direct manipulation interfaces is that they have difficulty distinguishing the depiction of an individual element from a representation of a set or class of elements. Such problems do not arise with palette segmentation indexing (MetaMapping). The improvements inherent in the MetaMap process enable it to fully exploit the potential of direct manipulation without encountering the problems of earlier systems.

The Principal Investigator (Doyle) originally developed this process with the aim of improving the interactivity of computer-aided medical instructional programs on very low-end microcomputer platforms. Previous such systems that used digital imagery dealt with the image as a single entity. These systems were predominantly text-based and did not allow the user direct interaction with the various elements which make up an individual image (Thursh, et al., 1985, 1986; Gilckman, 1987). This was partly due to the fact that the computer hardware necessary to display high-quality color digital images was far too expensive to be considered for inclusion in an educational workstation. As a result, many existing computer-aided instructional systems have entirely text-based interfaces or depend upon the use of graphic symbols (icons) to represent real-world objects or concepts (Nelson, 1988). Using the MetaMap process, realistic images are employed to communicate instructions and concepts to the user. This is accomplished by exploiting several capabilities of today's high-quality graphic computer display hardware.

Human color perception outstrips the abilities of even the most powerful supercomputers. Due to this fact, developers of interactive graphic systems are faced with numerous compromises which must be made. If monochromatic (gray-scale) images are used, this problem can be significantly reduced. Whereas humans are capable of distinguishing approximately 7.5 million colors over a wide dynamic

range of intensities and saturations, variations in grayscale are only perceptible to a few hundred gradations (Murch, et al., 1988).

If an image is rendered in too few colors or grayscales, a phenomenon called contouring occurs. This involves the appearance of visible bands across areas that should be smoothly shaded, looking somewhat similar to elevation bands on a topographical map. Continuous tone images rendered in 2 to 8 grayscales show obvious contouring. If 16 to 32 gray levels are used, a drastic improvement is visible (figure 1). This usually produces only subtle contouring for most images. High frequency images, i.e., those that are highly textured or have many abrupt changes from dark to light, can typically be rendered well with fewer gray levels than softer, smooth shaded, low-frequency images.

This requirement can usually be reduced if an image is divided up into its constituent objects. Most objects have a characteristic tonal range which can often be represented with a limited number of gray values. As will be shown below, MetaMapping allows each class of image objects to be treated separately when determining the optimal number of grayscales or colors necessary to faithfully render the image.

Today's microcomputer systems are commonly capable of displaying 256 colors onscreen at a time. Each of these 256 colors can typically be any of 256 thousand or more possible combinations of hue, saturation and intensity. Many of the newer workstation-level computers can display between 4096 and 16 million colors onscreen simultaneously. The majority of images contain no more than 30 thousand distinguishable colors (Campbell, et al., 1986). The problem of contouring is more complicated with color imagery. Whereas with grayscale images, intensity changes alone contributed to perceptible banding, color rendition includes hue and saturation components to further complicate the issue.

One of the primary motivations in limiting the number of onscreen colors to as few as permissible, is that, with more colors displayed simultaneously, the computer must manage and store a significantly larger amount of data. One unit of computer memory (1 byte) can store the information for eight pixels in a 2 grayscale image, or one pixel in a 256 grayscale image. If a 16 million color image is to be represented, then it will require three units of memory (3 bytes) just to store one pixel's color information.

To combat this problem, computer hardware designers introduced the concept of a color look-up table (CLUT) to allow many different color combinations to be displayed while requiring usually only one byte per pixel to store the displayed image file.

The byte for each pixel location stores an index from 0-255 which is then linked to a table that contains 3 numbers, each of which describe the relative proportions of red, green and blue to be displayed onscreen at the pixel location. This table is usually referred to as the "color look-up table" (CLUT), "palette", or "color map." The collection of indices which together describe the entire image is called the "bit map." The palette values can be reset for each image to be displayed and a file containing these values is usually stored on disk together with the image file. When the computer reads in an image for display, it first reads the image file to find the index value for each pixel location on the screen, and then it reads the palette file to find the R, G, B numbers that dictate the color for that index. Some file formats, such as the popular "PCX" type, combine the image pixel data together with the palette data into a single composite file.

If the specific colors to be used to render an image are carefully chosen, this can result in the capability to display and store pleasing, high quality images while keeping the data file sizes manageable. Several methods have been published for the optimal selection and mapping of limited numbers of colors to render realistic looking images. One of the best, and the most commonly used today is the median cut algorithm (Heckbert, 1982; Campbell, 1986). This process uses a method called adaptive color quantization that uses color histogram analysis and specialized sorting techniques to convert an image from 16 million colors to a specified subset of values, i.e., 256 or 4096 colors.

Heckbert's method uses the standard R, G, B model (Murch, et al. 1988) for representing the original image space, however he states that better results could be obtained by using a color model that is more closely based on actual human color perception. Many companies, notably Kodak, are actively doing research to develop such models, but they usually keep most of their results secret so they can implement them in proprietary technology. One of the most promising perceptual-based models is the well-known Munsell color system. This system represents colors in an hierarchical "stem and leaf" organization where a range of similar colors appears on a single leaf. Ongoing research is underway to develop algorithms to allow computer systems to reference color values in such a way (Silverstein, 1990).

Developers of interactive computer educational training systems have just recently begun to exploit the ability to incorporate high quality graphics into their presentations. Often times it is desirable to allow the user to interact with the system by manual selection of a feature of an image on the video display. Developers using earlier computer systems with low resolution screens, and only a small number of color possibilities per pixel, often used the computer's bit map as a basis for referencing object identity on the computer display.

As was stated earlier, the computer represents each image as a set of indices corresponding to each pixel location. These pixels are arranged on a Cartesian coordinate system, called the bit map, where X and Y value relate to a specific location on the screen. To identify objects on the screen using bit-map methods, one must store a set of x,y coordinates in the computer's memory for each object to be identified. One bit-map-based method constructs a secondary bit map in memory that stores for each x, y screen location an index, or pointer, that links that screen location to additional data concerning the identity of the image object at that location. To illustrate this point, reconsider the earlier-described example of arm anatomy (figure 2). This user would move the mouse to point at a muscle that he or she couldn't identify. If the screen location where the mouse clicked was, for example, pixel location x,y = 118, 390, the computer system would then examine the secondary bit map at x, y = 118, 390 and find that the number at that location pointed to data indicating that the object in question was the brachialis muscle. The system would then respond with text, sound or other images to explain important facts about the brachialis muscle. As we will see below, there are a variety of problems associated with this approach.

An alternative approach would be to store just the x, y locations that outlined each image object. Then to identify the object, the system would have to search through all of the outlines relating to that image and determine for each outline if the selected screen coordinate lied inside, outside or on the outline. Once the relevant outline was located, the system would then look for a pointer from that outline data to other data identifying the anatomical structure in question.

...Mosaic's implementation of Interactive hotspots - ISMAP...

The two methods of object identification described above have a variety of advantages and disadvantages associated with them. The first method, that of storing an object-identity pointer for each screen location, can be very fast if the resolution of the image is very low. It also provides good accuracy in that every pixel on the screen has a defined identity. As the resolution of the image increases, however, the speed of object identification decreases geometrically. Also the amount of extra data that the computer is required to manage for each image, the data overhead, is equivalent to the size of the original image file (100% overhead). To combat these problems, many developers instead use an invisible grid which is overlaid on the image. The system would then look for the closest grid intersection to the selected screen location. Each grid intersection would then contain a pointer to object identity information. This has resulted in drastic improvements in speed and reduced data overhead, but the accuracy of the earlier method was lost. Irregularly-shaped objects would often be erroneously associated by the identification algorithm with the wrong grid intersection and consequently with the wrong object identity information.

Because of this fact, the outline-search method is the most commonly found method today (Supercard, Opus, Toolkit). This technique had the advantage of good accuracy even for irregular objects, and both

fast identification times and low data overhead if the number of objects and complexity of object contours is low. If the number of objects is large, however, and/or the average complexity of object contours is high, then the advantages of this technique are lost. Also, oftentimes image objects can be grouped into classes on objects which share common identity information. A good example would be a microscopic image of a section through a lymph node. There may be several hundred cells visible on the screen, but they may be groupable into, for example, lymphocytes, macrophages, plasma cells and reticular cells. If one were to use outline methods and wanted to identify every cell in the image, then several hundred outlines would have to be stored, and both identification speed and data overhead would suffer considerably.

The current trend in display technology is towards megapixel (1000 x 1000) and higher display resolutions. Any interactive graphics system that is to have a long-term value must be easily converted to operate on new generations of hardware as it becomes available. Systems based on bit-map methods for object identification will have to be entirely rewritten to adapt to the required higher-density indexing schemes. Developers will also have to compensate for the geometric decreases in identification speed and increases in data overhead that will necessarily follow. This will, to some extent, be aided by normal increases in processor speed and memory capacity, but the performance of such systems will always suffer in comparison to color-map-based techniques.

The MetaMap process of palette segmentation indexing (PSI) was developed in response to the disadvantages of bit-map methods. Rather than establishing an object-identity pointer for specific screen locations based on the bit-map, MetaMapping involves using the value of the color index as the basis of object identification. In a 256 color system, for example, one class of objects could be rendered using color index values from 0-31, another class of objects using values 32-63 and so on. Suitable selection of R, G, B values for the CLUT would then render the objects within each class correctly. If one were again looking at an image of arm anatomy (figure 4), for example, and had clicked the mouse at pixel location $x, y = 118, 390$, then the system would look to see what color index was stored at that location. Let's say that the color index so selected was 55. The system would determine that this value falls between 32 and 63. The structure in question would therefore be classified as falling within the second range of index value. The system would then look to see that object #2 is the brachialis muscle and respond accordingly with text, sound or other images that explained the anatomical significance of the brachialis muscle.

MetaMapping images involves processing the image data so that each class of objects is rendered exclusively with index values that comprise a discrete subset of the available indices in the CLUT. This is the most labor intensive aspect of the process and is usually accomplished during the development of the interactive application. Although the object class subsets of indices are most easily grouped into contiguous ranges of entries in the CLUT (i.e., 0-31, 32-63, etc.), subsets may be defined by any consistent scheme, such as all-even vs odd indices, common multiples, etc.

The advantages of such an object identification scheme are several:

- 1) Data overhead is minimized. The data storage load for a single image and object identification pointer table is typically less than 1% larger than the original image file.
- 2) Identification speed is maximized. Rather than searching through hundreds of thousands of pointers at each identification event, as some bit-map methods require, only 256 values must be searched for an image with 256 colors.
- 3) Applications are resolution independent. Performance will not degrade as display resolutions increase. MetaMap based applications can be easily converted to operate on higher-resolution systems.
- 4) Object oriented images are produced. The MetaMapping process encodes the object identification mechanism into the images themselves, so they can be easily incorporated into newly-emerging object oriented program development systems.

5) The number of objects within each class of objects is limited only by the spatial resolution of the display. The system response time and data overhead are not affected at all by large numbers of objects within object classes.

The one disadvantage of palette segmentation indexing is that, as the number of object classes within an image increases, the quality of the rendered image will decrease. This is compensated for partly by the fact that objects within a given class will tend to have characteristic colors and therefore can be effectively optimized using methods such as Heckbert's Median Cut algorithm (Heckbert, 1982). Color dithering can also be used to improve object rendering in images with many object classes. Although dithering produces somewhat grainy appearing images at today's average display resolution, this effect will be progressively less noticeable as display resolution increases in the future. Another factor that will minimize the problem of multiple object class rendering is the current trend towards 12-bit, color mapped display systems capable of displaying 4096 simultaneous colors from 16 million possibilities. These display capabilities can currently be found in graphics workstations such as the Personal Iris, from Silicon Graphics, Inc., and the RS/6000 from IBM Corp. Such display systems allow many more colors for each palette segment.

A greatly simplified protocol for implementing the MetaMap process with grayscale images would be as follows. First the number of object classes in the image is determined. This number is then divided into the number of colors in the CLUT to calculate the average size of a palette segment. The dynamic range of grayscales in the image is then compressed to the palette segment size using linear interpolation. Image objects within each class are then manually outlined and assigned to a particular palette segment. All of the indices of pixels within a given object class are then multiplied by the palette segment number to map them to that segment of the CLUT. The R, G, B values for all of the entries in each palette segment are then reassigned to correspond to an optimal approximation of the original image object tonal range.

This would complete the remapping of the images. Next, an image object pointer table would be created for each image to link the object identities to associated data. This would then be integrated into an interactive application for database access, training or education.

Much more sophisticated implementations of the process are possible as well. An example would be to use 24 bit color image digitizations as the original data in an automated application development system. The application developer would outline all of the objects within an object class. The system would prompt for an object class name. This name would be stored in a table in memory. The portions of the image corresponding to this object would then be copied to a buffer in memory and erased from the original display. This process would be repeated until the original image has been progressively erased to the point that the pixels representing each object class reside in discrete buffers, with the number of buffers equal to the number of object classes. The color or grayscale attributes of the pixels within each object class would then be analyzed and an adaptive color quantization algorithm (Heckbert, 1982) would be applied to optimally map each object class to a discrete segment of the CLUT and to set the R, G, B values for each segment accordingly. The object class names would then be correlated to the CLUT entries and used as a pointer table to link object identities to other related data. The image would then be rebuilt automatically onscreen and saved to disk.

Several types of links could be created between each object and other related data. One could easily design a system for a 3-button mouse so that the left mouse button caused the program to display related text, the middle button could invoke an audio description, and the right could bring up a related image. Pressing all 3 buttons could cause a related program, such as an animation or a test program, to be executed.

The MetaMap process of palette segmentation indexing has been implemented in the creation of a prototype interactive atlas of histology (figure 4). This system allows a medical student to browse through

a database of micrographic images in a manner similar to the way they look at slides in the lab. At any point, a mouse click anywhere on an image will elicit further information concerning that image. A particular mouse button combination will activate the zoom feature, which will cause the display of the next higher or lower magnification image within the database that is linked to the selected object type. This system allows the student to interact with the computer in an intuitive manner much the same as that a teaching assistant is used in the laboratory. No distracting labels are visible within the images until the student asks for further information by clicking the mouse. This system can also easily be converted into a testing mode where the computer presents images in either a predefined or random sequence and prompts the student to identify named structures by pointing to them and clicking.

Another important application of the MetaMap process involves the display of multiple color images onscreen in multimedia systems. In systems that don't use MetaMapping, only one color mapped image can be displayed at a time. When another image is loaded, this causes the palette from the new image to replace the palette from the previous image. In this case the second image looks correct, but the first image is usually distorted. If a MetaMap-based system is used, each image is treated as a separate object on the display. As new images are loaded, their palettes are dynamically loaded into discrete segments of the CLUT. This allows new images to be added without affecting ones already displayed. The presence of an object-identity pointer table would allow individual images to be modified without affecting the colors of the other images. If the CLUT has 4096 index entries, then 16 images could be simultaneously displayed with each image using 256 colors. The optimal palettes for each image could be computed in real time on a powerful workstation, or they could be computed in advance for display on less expensive machines.

MetaMapping would also be of considerable value for linking object identity information to relevant data in 3-dimensional systems for Medical imaging and scientific visualization. In such a case, bit-map methods suffer to a greater degree from the same problems encountered in 2-dimensions. The addition of a third dimension to the dataset exponentially compounds the difficulties of slow object identification speed and large data overhead. MetaMap-based systems would not suffer either of these disadvantages.

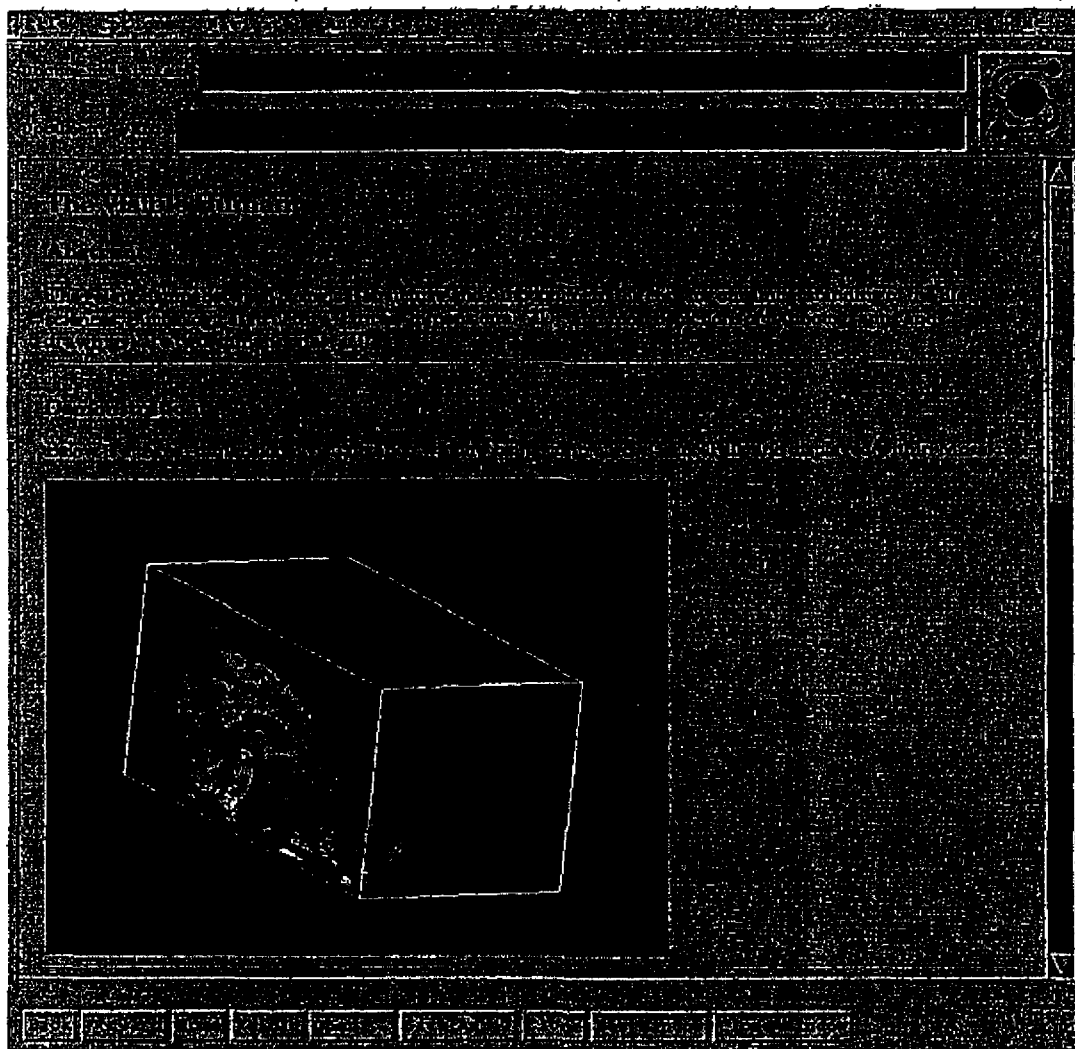
Palette segmentation indexing has additional areas of application as well. These include, but are not limited to the following: interactive catalogs; technical manuals; interactive educational systems for other image-oriented topics such as geography, the life sciences or art; intuitive interfaces for high stress military situations; archiving of aerial/satellite photographic data; diagnostic imaging annotation and archiving systems; cognitive rehabilitation for stroke and head-injury victims; foreign language training systems; and reading programs for preschoolers.

The MetaMap interactive digital video process is an attempt to redefine the role that computer graphic display images play in the function and purpose of application programs. It extends the concept of human-computer interaction to allow a user to interface directly with an image without the distractions of unwanted information or the mechanics of computer operation.

Metamapping allows useful interactive digital video applications to operate on inexpensive and commonly-available computer configurations and to utilize conventional economical mass-storage media. This process allows both direct-manipulation interfaces and hypermedia databases to begin to reach their full potentials. The tools necessary to produce truly innovative and effective interactive computer graphic presentations are now available. The most difficult aspects of generating such presentations, however, remain the initial conceptualization and design stages. Innovative tools such as the MetaMap process will not realize their fullest potentials until discipline-specific authors embrace these technologies.

The Visible Human Project

Importance of 3D data
needed for clinical, educational and scientific purposes
Loss of anatomists
NLM's Visible Human project -- Standard Man/Woman
Huge datasets -- problems in accessing/using such data
education
computer-aided diagnosis - deformable templates
VR
only a pilot -- the start of an ongoing process
applicability of embedded IRV



The Visible Embryo Project

Much of what is known about early human morphological development was discovered through careful examination of the anatomy of embryos of human and other species at a variety of stages of growth. Early investigators prepared meticulous collections of serial microscopic sections through these embryonic

E 006796

specimens. Through analysis of consecutive sections, and through handmade models built from these sections, pictures were generated of the development of the various shapes that would eventually transform into an adult organism. The accuracy and complexity of the models that could be generated, and, consequentially, the depth of knowledge about development that was learned, were constrained by the limitations of processing artifacts, human dexterity and a lack of adequate analytical tools. As a consequence, the study of development through the analysis of macro cellular structure seemed to hit a barrier that impeded significant further progress. Since then, advances in technologies for molecular biology have spurred many investigations into the molecular mechanisms of development. The important successes of these approaches have greatly shaped the study of developmental biology, and have shifted the primary focus of most investigations from macro cellular scales of interest to detailed studies of genetics, micro cellular environments and macromolecular functions. These many studies have, over the years, yielded massive amounts of information of interest to developmental investigators. Indeed the problem of accessing, synthesizing, and using this information has often been likened to trying to take a drink from a firehose.

A new branch of information science, or informatics, has sprung up to address the problems of analyzing and interpreting such massive volumes of data. Biomedical researchers are particularly needful of advances in this area. A good example of this situation concerns the Human Genome Project. Although there is no question that the creation of a complete map of human genetic sequences will be an immediate boon to many areas of research, the ultimate usefulness of such voluminous and highly detailed information resources will be severely constrained without an adequate macro cellular context within which to use the data.

Recent advances in visualization technology are beginning to allow investigators to break through these previous limitations and investigate macro cellular structure in search of universally-applicable rules for pattern formation and shape development in organisms. By applying these new technologies to the existing archives of cross-sectional image information that exist in the literature and in collections around the world, we can tap into an enormous amount of new information that can be extracted from these databases. Towards this aim, we have been working for several years on strategies to allow new information to be extracted from existing "Image-centric" databases. This work initially began as an assortment of inquiries into somewhat disparate topics, but quickly congealed into a single project, "The Visible Embryo Project," that consolidates these previously scattered activities into one coherent endeavor.

Drawing from the world renowned Carnegie Collection of Human Embryology, the Visible Embryo Project begins with the creation of a large database of 3-dimensional image data of human embryos, covering all of the accepted developmental stages of human developmental stages of human development through the first trimester, as well as additional stages extending through the fetal period. Collections of light-microscopic serial cross-sections through preserved human embryos are used as the input data for the project. Since the specimens in these collections weren't originally prepared with the intention of 3-D computer reconstruction, fiducial markers were not introduced by the technicians when the tissue was processed decades ago. Among the earliest problems that we addressed in the Visible Embryo project, then, was the creation of methodologies for "reverse engineering" the morphology through analysis of information within the images of consecutive serial sections. These sets of consecutive image series were digitally processed and passed through a series of analytical procedures which created a set of images that were registered with each other, so that a voxel dataset of the 3-D structure of each embryo's anatomy was produced, similar to the voxel data produced by a conventional MRI scanner.

Various strategies need to be developed to enable these 3-D datasets to be quantitatively analyzed and compared in order to extract new knowledge concerning the mechanisms of human development. No single investigator has the skills, resources or time to address all of the research issues necessary to accomplish all of the goals of this work. We have therefore brought together a multidisciplinary research team of scientists in fields including anatomy, cell and developmental biology, surgery, image data acquisition, biomedical engineering, software development, medical imaging and visualization, educational technology, virtual reality, and many other fields in biology, medicine, engineering, and

computer science, to initiate the conception, design, and initial implementation of a national computational resource for developmental biologists. This team of collaborators across the country is working on the creation of tools for registration, segmentation, visualization, quantitative analysis & animation of embryonic anatomical developmental phenomena. One important consequence of this work will be the creation of a "morphological map" of gene expression. The three-dimensional models created by this project will provide workers in developmental genetics with a uniform "anatomical coordinate space" within which to report their findings and through which to compare findings across a large number of studies.

This project will result in the creation of what we call a national "metacenter" for developmental biology. The resulting national network of computational and data resources will serve the dual purpose of providing a testbed for new technology development in high performance computing and communications, as well as creating powerful new tools for the developmental biology research community.

This effort will establish an on-line computer-based multimedia information resource center which is physically distributed through high-speed networks among over 20 collaborating institutions across the nation, but in such a way that the distribution of resources is invisible to the user. Researchers, clinicians and students will be able to access the metacenter through any low-end Mac, PC or UNIX workstations which have Internet connections. Once on-line, the user will be able to access an enormous amount of information on developmental biology, including real time interactive control of the various 3D reconstructions of embryos from the Carnegie collection. The user doesn't even need to be aware of the fact that the images seen on the screen are being dynamically computed by a network of supercomputers distributed across the country. The power of this approach will allow the worldwide delivery of a complete medical school curriculum in human embryology through on-line access to the system.

The task of integrating access to such massive information and computational resources is nontrivial. Just one embryo from the 650 serially-sectioned specimens in the collection can yield as much as a terabyte of anatomical volume data (a 20mm specimen, sectioned at 5 microns and digitized at a resolution of 8000x8000 pixels/section at 36bits RGB produces 1.073 TB of voxel data). It is clear that no single workstation, or even supercomputer, can manipulate, process and analyze such a large quantity of data as a single unit, much less perform computational operations on a database of hundreds of such datasets. For this reason, the Visible Embryo team members at UCSF have developed tools to allow integrated Internet access (through NCSA Mosaic) to remote volume visualization engines which can distribute computation across a large number of graphics supercomputers connected by high-speed networking. This allows the integration, through NCSA Mosaic and the World Wide Web, of text-based, image, audio, and video data with real-time interactive control of high-performance visualizations embedded within Mosaic documents. We are also exploring the potential of using this technology for delivering interactive access to virtual reality applications through the Internet. The success of these efforts will allow widespread access to highly accurate and complex virtual reality simulations of human developmental anatomy, using inexpensive workstations and personal computers.

As biology moves into the information age, biologists can no longer sit idly aside and wait for the computer scientists and engineers to solve their problems with respect to dealing with the deluge of information entering the field at a constantly-increasing rate. It will be only through taking a proactive role in as part of a multidisciplinary team that biological researchers will be able to influence the directions of progress in the information sciences in ways that will support advances in the study of biological structure, function and development.

E 006798

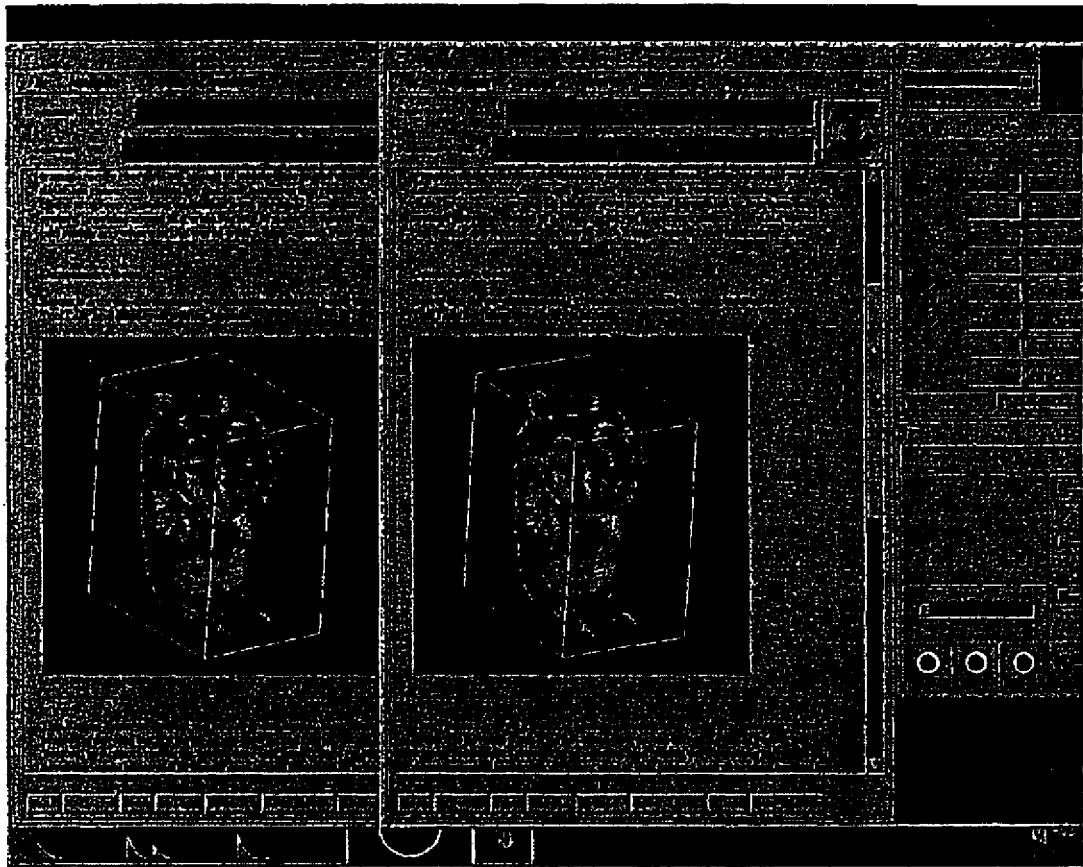


Figure 1: A stereo-pair illustration of interactive real-time control, embedded within an NCSA MOSAIC document, of a 3-dimensional volume reconstruction of human embryonic anatomy, showing a 7-week old human embryo found in the Carnegie Collection as 1997 serial cross sections. This technology was developed by the Center for Knowledge Management at the University of California, San Francisco, and was demonstrated there in November, 1993.

Molecular/genetic visualization

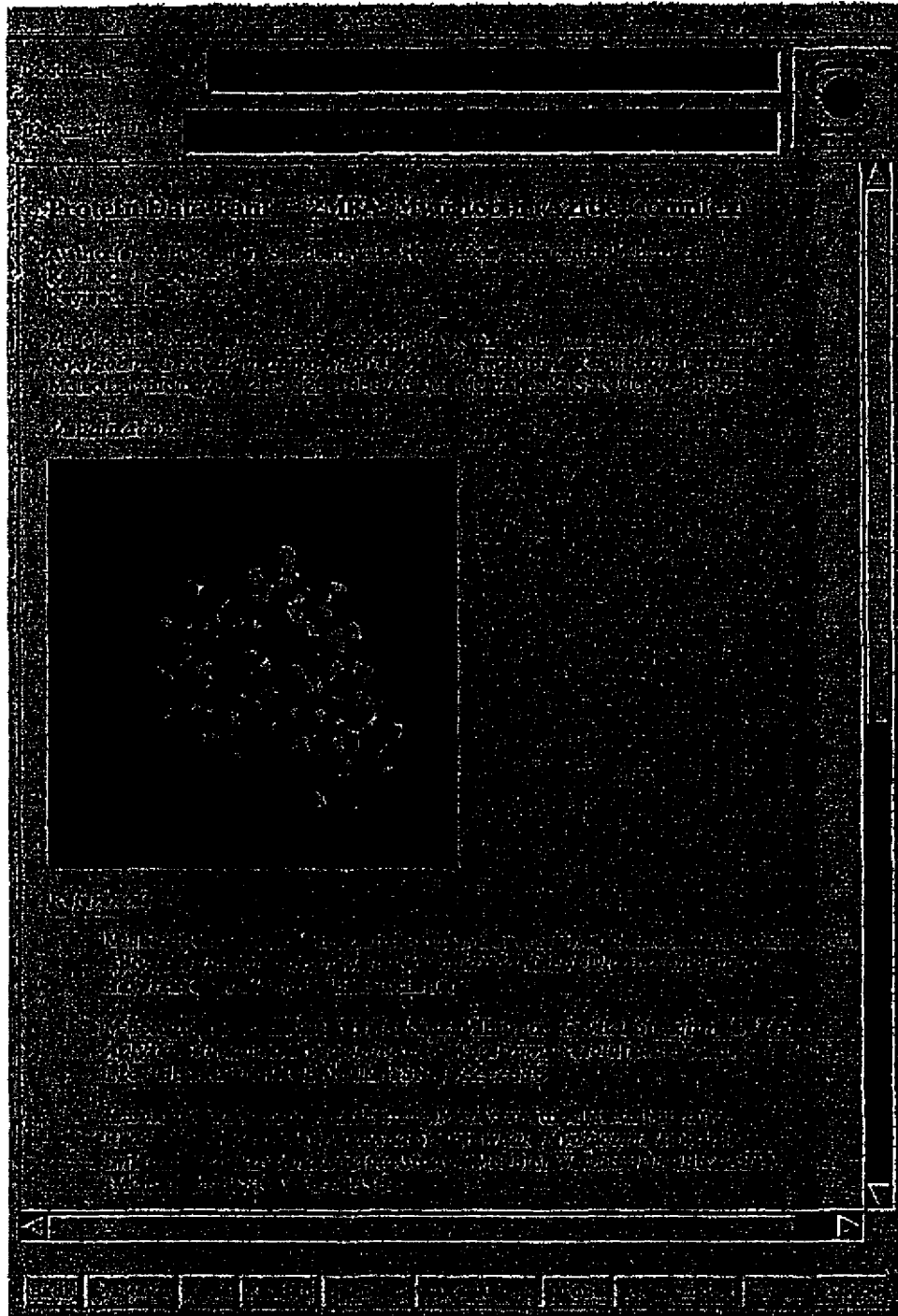
The recent revolution in genetic engineering is having an immense and growing impact on both science and industry and is likely, over the next several decades, to have consequences comparable to the electronic revolution of the present and previous decades. Two of the key unsolved problems in this burgeoning field are (1) the prediction of the three dimensional structures of proteins and other polymers from knowledge of their monomer sequences and (2) correlation of three-dimensional structure with catalytic activity. These closely related fields of research are both of fundamental scientific importance and considerable potential economic benefit.

DNA, the double helix, carries the genetic code and is the prime raw material of the genetic engineer. The coding sequences of DNA can now be determined so rapidly that the primary sequences of several tens of thousands of proteins are known. This number is increasing very rapidly. But the ultimate *function* of a protein depends on its three dimensional structure, and the three dimensional structures of only a few hundred proteins are available from long, laborious and expensive X-ray diffraction analyses. New structures appear at the rate of only a few dozen a year. This rate is hardly increasing, and certainly cannot match the rate of determination of genetic coding sequences. Although techniques such as

multidimensional NMR are now contributing valuable structural data to our knowledge base, no new methods are on the horizon which will provide the orders of magnitude increase in productivity required. Thus, it is critically important for scientists to utilize the existing database of protein structures with the most efficient, informative, and mentally stimulating means possible. While a couple of decades ago the case often had to be made that interactive computer graphics provided a real advantage to scientists and was not merely a generator of pretty pictures, it is now recognized by even the early critics that interactive computer graphics is a routine and useful tool in molecular modeling. Computer graphics requires numbers to generate its pictures, but the main value lies not in the numbers themselves, but in the insight gained when the graphics is used as a window on the process or system being studied. Computer-assisted insight is not always quantifiable, but is nonetheless real. High quality interactive computer graphics, with a well-designed graphical interface for the scientist user whose main interest is in chemistry or molecular biology, is an *idea generator*.

The UCSF Computer Graphics Laboratory has had a long history in designing, building and disseminating interactive molecular modeling tools. The laboratory receives the majority of its funding from the National Institutes of Health, National Center for Research Resources (grant RR-1081). The specific aims of the Laboratory are:

- To design, build, integrate and disseminate computation and visualization tools for structural biology and volumetric biomedical imaging
- To apply these software tools to problems in medicinal chemistry and molecular biology with particular emphasis on elucidation of molecular structure using nuclear magnetic resonance, studies of molecular interaction, drug design and protein engineering
- To apply these software tools to problems in medical radiology and human developmental anatomy....
- To assemble hardware with which the software can best be utilized
- To provide access to, training in, and assistance with this software and hardware for outside users
- To disseminate as documented source code the software developed at the resource to allow others to use the software both for research application and as a starting point and training tool for their own developments



One of the key software packages that has been developed at the UCSF Computer Graphics Lab is MidasPlus (Molecular Interactive Display and Simulation). This is an interactive graphics program designed for the efficient display and manipulation of proteins and nucleic acids, including very fast van der Waals surface calculations, space filling images with shadows cast from multiple light sources, and

E 006801

real time "ribbon" diagrams for depicting protein secondary structure and nucleic acid helices and base pair planarity and directionality. MidasPlus emphasizes highly interactive display and manipulation, with a data structure designed for interactive selection, manipulation and docking of drugs and receptors. More than 500 papers have been published based on work done with MidasPlus and its predecessor, Midas. Through a licensing program developed by the Regents of the University of California, MidasPlus has been distributed to over 300 other laboratories. MidasPlus currently runs on the Silicon Graphics "IRIS" and "Indigo" family of workstations, Digital Equipment's "Alpha AXP" workstations equipped with an optional 3-D graphics card, and, with more limited capabilities, on the NeXT color workstation.

Linking information

Toni's stuff

Online Publishing

Despite the significant growth in digital networks and in the number of publications in digital form on those networks, there have been few attempts to create a comprehensive set of digital authoring tools which would allow authors, editors, and reviewers to collaborate in a networked environment to produce fully integrated documents. To be effective, these tools should be accessible and easy to use and be capable of performing all of the tasks--manuscript creation, editorial comments and reviewer tracking, copy editing and a final formatting step prior to publication--that are necessary for publishing, in print, on a network, or in a database of articles.

Authoring packages have been around for some time, but they vary by discipline. Mathematicians routinely use complex procedural coding such as eqn in Unix, and TeX or LaTeX on a variety of platforms. Generally, these author packages decline in sophistication as one moves away from computer science and mathematics to other fields such as biology and chemistry. The authors in these disciplines tend to use macro packages for specific publications, which are supplied by the publisher. Furthermore, these macro packages tend to be limited to the more popular word processing programs such as WordPerfect or MicroSoft Word. Disciplines such as medicine are even less sophisticated and very few authors use any tailored software. At the editorial level most journal publishers have customized software to aid editors in the management of the editorial process: that is to maintain reviewer lists and track the status of manuscripts. These programs typically feed this information into a database that can be used to generate statistical reports and project publication schedules. Very few tools exist, however, that assist in the editing and reviewing of electronic documents that contain complex mathematics, tabular material, graphic data, and video.

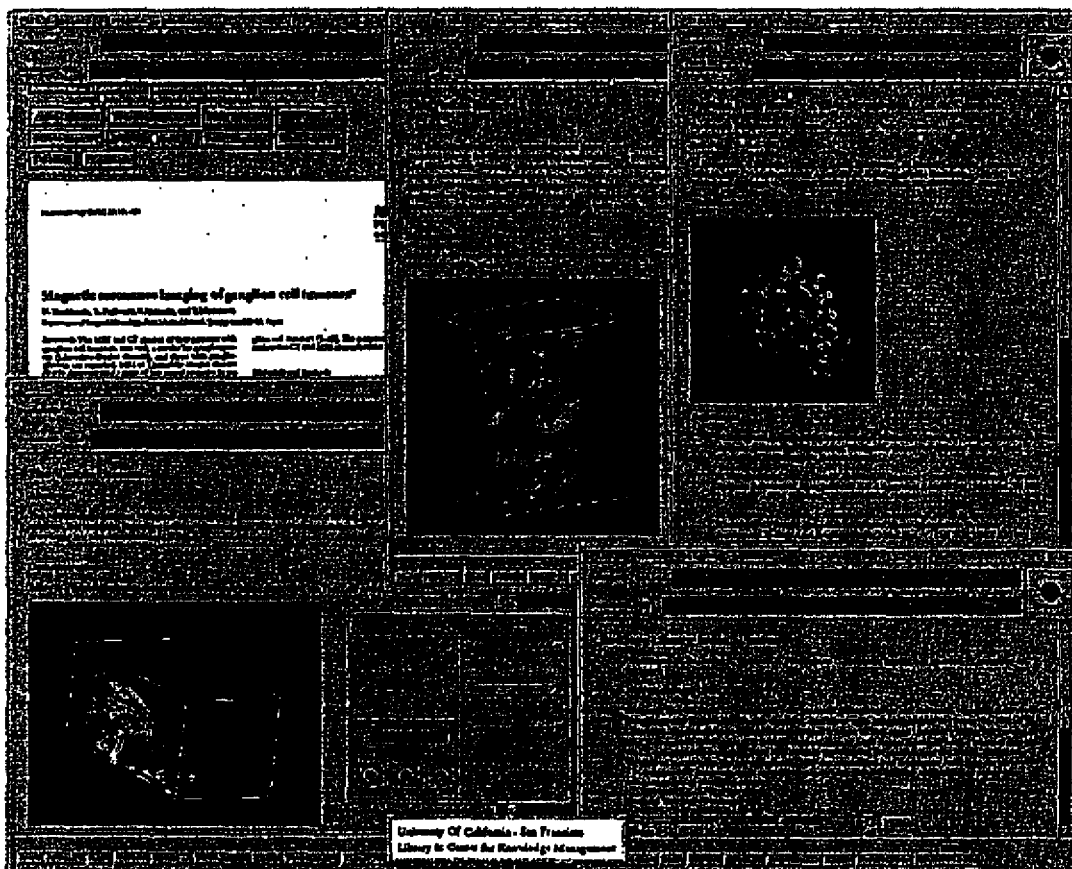
In the proposed system, we will develop a set of software tools for: (1) creating in a WYSIWYG environment a manuscript that includes features such as HTML+ coding with embedded links to visualizations, datasets or to the author's original dataset. (2) assisting the journal editor in the efficient and thorough execution of the peer review process. The goal here would be to preserve the anonymity of the process while enhancing the speed and depth of communication between authors, editors and reviewers. (3) formatting a reviewed and accepted manuscript in preparation for publication (be it in print or electronic form). This formatting program must support all different content (e.g. text, tabular, graphics) and technical (e.g. TeX, eqn, tbl) elements/requirements that may be present in the document.

An Integrated System

Integration is the key

MOSAIC allows that integration

Common user interface, platform independent, distribution of computational load, free client, ubiquitous (de-facto standard)

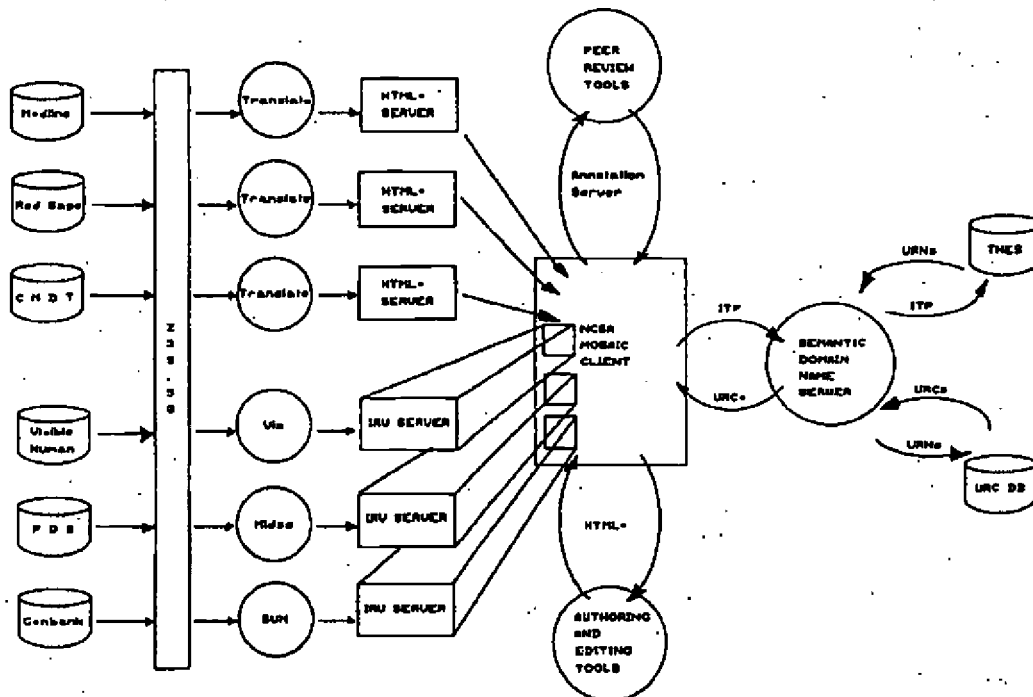


Perhaps the greatest benefit of this research is that it will provide national leadership in a number of areas critical to the success of the coming National Information Infrastructure. The presently-proposed project will begin to produce a body of information that others can use in their efforts to plan the on ramps, off ramps, bridges, and toll booths for the next century's national "information superhighway."

Work Proposed:

General System Strategy:

Below is a diagrammatic overview of the structure of the proposed system:



- Definitions:**
- HTML+:** Hypertext Mark-up Language -- This is the language that World Wide Web databases are encoded in, and that MOSAIC interprets.
 - IRV Server:** UCSF CKM's Interactive Remote Visualization Server -- This allows interactive real-time visualization tools to be embedded into MOSAIC documents.
 - Vis:** UCSF CKM's distributed remote volume visualization tool
 - Midas:** UCSF CGL's molecular visualization package
 - SVM:** Sequence Visualization Module -- An as-yet unnamed tool for graphical display of genetic sequence data.
 - ITP:** Informal Text Phrase -- A user-entered search term, or a word or phrase of text that the user highlights from within a document.
 - URN:** Universal Resource Name -- A persistent, location-independent identifier for an object.
 - URL:** Universal Resource Location -- The address of an object. It contains enough information to identify a communications protocol and retrieve the object.
 - URC:** Universal Resource Characteristics -- Any combination of one or more URNs or URLs with meta information (e.g. author, format, compression method).

Explanation of system diagram:

The system will draw from a number of fundamental databases including bibliographic data (Medline) in the form of MARC records, journal publication data (Red Sage) in the form of SGML header and Postscript files, encyclopedic reference text data (CMDT) stored in an object-oriented SGML database, volumetric anatomical data (Visible Human Project) stored as NCSA HDF datasets, protein structure data (Protein Data Bank) stored as PDB files, and genetic sequence data (Genbank) stored as compressed ASCII strings (? I'm guessing about Genbank).

These databases will reside behind a Z39.50 interface layer which yields, to the requesting client, the respective dataset in its native form. This data then goes through a translation layer where the data is either translated directly into HTML+ (Medline, Red Sage, CMDT) or loaded into a native-data visualization tool (Visible Human, PDB, Genbank). The HTML+ code is then passed to a set of HTML+ servers, which can be browsed by the MOSAIC client. The visualization data is handled differently. The graphical I/O of the relevant visualization tool is passed to an interactive remote visualization (IRV) server, which handles both mapping of the display output from the visualization tool onto embedded live-visualization windows within the MOSAIC-browsable HTML+ documents, as well as capture of user-entered mouse and keyboard events within the visualization windows and transmission of those mouse and keyboard events back to the relevant visualization tools. The user, browsing the system with the project's enhanced version of the MOSAIC client, is presented with data and visualizations derived from these various databases, yet embedded into coherent, multimedia MOSAIC documents.

For multimedia documents that have been explicitly pre-composed, the linking of these various data resources can take the form of universal resource names (URNs) that are encoded as tags into the HTML+ documents. This is passed to the system's semantic domain name server, for resolution of the information object's location and retrieval means. The URNs are used as indices in order to look up the relevant universal resource characteristics (URCs) in a URC database, which yields the universal resource location (URL), or physical address, of the information object in question.

Semi-automatic means will be provided for a user to search for arbitrary information objects on the system by either keying in a search word or phrase, or by highlighting a not-already-hyperlinked section of text that (s)he happens to be viewing within the MOSAIC client at the time. This informal text phrase (ITP) is then passed to the semantic domain name server, which passes it on to a universal resource thesaurus (which will incorporate elements of the NLM's UMLS system). The thesaurus compares the ITP to its database of terms and phrases and returns a rank-ordered list of URNs that are likely to match the object in question. These URNs are then passed to the URC database for resolution of URLs that point to information objects on the Internet that are most likely to match the ITP that the user employed to initiate the search. The user is presented with a rank ordered set of textual descriptions of likely matches which are hyperlinked, via their URLs, to the data in question. Clicking upon a selection from this list loads the related data into the relevant visualization server (IRV) or HTML+ server, and a second MOSAIC window pops up to allow viewing or interaction with that dataset.

A set of authoring and editing tools will be designed to allow the interactive WYSIWYG creation of HTML+ documents, as well as allowing the embedding of visualizations, etc., which can be created using the interactive remote visualization tools, and which can use data from the various scientific databases mentioned above. Alternatively, the author can use his/her own datasets, which would be uploaded to an Internet-accessible World Wide Web server. The journal editor can use the same set of tools to edit submitted articles and to communicate changes to the text with the author. This, of course, would occur in a private, access-controlled, area of the system, so that confidentiality of the material to be published can be controlled.

It is important to understand that there are two distinctly different types of links involved in this system...

Other private, access-controlled HTML+ servers will be used to administer the peer review process. A modification of NCSA's MOSAIC-based group annotation server will be developed to allow the journal editor to exercise precise control and documentation of each reviewer's comments and suggestions.

Detailed Work Plan

Foundation Databases

Following is a sketch of the workplan for the Z39.50 interface to RightPages in the first year of the grant. I have not included development of the basic SGML technology that AT&T will be funding independently:

Z39.50 interface to RightPages (year 1)

#	Months	Task
2		Design (protocol translation, client and server components)
5		Implementation of all components
2		Component testing
1		Integration into RedSage system and testing
2		Packaging and documentation (both system and RuserS)

Our plan is to have our contractor contribute to the implementation of the various SGML databases: CMDT, URC, and the semantic domain server and thesaurus components. Are you specifying workplan details for those efforts? I guess I am not sure at the moment how long they will take without learning

more about the contributions of the other collaborators, plus being further along in the development of the underlying OO SGML stuff itself. Generic database development, depending on complexity, might look something like this: # Months Task

2-4		DTD definition, plus any additional class definitions
0-4		implementation of any additional class methods etc.
0-2		implementation of any additional content prep or import tools and components
1-2		initial document data load and testing

Translation Modules

MOSAIC-based Browser-Editor

Interactive Remote Visualization Servers

MOSAIC Extensions**Browsing vs detailed views****Vis****Midas**

Primary goals of the molecular visualization component of the presently-proposed project include:

- To develop an "Interactive Molecular Visualizer" for display and manipulation of molecular structure data, based on UCSF CGL's MidasPlus package, which will interface with the enhanced MOSAIC client developed at UCSF's CKM. To use a client/server architecture in order to maximize the number of hardware platforms capable of interactively viewing three dimensional molecular structure:
- To convert the existing MidasPlus User's Manual (130 pages, including an extensive index) into Hypertext Markup Language (HTML), so that an on-line hypertext-based user's guide is available;
- To collaborate on the design and implementation of a Semantic Domain Name Server with the specific goal of integrating protein structure data sources such as the Protein Databank [5] into the MOSAIC HTML framework. We intend not only to provide direct access to the PDB database (at least for experimental purposes), but also to provide a mechanism for a server to locate and access on-line protein structure data remotely as a part of "following" an HTML link.
- As resources permit, to integrate additional structure and sequence visualization tools being developed at the Computer Graphics Laboratory into the CKM enhanced MOSAIC client:

To interface MidasPlus and the enhanced MOSAIC client, CGL will collaborate with the CKM on two key project components: the requirements analysis and formulation of design specifications for data processing to be performed locally on a MidasPlus/MOSAIC client versus that processing to be performed on a more powerful server (i.e. division of functionality between client and server); and the design of the communications protocol which will be used between the client and server. The client-server division is an important software architectural decision because it greatly influences the performance optimization possibilities. By properly dividing the work load between client and server, it is possible to reduce simultaneously the data communications between the applications as well as the computational load on each end. A careful design should permit the use of "low-end" hardware platforms such as Intel 486-based PC's for clients, and thus make interactive three dimensional molecular visualization available to a significantly larger fraction of the scientific user community. The actual communications protocol used between client and server is also an important aspect of the software architecture. A good protocol will facilitate the software development, while a poor protocol will create more work during implementation and reduce efficiency.

Molecular structure data generally consist of a set of atomic coordinates rather than the regularly spaced sample points that an MRI scanner produces; hence the images generated from molecular structure data are more suitably described using an algorithmic format such as OpenGL [6,7]. Once a suitable data exchange format between the molecular visualizer and MOSAIC is chosen, we plan to implement the visualizer as a remote service available to the enhanced MOSAIC client. The implementation will also include performance analysis to determine whether interactive response is sufficient and to locate bottlenecks where improvements are needed

Of critical importance to the successful deployment of our new molecular visualizer for MOSAIC is the availability of high quality documentation for the scientist user. Since it is desirable that even casual users be able to efficiently manipulate and control the molecular structures that they see, we need a user "manual" that may be browsed as a standard MOSAIC HTML+ document. As stated

above, we plan to convert our existing MidasPlus User's Manual (see appendix) written using UNIX "troff" formatting commands into HTML. Although this task does not present any particular challenges intellectually, it will nevertheless be demanding since the schedule calls for us to begin this project before the new authoring and editing tools are completed. We must still be able to generate a printed version of the manual (using PostScript) in addition to the MOSAIC-viewable document, so there are several issues that must be dealt with if we are to be successful.

RE: Semantic Domain Name Server...

Semantic Domain Name Server

**Kazic's stuff
URN Thesaurus
URC Database**

Authoring and Editorial Tools

Work Plan for Authoring/Editing Tools

Phase I (6 months)

Interview and hire scientist for technical work Under the direction of Bob Badger and Craig Van Dyck, Springer-Verlag New York will interview and hire a computer scientist who has sufficient background and knowledge to develop authoring/editing tools that are suitable and consistent with the aims of the project.

Installation of hardware and software

Under the direction of Craig Van Dyck, the scientist will install and configure hardware and install Galaxy, Avalanche and SoftQuad software systems. Thorough testing of these systems connecting with the University of California, San Francisco (UCSF) and AT&T Bell Labs through Internet will be performed at Springer-Verlag New York's office.

Phase II (12 months)

I Develop author package and sample content

--Needs analysis: Determine the specific needs of the authoring package by becoming familiar with the publishing process; and by understanding the requirements of UCSF's Center for Knowledge Management and the National Center for Supercomputer Applications' HTML+ coding.

--Develop and review the specifications of the authoring program with all of the principal project investigators.

--Evaluate existing programs, tbl and other editors, to determine which could be adapted or modified. It is very probable that existing editors in Unix or others could be modified to fulfill most but probably not all of the program specifications.

E 006809

PH_001_0000787110

--Develop final program.

--Work with Washington University, UCSF's CKM, and UCSF's CGL to develop sample content.

--Work with UCSF and Washington University to test the program with sample content.

Phase III (12 months)

1 Develop journal editor and peer review program.

--Needs analysis: assess requirements of journal editors' work flow; UCSF's hardware requirements; and information about journal editing procedures from the staff of Springer-Verlag New York.

--Develop specifications for program. Revise specifications with Springer New York staff and all key project participants.

--Write a test program with author package and content.

Phase IV (18 months)

1 Adapt an existing commercial program to format content from SGML-based URC database.

--Access in cooperation with Bell Labs and UCSF's CKM the requirement for a program to store documents in the database.

--Develop and review specifications of the formatting program.

--Assess commercial programs that meet specifications.

--Assess storage and retrieval with sample documents.

HTML+ Authoring, with Embedded Objects
Annotation Server
Peer Review
Editorial Process

Testbed Facility

WWW server

Beta Testing

Evaluation

Expected Accomplishments

Timetable

Year 1

Year 2

Year 3

Year 4

Plans for Preservation, documentation and dissemination of results

Set up a WWW server

Documented source code

Papers presented at meetings

Results from Prior NSF Support

Siggraph '92 -- Showcase demonstration funded by a combination of NSF and NIH money:

3-D Visualization Tools

Software tools were developed to allow the interactive three-dimensional visualization of the embryo reconstruction in real time. Figure 3 shows the display of the application as it appeared at the

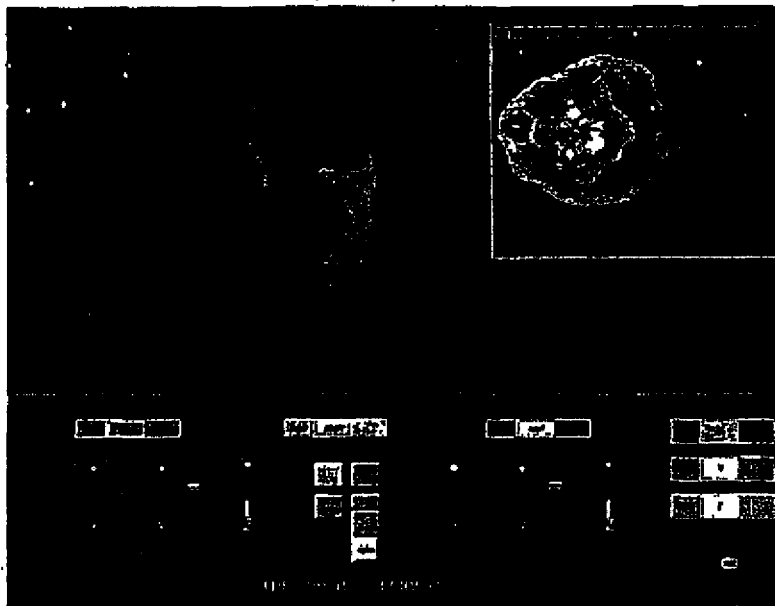


Figure 3: Screen image of the 3-D visualization program demonstrated at SIGGRAPH '92, in Chicago. This application was distributed between a Silicon Graphics Crimson VGXT workstation and a Convex supercomputer.

SIGGRAPH '92 conference in Chicago (Doyle, et al., 1992). The left of the screen shows a surface-based model of the embryo's exterior. This model was built from data which was derived, through three-dimensional interpolation, from the original embryo dataset. Two-hundred volume slices of the embryo (stored as texture maps) can be interactively displayed at this lower resolution while the model is rotated freely in three dimensions. A cutting-plane can be seen to intersect the surface-based model. This cutting plane can be interactively controlled to intersect with the embryo model at any arbitrary angle and position. To the right of the screen, one can see a window that displays a high-resolution image of the oblique section through the embryo as indicated by the interactive cutting plane. In order to maintain the quick response needed for effective real-time interaction, the computational load of this application was distributed so that the interface panel, seen at the bottom of the screen, and the 3-D surface model were running on the CPU of the Silicon Graphics workstation. Computation of the high-resolution oblique section image displayed in the right window took place on the Convex supercomputer. Both of these operations occurred simultaneously, communicating through a high-speed fiber optic network.

Virtual Reality

The surface-based embryo-model described above was also implemented within a virtual reality environment, called "The Cave," at the 1992 SIGGRAPH conference. The Cave was a 10' x 10' room made up of back-projection screens upon which were projected stereo views of three dimensional data

that a viewer, wearing LCD stereo glasses and a 3-D tracking device, could move around in as the system tracked his or her motion. One could walk around the data and receive the sensation that one was actually "in" the computer graphic environment. This system was also shown at the Supercomputing '92 conference and the 1992 meeting of the Radiological Society of North America.

Bibliography

1. Bush, V.: As We May Think, in *The Atlantic Monthly*, July 1945: 101-108.
2. Campbell, G., T. DeFanti, J. Fredericksen, S.A. Joyce, L.A. Leske, J.A. Lindberg and D.J. Sandin: Two Bit/Pixel Full Color Encoding. *Proc. ACM SIGGRAPH*, August, 1986. 20/4:215- 223.
3. Doyle, M.D.: Method and Apparatus for Identifying Features of an Image on a Video Display. United States Patent #4,847,604, July 11, 1989.
4. Doyle, M.D.: The Interactive Digital Video Interface. in *Spatial Display and Spatial Instruments: Proceedings of a conference sponsored by NASA Ames Research Center*. NASA Conference Publication 10032, 1989. pp 24-1 to 24-4.
5. Foley, J.D.: Interfaces for Advanced Computing. *Sci. Am.*, October, 1987. pp 127-135
6. Heckbert, P.: Color Image Quantization for Frame Buffer Display. *Proc. ACM SIGGRAPH*, July, 1982: 297-307.
7. Hutchins, E.L., J.D. Hollan and D.A. Norman: Problems with Direct Manipulations, in *Readings in Human-Computer Interaction*. Ed. R.M. Baecker and W.A.S. Buxton, 1987. pp 468-471.
8. Murch, G.M. and J.M. Taylor: The Effective Use of Color in Graphics. in *Color in Computer Graphics: ACM SIGGRAPH Tutorial Notes*. August, 1988: 1-33.
9. Nelson, T.: Managing Immense Storage, *Byte*, Jan. 1988: 225- 238.
10. Schneiderman, B.: The Future of Interactive Systems and the Emergence of Direct Manipulation, Behavior and Information Tech., 1982. 1:237-256.
11. Schneiderman, B.: *Direct Manipulation: A Step Beyond Programming Languages*. IEEE Computer, August, 1983, New York: IEEE Computing Society.
12. Silverstein, L.D.: Human Factors for Color Display Systems: Concepts, Methods, and Research, in *Color and the Computer*. H.J. Durrett (Ed). Academic Press, NY, 1990
13. Thursh, D.R., F. Mabry and A. Levy: The Knowledge Access, Management and Extension System in Pathology. in Levy, A.H, and B.T. Williams (eds) *Proceedings AAMSI Congress*, 1985. pp 365-406.

14. Thursh, D.R., F. Mabry and A. Levy: Computers and Videodisks in Pathology Education: ECLIPS as an Example of One Approach, *Human Pathology*, 1986, 17/3:216-218.
15. Glickman, J., W. Eicholzar, D. Screnci and L. Glass: PC- Based Imaging, *Biomed Imaging & Comm.*, November, 1987, pp B12-B16.
16. Osgood, D: The Difference in Higher Education, *Byte*, 1987, 12/2:165-183.

858 PH Ex. 8



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 04/18/2008

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

EXAMINER

ART UNIT PAPER NUMBER

DATE MAILED: 04/18/2008

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



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

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

MAILED

APR 18 2008

CENTRAL REEXAMINATION UNIT

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.



Joseph R. Pokrzywa
Primary Examiner
Art Unit: 3992

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 01 October 2007. 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 type="checkbox"/> _____. |

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 4, 5, 9 and 10 are patentable and/or confirmed.
4. Claims 1-3 and 6-8 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)

DETAILED ACTION

Summary of Current Proceedings

1. Claims 1-10 of U.S. Patent Number 5,838,906 (hereafter "the '906 Patent") remain subject to reexamination.
2. A previous reexamination certificate for the '906 Patent (in reexamination number 90/006,831) was issued June 6, 2006.
3. Within the current reexamination proceeding, an Office action dated 7/30/07 rejected claims 1-10 with the references of "A Brief Overview of the VIOLA Engine, and it's applications", written by Pei Wei, noted as "Viola", and rejected claims 1-3 and 6-8 with the reference of Cohen et al. (U.S. Patent Number 5,367,621), noted as "Cohen", when viewed with "Introducing NCSA Mosaic", noted as "NCSA Mosaic".

Oath/Declaration

4. The Patent Owner submitted arguments on 10/1/07 and submitted a Declaration under 37 CFR 1.131, which establishes the invention prior to August 16, 1994, being the date utilized as the publication date of the Viola reference noted above.

5. With this, the Declaration filed on 10/1/07 under 37 CFR 1.131 is sufficient to overcome the Viola reference utilized in the rejection noted in the Office action dated 7/30/07. The examiner notes that the Viola reference lists on the first page, titled "The Viola Home Page" (being TT 05441), that "Vintage Viola screendumps" are included from "applications of the old viola (1991)". However, the examiner cannot find any other documents in the record that disclose the specific teachings of the Viola browser, as described in the previous Office action dated 7/30/07, that establish a date prior to August 16, 1994. Therefore, the rejection of claims 1-10, as indicated in the previous Office action under 35 U.S.C. 102(e), as being anticipated by Viola, has been withdrawn.

Response to Arguments

6. Patent Owner's arguments filed 10/1/07, with respect to the Cohen reference, have been fully considered, but they are not persuasive.

7. First, regarding much of the Patent Owner's arguments, which beginning on page 1, the Patent Owner states that the claim construction set forth in the Markman ruling in the related litigation, which was affirmed by the U.S. Court of Appeals for the Federal Circuit, is utilized in the subsequent remarks. For instance, the Patent Owner argues on page 10 that Cohen fails to expressly teach of the feature of "interactive processing", whereby the Patent Owner interprets the limitation as being the processing of the user utilizing a mouse or keyboard or similar device, to change the structure or presentation of an object. However, the examiner notes that this is not the proper standard for claim construction during examination before the Office, as recognized by the Courts.

8. In this regard, MPEP 2111 [R-5], under the title, "CLAIMS MUST BE GIVEN THEIR BROADEST REASONABLE INTERPRETATION" states:

During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." >The Federal Circuit's *en banc* decision in *Phillips v. AWH Corp.*, 415 F.3d 1303, 75 USPQ2d 1321 (Fed. Cir. 2005) expressly recognized that the USPTO employs the "broadest reasonable interpretation" standard...

9. MPEP 2111 continues,

See also *In re Morris*, 127 F.3d 1048, 1054-55, 44 USPQ2d 1023, 1027-28 (Fed. Cir. 1997) (The court held that the **P.T.O. is not required, in the course of prosecution, to interpret claims in applications in the same manner as a court would interpret claims in an infringement suit.** Rather, the “P.T.O. applies to verbiage of the proposed claims the **broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art,** taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in applicant's specification.”).

The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. *In re Cortright*, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999).

[Emphasis added].

10. Continuing, MPEP 2111.01 [R-5] states under heading “I. THE WORDS OF A CLAIM MUST BE GIVEN THEIR “PLAIN MEANING” UNLESS **>SUCH MEANING IS INCONSISTANT WITH< THE SPECIFICATION”:

>Although< claims of issued patents are interpreted in light of the specification, prosecution history, prior art and other claims, this is **not the mode of claim interpretation to be applied during examination. During examination, the claims must be interpreted as broadly as their terms reasonably allow. *In re American Academy of Science Tech Center*, 367 F.3d 1359, 1369, 70 USPQ2d 1827, 1834 (Fed. Cir. 2004) (**The USPTO uses a different standard for construing claims than that used by district courts; during examination the USPTO must give claims their broadest reasonable interpretation >in light of the specification<.**). This means that the words of the claim must be given their plain meaning unless **>the plain meaning is inconsistent with< the specification.[Emphasis added].

11. If the Patent Owner wishes the claims to be narrowed based upon the Court's construction of the claim terms, then the claims must be amended accordingly. The case law above makes clear that claims during examination are interpreted broadly not narrowly. To incorporate the limitations into the claims would countermand the case law prohibiting reading of limitations from the specification into the claims during examination. The same standard applies to issued patents under reexamination because, the statutory presumption of validity, 35 U.S.C. 282, has no application in reexamination (*In re Etter*, 756 F.2d 852, 225 USPQ 1 (Fed. Cir. 1985)).

12. With this, in response to the Patent Owner's arguments on page 10, which argue that Cohen fails to expressly teach of the feature that "enable[s] interactive processing of said object", whereby, as noted above, the Patent Owner defines "interactive processing" as being a processing of the user utilizing the mouse or keyboard or similar device, that changes the structure or presentation of the object, thus being an interactive process. However, the current claim language does not specify this. Further, there is no requirement that the "interactive processing" be a process performed by the "user".

13. Further, on pages 10 and 11 of the Patent Owner's arguments, the Patent Owner additionally provides sections in the specification of the '906 Patent that describe the "interactive processing". However, the examiner notes that it is improper to import features found in the specification into the claim language. In this regard, MPEP 2111.01 [R-5] states under the heading "II. IT IS IMPROPER TO IMPORT CLAIM LIMITATIONS FROM THE SPECIFICATION":

“Though understanding the claim language may be aided by explanations contained in the written description, **it is important not to import into a claim limitations that are not part of the claim.** For example, a particular embodiment appearing in the written description may not be read into a claim when the claim language is broader than the embodiment.” *Superguide Corp. v. DirecTV Enterprises, Inc.*, 358 F.3d 870, 875, 69 USPQ2d 1865, 1868 (Fed. Cir. 2004). [Emphasis added.]

14. Currently, claim 1 states “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 ...”. With this, the reference of Cohen states in column 9, lines 41-49, that “The profile 300 includes the hardware types for a particular I/O function, characteristics for each hardware type, and the software drivers, which **enable** the application programs and I/O handler programs to **interact** with the particular I/O hardware or software.” [Emphasis added]. Thus, Cohen recognizes that the application programs perform an interactive process with the particular I/O hardware or software, and are enabled by the profile 300. There is no limitation in the current claim language that particularly requires that the process of interactive processing includes the function that the user, by using a mouse or keyboard or similar input device, can change the structure or presentation of the object, as argued. If the Patent Owner wishes that this function be considered, the Patent Owner must add the particular language to the claim.

15. Further, as noted by the Patent Owner on page 11 of the arguments, the previous Reexamination proceeding stated in the Reasons for Patentability/Confirmation mailed 09/27/05 on page 4 that:

To be consistent with the specification, the claimed “interactive processing” necessarily requires some capability of ongoing real-time manipulation and control by the user of the object displayed within the browser-controlled window.

However, that Reasons for Patentability/Confirmation mailed 9/27/05 continues with:

In particular, the claimed “interactive processing,” when properly construed in a manner consistent with the specification, requires:

“Interprocess communication between the hypermedia browser and the embedded application program is ongoing after the program object has been launched” [see instant ‘906 patent, col. 7, lines 1-4].

16. As similarly noted above, in column 9, lines 39-49, Cohen sates that “FIG. 5 depicts **the user’s** workstation profile 300, which characterizes the input and output devices which are available to the workstation 200 in FIG. 4. The profile 300 includes the hardware types for a particular I/O function, characteristics for each hardware type, and the software drivers, which **enable** the application programs and I/O handler programs to **interact** with the particular I/O hardware or software.” [Emphasis added]. Thus, with this, Cohen is seen as teaching of interprocess communication that is “ongoing after the program object is launched”, whereby through the user’s profile 300, there is “**some capability** of ongoing real-time manipulation and control by the user” [Emphasis added].

Thus, as mentioned above, there is no specific limitation in the current claim language that requires the interactive processing be the process where the user, by using a mouse or keyboard or similar input device, can change the structure or presentation of the object. As described above, Cohen is seen as teaching, in a broad and reasonable manner, without reading limitations from the specification of the '906 Patent, the claimed "interactive processing" limitation, as currently worded in independent claims 1 and 6.

17. Continuing, in response to the Patent Owner's arguments regarding the "embed text format", whereby the Patent Owner argues on pages 15 and 16 that Cohen does not teach of the "embed text format", which, as claimed, is located at a first location in said first distributed hypermedia document, and which specifies the location of at least a portion of an object external to the first distributed hypermedia document. The Patent Owner continues, arguing that the LDESC tag of Cohen "is not located at a first location in the document where a display window is created", and the link tag :L of Cohen "lacks the claimed feature that the embed text format specifies the location of at least a portion of an object external to the first distributed hypermedia document".

18. First, the examiner notes that independent claims 1 and 6 do not particularly require tags "located at a first location in the document where a display window is created", as argued, but rather require the "embed text format" be "located at a first location in said first distributed hypermedia document", having "an 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". Thus,

the display area for the object that is within the displayed portion of said first distributed hypermedia document is “created at said first location”. With this, the current claim language does not necessarily require that the “said first location” is within the displayed portion of the hypermedia document, or that the “said first location” is within the created “display area”, as the claim only states that a display area for said object is “created at said first location”.

19. Therefore, with this interpretation, Cohen can be interpreted that the :LDESC tag seen in Fig. 1a can be considered as an “embed text format”. Particularly, Cohen states in col. 3, lines 23 and 24, referring to Fig. 1a, that a “display area” is created within “the link description tags 102 of the softcopy book file 100”, being at the :LDESC tag location, which is within the hypermedia document. Further, as seen in Figs. 4b and 4c, the display buffer 238 includes the chart 190’, titled “Worldwide Elephant Population”, and the text 174’. Continuing, in col. 9, lines 20-26, Cohen states that “The display buffer 238 stores the resulting picture displayed on the monitor display screen 208.” Thus, the created “display area” will be displayed within a portion of the hypermedia document that is displayed in the browser-controlled window, as seen in Figs. 4b and 4c.

20. However, the examiner also notes that the term “embed text format” is not particularly defined in the specification of the ‘906 Patent. Thus, the phrase “embed text format” will be given the “plain meaning” of the phrase. In the specification of the ‘906 Patent, in col. 12, line 54-col. 13, line 36, an HTML tag format is described “to embed a link to an application program within a hypermedia document”. Therefore, the plain

meaning of the phrase “embed text format”, being consistent with the specification, will thus be viewed as a format that can embed a link, being text, to an application program within a hypermedia document.

21. Continuing, with this, an alternate interpretation of the claim language is also seen. Particularly, claim 1 requires a format, being characterized as embedding text. Thus, the format of Cohen, has a multimedia hypertext link tag, which is text, is embedded in the document, as seen in Fig. 1b. Further the format includes a corresponding link description tag, which is seen in Fig. 1a. Therefore, as an example, in the link 164 seen in Fig. 1b, the embedded text contains “eleph_movie”. The format of Cohen further specifies in Fig. 1a that the tag 120 includes the embedded text “eleph_movie”, whereby the object “family_clip.vid” is stored “external” to the workstation. Thus, with this, Cohen is seen as teaching of an “embed text format” that is “located at a first location in said first distributed hypermedia document”, being seen in Fig. 1b, whereby the format “specifies the location of at least a portion of an object external to the first distributed hypermedia document”, as seen in Fig. 1a. Further, Cohen is seen in Figs. 4b and 4c, that the object, being in this case “population.gph”, seen in Fig. 1a at tag 150, is within a display area created at said first location, as seen in Fig. 1b at portion 172, within the portion of said first distributed hypermedia document being displayed in said first browser-controlled window, whereby Cohen states in col. 9, lines 20-26, that “The display buffer 238 stores the resulting picture displayed on the monitor display screen 208.” Thus, the “embed text format”, as currently worded in independent claims 1 and 6, can be interpreted as being taught by Cohen.

22. Continuing, in response to Patent Owner's arguments on page 17, which state that Cohen fails to expressly teach of the claimed element of "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". As noted in the previous Office action, the IBM BookManager READ program can be considered as a "browser", as the functionality is equivalent. Further, the displays produced when utilizing the BookManager READ program can be subsequently interpreted as "browser controlled windows". This is further seen in a newly introduced reference of "How People Use Softcopy Documentation: A Case Study", written by Dave Hendry et al., whereby as seen on page 87 in Figure 6, the display of a book is within a "browser controlled window". As seen in Figure 6, using the BookManager READ/DOS 1.2 program, the displayed page of a book's "Table of Contents" includes page controls on the bottom of the page, noted as "F1=Help F3=Exit F7=Bkwd F8=Fwd F10=Actions", and also includes various headings for other browser controls, noted as "Books GoTo Search Notes serVices Options Help". With this, the BookManager READ program, which is taught by Cohen, is seen as inherently displaying a page of a book with these various controls, which can thus be equated to a browser-controlled window.

23. Thus, the items that are to be displayed, as shown in Figs. 4b and 4c of Cohen, would inherently include page controls, as well as the drop-down headings across the top of the displayed page, as seen in the controls for the BookManager READ program. This display can be considered as a "browser-controlled window". Further, as seen in Figs. 4b

and 4c, Cohen is teaching that the object to be displayed (noted as the graph 190', being the "population.gph", as seen in Figs. 1a and 1b) would be displayed in the same window as the text 174'. Therefore, the reference of Cohen can be interpreted as teaching the claimed features of "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."

24. Continuing, in response to Patent Owner's arguments on page 19, whereby the Patent Owner argues that Cohen fails to explicitly teach of the claimed element of type information, associated with the object, utilized by the browser to identify and locate an executable application external to the first distributed hypermedia document. The term "type information" can be understood in a broad and reasonable interpretation as being information regarding the type of object. As seen in Fig. 1a, Cohen is seen as teaching of object data (being "family_clip.vid", "trumpet.aud", and/or "population.gph"). With this, as also seen in Fig. 1a, Cohen teaches of "type information" being the information that describes the type of object, such as "CD Video File Format A" or "GOCA Format C", as well as "STORE=external" or "OBJTYPE=video". Further, as read in col. 9, lines 39-53, "The profile 300 includes the hardware types for a particular I/O function, characteristics for each hardware type..." Thus, this "information" is utilized to identify and locate an executable application external to the first distributed hypermedia document

25. Further, as seen in Figs. 7a-7c, Cohen shows that the executable applications of “video.exe”, “audio.exe”, and “graph.exe”, as seen in steps 504, 534, and 564, respectively, are each started. The method continues with the data string being parsed by the BookManager READ program, which as discussed above, is equivalent to a browser, to find the required hardware and software support, as read in steps 506, 536, and 566, via the profile 300 seen in Fig. 5. Finally, as an example, in step 584 in Fig. 7c, the diagram states “If workstation profile includes required graphics support, then output data string to graphics support and access object at store location and output object to graphics support”, with the corresponding text in col. 13, line 52-col. 14, line 10, wherein “In this case, the required support is the GOCA Format C characteristic, which the workstation profile indicates is present in the workstation 200.” Thus, Cohen is seen as teaching of type information, associated with the object, that is utilized by the browser to identify and locate an executable application external to the first distributed hypermedia document.

Information Disclosure Statements

26. First, it is noted that the Court documents submitted on 10/15/07 have been considered by the examiner. However, the citations listed in the Information Disclosure Statement dated 10/15/07 within the “Other Prior Art – Non Patent Literature Documents” section of the various Court papers and documents have been indicated as having a line through their citations. The indicated Court documents are not considered as “Prior Art” documents, since the Court documents are each dated June-August 2007, which is not prior to the filing date of the ‘906 Patent of Oct. 17, 1994. Thus, the citations of these Court documents should not be listed in the Information Disclosure Statements.

27. Continuing, the examiner notes that 37 CFR 1.98, having the heading “Content of information disclosure statement”, under Section (b)5 states:

Each publication listed in an information disclosure statement must be identified by publisher, author (if any), title, relevant pages of the publication, date, and place of publication.

With this, the Information Disclosure Statements filed on 10/31/07 and 1/8/08, both include listings in the “Other Prior Art - Non Patent Literature Documents” section that do not conform with 37 CFR 1.98 (b)5. As an example, the first listing noted in the IDS dated 10/31/07 is cited as “DTX1031-c.pdf (MS_SUPP1205 002 part 1)” and the first listing in the IDS dated 1/8/08 is cited as “DX273-text.pdf”. These listings do not identify the “publisher, author (if any), title, relevant pages of the publication, date, and place of publication”.

28. 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 above noted citations in the Information Disclosure Statements dated 10/31/07 and 1/8/08 do not comply with the content requirement of 37 CFR 1.98. Therefore, the information has not been considered by the examiner and has been indicated so by the line through the citations.

Claim Rejections - 35 USC § 102

29. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

30. **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”).

31. As noted in the previous Office action dated 7/30/07, 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.

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.”],

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 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 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 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 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 = ‘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 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 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 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].

**STATEMENT OF REASONS FOR PATENTABILITY AND/OR
CONFIRMATION**

32. The following is an examiner's statement of reasons for patentability and/or confirmation of the claims found patentable in this reexamination proceeding:

Claims 4, 5, 9, and 10 are confirmed as patentable.

33. With respect to dependent *claims 4 and 9*, the examiner believes that it would not have been obvious to one of ordinary skill in the art at the time of the invention to have the method and computer program product, as claimed, further include the features of issuing one or more commands to the network server from the client workstation, executing the one or more instructions on the network server, and sending the information from the network server to the client workstation in response to the executed instructions, and processing the information at the client workstation to interactively control the application.

34. As discussed above, the prior art of Cohen can be interpreted as teaching of a system that includes an embed text format that specifies a location of at least a portion of the object external to a hypermedia document, which is further utilized to identify and locate an executable application that is external to the hypermedia document. However, Cohen does not explicitly teach if the external application is located at a server, whereby the instructions would be executed at the server, with the client workstation and server performing the process defined in claims 4 and 9, respectively. Further, the examiner can

find no other teaching that would motivate one of ordinary skill in the art to modify the Cohen teachings so perform these features.

Conclusion

35. **THIS ACTION IS MADE FINAL.**

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, and it must be accompanied by the petition fee set forth in 37 CFR 1.17(g). 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.

The filing of a timely first response to this final rejection will be construed as including a request to extend the shortened statutory period for an additional month, which will be granted even if previous extensions have been granted. In no event

however, will the statutory period for response expire later than SIX MONTHS from the mailing date of the final action. See MPEP § 2265.

36. 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.

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

Application/Control Number:
90/007,858
Art Unit: 3992

Page 33

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

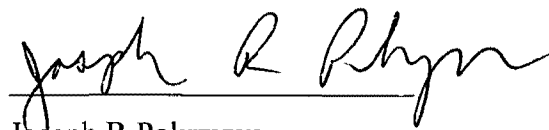
By Mail to: Mail Stop *Ex Parte* Reexam
Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

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:

R6F 

Notice of References Cited	Application/Control No. 90/007,858	Applicant(s)/Patent Under Reexamination 5838906	
	Examiner JOSEPH R. POKRZYWA	Art Unit 3992	Page 1 of 1

U.S. PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A US-			
	B US-			
	C US-			
	D US-			
	E US-			
	F US-			
	G US-			
	H US-			
	I US-			
	J US-			
	K US-			
	L US-			
	M US-			

FOREIGN PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N				
	O				
	P				
	Q				
	R				
	S				
	T				

NON-PATENT DOCUMENTS

*	Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)			
U	"How People Use Softcopy Documentation: A Case Study", Dave Hendry et al., Conference of the Centre for Advanced Studies on Collaborative Research (CASCON) 1991, pp. 77-93			
V				
W				
X				

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

858 PH Ex. 9



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 06/03/2008

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

EXAMINER

ART UNIT PAPER NUMBER

DATE MAILED: 06/03/2008

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 Interview Summary	Control No.	Patent Under Reexamination	
	90/007,858	5838906	
	Examiner	Art Unit	
	JOSEPH R. POKRZYWA	3992	

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

- (1) Joseph R. Pokrzywa (3) Charles Krueger
(2) consulted Sue Lao & Roland Foster (4) David Cochran

Date of Interview: 6/3/08 & 5/9/08

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: Facsimile of Patent Owner's proposed amendment

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.

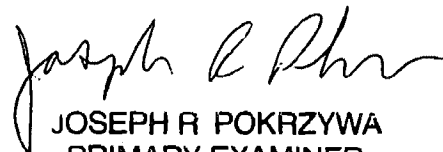
Identification of prior art discussed: Cohen (U.S. Patent Number 5,367,621).

Description of the general nature of what was agreed to if an agreement was reached, or any other comments:

Discussed the proposed amendment presented by the Patent Owner. The proposed amendment appears to overcome the reference of Cohen, as upon initial review, Cohen is unclear if an end-user directly interacts with the object, rather than the "interactive processing" as claimed. The examiner will reconsider the rejection fully upon filing of a formal amendment.

(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).**


JOSEPH R POKRZYWA
PRIMARY EXAMINER
CRU - AU 3992

cc: Requester (if third party requester)

Examiner's signature, if required

858 PH Ex. 10

06/23/08

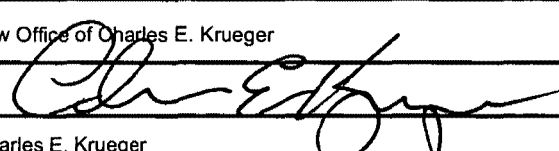
Approved for use through 07/31/2006. OMB 0651-0031
 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

67274 U.S. PTO

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 <i>(to be used for all correspondence after initial filing)</i>	Application Number	REX Control No. 90/007,858	
	Filing Date	12/22/2005	
	First Named Inventor	Patent No. 5,838,906	
	Art Unit	3992	
	Examiner Name	Pokrzywa, Joseph R.	
Total Number of Pages in This Submission	17	Attorney Docket Number	006-1-5

ENCLOSURES (Check all that apply)		
<input type="checkbox"/> Fee Transmittal Form	<input type="checkbox"/> Drawing(s)	<input type="checkbox"/> After Allowance Communication to TC
<input type="checkbox"/> Fee Attached	<input type="checkbox"/> Licensing-related Papers	<input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences
<input checked="" type="checkbox"/> Amendment/Reply	<input type="checkbox"/> Petition	<input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
<input checked="" type="checkbox"/> After Final	<input type="checkbox"/> Petition to Convert to a Provisional Application	<input type="checkbox"/> Proprietary Information
<input type="checkbox"/> Affidavits/declaration(s)	<input type="checkbox"/> Power of Attorney, Revocation	<input type="checkbox"/> Status Letter
<input type="checkbox"/> Extension of Time Request	<input type="checkbox"/> Change of Correspondence Address	<input checked="" type="checkbox"/> Other Enclosure(s) (please identify below):
<input type="checkbox"/> Express Abandonment Request	<input type="checkbox"/> Terminal Disclaimer	(1) Fee Determination SB/06.
<input type="checkbox"/> Information Disclosure Statement	<input type="checkbox"/> Request for Refund	(2) Credit Card Payment Form 2038
<input type="checkbox"/> Certified Copy of Priority Document(s)	<input type="checkbox"/> CD, Number of CD(s) _____	(3) Patent Owner's Statement...Interview
<input type="checkbox"/> Reply to Missing Parts/Incomplete Application	<input type="checkbox"/> Landscape Table on CD	(4) Return Postcard
<input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53	Remarks	

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT			
Firm Name	Law Office of Charles E. Krueger		
Signature			
Printed name	Charles E. Krueger		
Date	June 17, 2008	Reg. No.	30,077

CERTIFICATE OF TRANSMISSION/MAILING			
I hereby certify that this correspondence is being facsimile transmitted to the USPTO or 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 the date shown below:			
Signature	06/25/2008 JWHITFIE 00000001 90007858		
	01 FC:1821 638.00 00		
Typed or printed name	Sharon D. Krueger	Date	June 17, 2008

This collection of information is required by 37 CFR 1.5. 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 1.11 and 1.14. This collection is estimated to 2 hours 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.

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.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number REX 90/007,858			
APPLICATION AS FILED – PART I									
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE (\$)	FEE (\$)	RATE (\$)	FEE (\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A		N/A			
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A		N/A		N/A			
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A		N/A			
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =		*	X	=	X	=		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =		*	X	=	X	=		
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$260 (\$130 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).								
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))									
* If the difference in column 1 is less than zero, enter "0" in column 2.									
APPLICATION AS AMENDED – PART II									
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	RATE (\$)	ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	*	14	Minus	**	20	=	0	
	Independent (37 CFR 1.16(h))	*	6	Minus	***	3	=	3	
	Application Size Fee (37 CFR 1.16(s))								
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
					TOTAL ADD'L FEE				
							TOTAL ADD'L FEE	630	
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	MINUS	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	RATE (\$)	ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	*		Minus	**		=		
	Independent (37 CFR 1.16(h))	*		Minus	***		=		
	Application Size Fee (37 CFR 1.16(s))								
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
					TOTAL ADD'L FEE				
							TOTAL ADD'L FEE		
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.									
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".									
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".									
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.									

This collection of information is required by 37 CFR 1.16. 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 12 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.

06/23/08

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

67274 U.S. PTO

In re reexamination control number:
90/007,858
Filed: 12/22/2005
Patent No: 5,838,906

Examiner: POKRZYWA, JOSEPH R
Art Unit: 3992

PATENT OWNER'S STATEMENT OF
THE SUBSTANCE OF THE INTERVIEW

Commissioner for Patents
Alexandria, VA 22313-1450

DATE: 5/9/2008 and 6/3/2008

TYPE: Telephonic

ATTENDEES: Examiner Pokrzywa, Attorney of Record Charles E. Krueger and associated attorney David Cochran.

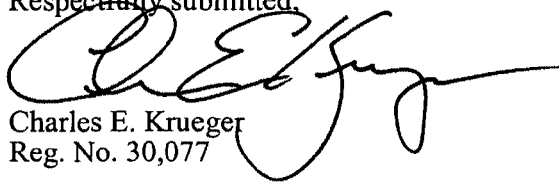
EXHIBITS AND DEMONSTRATIONS: None.

CLAIMS DISCUSSED: Independent claims 1 and 6.

REFERENCES DISCUSSED: Cohen.

SUMMARY: An amendment to claims 1 and 6 was proposed to overcome the outstanding rejection. The examiner indicated that upon initial review the amendment appears to overcome the rejection based on the Cohen reference.

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

06/25/2008 JWHITFIE 00000001 90007858
01 FC:1021

630.00 0P

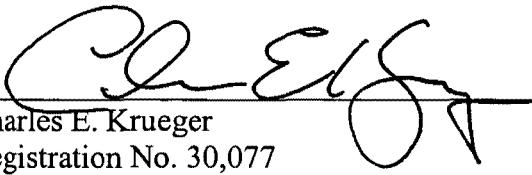
REX Control No. 90/007,858
Attorney Docket No. 006-1-5

CERTIFICATE OF SERVICE

I hereby certify that on this date I caused a true and correct copy of the attached Patent Owner's Statement of the Substance of the Interview to be served via first class mail on the following:

Stephen A. Wight
Klarquist Sparkman, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon St.
Portland, OR 97204

6/17/2004
Date


Charles E. Krueger
Registration No. 30,077

Law Office of Charles E. Krueger
P.O. Box 5607
Walnut Creek, CA 94549
(925) 944-3320

67274 U.S. PTO

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Reexamination Control No:

90/007,858

Filing Date: 12/22/2005

Patent No: 5,838,906

Examiner: POKRZYWA, JOSEPH R.

Art Unit: 3992

AMENDMENT AFTER FINAL

Commissioner for Patents
Alexandria, VA 22313-1450

5 Sir:

In response to the Office Action mailed 04/18/2008, please amend the application as follows:

Amendments to the Claims begin on page 2 of this paper.

10

Remarks/Conclusion begins on page 10 of this paper.

AMENDMENT TO THE CLAIMS:

Please amend the following claims.

1 1. (Amended) A method for running an application program in a computer
2 network environment, comprising:

3 providing at least one client workstation and one network server coupled to
4 said network environment, wherein said network environment is a distributed hypermedia
5 environment;

6 executing, at said client workstation, a browser application, that parses a first
7 distributed hypermedia document to identify text formats included in said distributed
8 hypermedia document and for responding to predetermined text formats to initiate processing
9 specified by said text formats; utilizing said browser to display, on said client workstation, at
10 least a portion of a first hypermedia document received over said network from said server,
11 wherein the portion of said first hypermedia document is displayed within a first browser-
12 controlled window on said client workstation, wherein said first distributed hypermedia
13 document includes an embed text format, located at a first location in said first distributed
14 hypermedia document, that specifies the location of at least a portion of an object external to
15 the first distributed hypermedia document, wherein said object has type information
16 associated with it utilized by said browser to identify and locate an executable application
17 external to the first distributed hypermedia document, and wherein said embed text format is
18 parsed by said browser to automatically invoke said executable application to execute on said
19 client workstation in order to display said object and enable an end-user to directly interact
20 with [interactive processing of] said object within a display area created at said first location
21 within the portion of said first distributed hypermedia document being displayed in said first
22 browser-controlled window.

1 4. (Amended) [The method of claim 3,] A method for running an
2 application program in a computer network environment, comprising:

3 providing at least one client workstation and one network server coupled to
4 said network environment, wherein said network environment is a distributed hypermedia
5 environment;

6 executing, at said client workstation, a browser application, that parses a first
7 distributed hypermedia document to identify text formats included in said distributed
8 hypermedia document and for responding to predetermined text formats to initiate processing

9 specified by said text formats; utilizing said browser to display, on said client workstation, at
10 least a portion of a first hypermedia document received over said network from said server,
11 wherein the portion of said first hypermedia document is displayed within a first browser-
12 controlled window on said client workstation, wherein said first distributed hypermedia
13 document includes an embed text format, located at a first location in said first distributed
14 hypermedia document, that specifies the location of at least a portion of an object external to
15 the first distributed hypermedia document, wherein said object has type information
16 associated with it utilized by said browser to identify and locate an executable application
17 external to the first distributed hypermedia document, and wherein said embed text format is
18 parsed by said browser to automatically invoke said executable application to execute on said
19 client workstation in order to display said object and enable interactive processing of said
20 object within a display area created at said first location within the portion of said first
21 distributed hypermedia document being displayed in said first browser-controlled window;
22 wherein said executable application is a controllable application and further
23 comprising the step of:
24 interactively controlling said controllable application on said client
25 workstation via inter-process communications between said browser and said controllable
26 application;
27 wherein the communications to interactively control said controllable
28 application continue to be exchanged between the controllable application and the browser
29 even after the controllable application program has been launched; and
30 wherein additional instructions for controlling said controllable application
31 reside on said network server, wherein said step of interactively controlling said controllable
32 application includes the following substeps:
33 issuing, from the client workstation, one or more commands to the network
34 server;
35 executing, on the network server, one or more instructions in response to said
36 commands;
37 sending information from said network server to said client workstation in
38 response to said executed instructions; and processing said information at the client
39 workstation to interactively control said controllable application.
40

1 5. (Amended) [The method of claim 4,] A method for running an application
2 program in a computer network environment, comprising:
3 providing at least one client workstation and one network server coupled to
4 said network environment, wherein said network environment is a distributed hypermedia
5 environment;
6 executing, at said client workstation, a browser application, that parses a first
7 distributed hypermedia document to identify text formats included in said distributed
8 hypermedia document and for responding to predetermined text formats to initiate processing
9 specified by said text formats; utilizing said browser to display, on said client workstation, at
10 least a portion of a first hypermedia document received over said network from said server,
11 wherein the portion of said first hypermedia document is displayed within a first browser-
12 controlled window on said client workstation, wherein said first distributed hypermedia
13 document includes an embed text format, located at a first location in said first distributed
14 hypermedia document, that specifies the location of at least a portion of an object external to
15 the first distributed hypermedia document, wherein said object has type information
16 associated with it utilized by said browser to identify and locate an executable application
17 external to the first distributed hypermedia document, and wherein said embed text format is
18 parsed by said browser to automatically invoke said executable application to execute on said
19 client workstation in order to display said object and enable interactive processing of said
20 object within a display area created at said first location within the portion of said first
21 distributed hypermedia document being displayed in said first browser-controlled window;
22 wherein said executable application is a controllable application and further
23 comprising the step of:
24 interactively controlling said controllable application on said client
25 workstation via inter-process communications between said browser and said controllable
26 application;
27 wherein the communications to interactively control said controllable
28 application continue to be exchanged between the controllable application and the browser
29 even after the controllable application program has been launched;
30 wherein additional instructions for controlling said controllable application
31 reside on said network server, wherein said step of interactively controlling said controllable
32 application includes the following substeps:

33 issuing, from the client workstation, one or more commands to the network
34 server;
35 executing, on the network server, one or more instructions in response to said
36 commands;
37 sending information from said network server to said client workstation in
38 response to said executed instructions; and processing said information at the client
39 workstation to interactively control said controllable application; and
40 wherein said additional instructions for controlling said controllable
41 application reside on said client workstation.

1 6. (Amended) A computer program product for use in a system having at
2 least one client workstation and one network server coupled to said network environment,
3 wherein said network environment is a distributed hypermedia environment, the computer
4 program product comprising:
5 a computer usable medium having computer readable program code
6 physically embodied therein, said computer program product further comprising:
7 computer readable program code for causing said client workstation to execute
8 a browser application to parse a first distributed hypermedia document to identify text
9 formats included in said distributed hypermedia document and to respond to predetermined
10 text formats to initiate processes specified by said text formats;
11 computer readable program code for causing said client workstation to
12 utilize said browser to display, on said client workstation, at least a portion of a first
13 hypermedia document received over said network from said server, wherein the portion of
14 said first hypermedia document is displayed within a first browser-controlled window on said
15 client workstation, wherein said first distributed hypermedia document includes an embed
16 text format, located at a first location in said first distributed hypermedia document, that
17 specifies the location of at least a portion of an object external to the first distributed
18 hypermedia document, wherein said object has type information associated with it utilized by
19 said browser to identify and locate an executable application external to the first distributed
20 hypermedia document, and wherein said embed text format is parsed by said browser to
21 automatically invoke said executable application to execute on said client workstation in
22 order to display said object and enable an end-user to directly interact with [interactive
23 processing of] said object within a display area created at said first location within the portion

24 of said first distributed hypermedia document being displayed in said first browser-controlled
25 window.

1 9. (Amended) [The computer program product of claim 8,] A computer
2 program product for use in a system having at least one client workstation and one network
3 server coupled to said network environment, wherein said network environment is a
4 distributed hypermedia environment, the computer program product comprising:
5 a computer usable medium having computer readable program code physically
6 embodied therein, said computer program product further comprising:
7 computer readable program code for causing said client workstation to execute
8 a browser application to parse a first distributed hypermedia document to identify text
9 formats included in said distributed hypermedia document and to respond to predetermined
10 text formats to initiate processes specified by said text formats;
11 computer readable program code for causing said client workstation to utilize
12 said browser to display, on said client workstation, at least a portion of a first hypermedia
13 document received over said network from said server, wherein the portion of said first
14 hypermedia document is displayed within a first browser-controlled window on said client
15 workstation, wherein said first distributed hypermedia document includes an embed text
16 format, located at a first location in said first distributed hypermedia document, that specifies
17 the location of at least a portion of an object external to the first distributed hypermedia
18 document, wherein said object has type information associated with it utilized by said
19 browser to identify and locate an executable application external to the first distributed
20 hypermedia document, and wherein said embed text format is parsed by said browser to
21 automatically invoke said executable application to execute on said client workstation in
22 order to display said object and enable interactive processing of said object within a display
23 area created at said first location within the portion of said first distributed hypermedia
24 document being displayed in said first browser-controlled window;
25 wherein said executable application is a controllable application and further
26 comprising:
27 computer readable program code for causing said client workstation to
28 interactively control said controllable application on said client workstation via inter-process
29 communications between said browser and said controllable application;

30 wherein the communications to interactively control said controllable
31 application continue to be exchanged between the controllable application and the browser
32 even after the controllable application program has been launched; and
33 wherein additional instructions for controlling said controllable application
34 reside on said network server, wherein said computer readable program code for causing said
35 client workstation to interactively control said controllable application on said client
36 workstation [step of interactively controlling said controllable application] includes:
37 computer readable program code for causing said client workstation to issue,
38 from the client workstation, one or more commands to the network server;
39 computer readable program code for causing said network server to execute
40 one or more instructions in response to said commands;
41 computer readable program code for causing said network server [sever] to
42 send information to said client workstation in response to said executed instructions; and
43 computer readable program code for causing said client workstation to process
44 said information at the client workstation to interactively control said controllable
45 application.

1 10. (Amended) [The computer program product of claim 9,] A computer
2 program product for use in a system having at least one client workstation and one network
3 server coupled to said network environment, wherein said network environment is a
4 distributed hypermedia environment, the computer program product comprising:
5 a computer usable medium having computer readable program code physically
6 embodied therein, said computer program product further comprising:
7 computer readable program code for causing said client workstation to execute
8 a browser application to parse a first distributed hypermedia document to identify text
9 formats included in said distributed hypermedia document and to respond to predetermined
10 text formats to initiate processes specified by said text formats;
11 computer readable program code for causing said client workstation to utilize
12 said browser to display, on said client workstation, at least a portion of a first hypermedia
13 document received over said network from said server, wherein the portion of said first
14 hypermedia document is displayed within a first browser-controlled window on said client
15 workstation, wherein said first distributed hypermedia document includes an embed text
16 format, located at a first location in said first distributed hypermedia document, that specifies

17 the location of at least a portion of an object external to the first distributed hypermedia
18 document, wherein said object has type information associated with it utilized by said
19 browser to identify and locate an executable application external to the first distributed
20 hypermedia document, and wherein said embed text format is parsed by said browser to
21 automatically invoke said executable application to execute on said client workstation in
22 order to display said object and enable interactive processing of said object within a display
23 area created at said first location within the portion of said first distributed hypermedia
24 document being displayed in said first browser-controlled window;

25 wherein said executable application is a controllable application and further
26 comprising:

27 computer readable program code for causing said client workstation to
28 interactively control said controllable application on said client workstation via inter-process
29 communications between said browser and said controllable application;

30 wherein the communications to interactively control said controllable
31 application continue to be exchanged between the controllable application and the browser
32 even after the controllable application program has been launched;

33 wherein additional instructions for controlling said controllable application
34 reside on said network server, wherein said computer readable program code for causing said
35 client workstation to interactively control said controllable application on said client
36 workstation includes:

37 computer readable program code for causing said client workstation to issue,
38 from the client workstation, one or more commands to the network server;

39 computer readable program code for causing said network server to execute
40 one or more instructions in response to said commands;

41 computer readable program code for causing said network server to send
42 information to said client workstation in response to said executed instructions; and

43 computer readable program code for causing said client workstation to process
44 said information at the client workstation to interactively control said controllable
45 application; and

46 wherein said additional instructions for controlling said controllable
47 application reside on said client workstation.

1 11. (New) The method of claim 3, wherein additional instructions for
2 controlling said controllable application reside on said network server, wherein said step of
3 interactively controlling said controllable application includes the following substeps:
4 issuing, from the client workstation, one or more commands to the network
5 server;
6 executing, on the network server, one or more instructions in response to said
7 commands;
8 sending information from said network server to said client workstation in
9 response to said executed instructions; and processing said information at the client
10 workstation to interactively control said controllable application.

1 12. (New) The method of claim 11, wherein said additional instructions for
2 controlling said controllable application reside on said client workstation.

1 13. (New) The computer program product of claim 8, wherein additional
2 instructions for controlling said controllable application reside on said network server,
3 wherein said computer readable program code for causing said client workstation to
4 interactively control said controllable application on said client workstation includes:
5 computer readable program code for causing said client workstation to issue,
6 from the client workstation, one or more commands to the network server;
7 computer readable program code for causing said network server to execute
8 one or more instructions in response to said commands;
9 computer readable program code for causing said network server to send
10 information to said client workstation in response to said executed instructions; and
11 computer readable program code for causing said client workstation to process
12 said information at the client workstation to interactively control said controllable
13 application.

1 14. (New) The computer program product of claim 13, wherein said
2 additional instructions for controlling said controllable application reside on said client
3 workstation.

REMARKS

Claims 1-10 have been reexamined, claims 1, 4, 5, 6, 9 and 10 are amended and claims 11-14 are added herewith. Accordingly, claims 1-14 are now pending in this reexamination. Patent Owner hereby requests reconsideration of all outstanding rejections and objections and notes with appreciation that claims 4, 5, 9 and 10 were confirmed as patentable.

Confirmed claims 4, 5, 9, and 10 have been rewritten in independent form including all the limitations of the base and intervening claims.

Further, claim 9 has been amended to replace "said step of interactively controlling said controllable application" with -- computer readable program code for causing said client workstation to interactively control said controllable application on said client workstation-- to correct a typographical error that occurred when the method claim 4 was being converted to the computer program product claim 9. Also, "sever" has been changed to --server- to correct a typographical error.

Claims 11 and 12 are the original claims 4 and 5 which depend on amended claim 1, and are thus allowable for the same reasons as claim 1. Claims 13 and 14 are the original claims 9 and 10 which depend on amended claim 6 and are thus allowable for the same reasons as claim 6. Claim 13 has been changed to correct the typographical errors noted above regarding claim 9.

Claims 1-3 and 6-8 stand finally 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. This rejection is traversed.

The same rejection under 35 U.S.C. §102(e) was set forth in the non-final office action mailed 7/30/2007 and a response to the non-final rejection was filed by Patent Owner that presented a detailed traverse to all the grounds of rejection set forth in the non-final office action. In particular, the response set forth a detailed analysis respectfully traversing the examiner's assertions that the Cohen reference fairly teaches or suggests the interactive processing, embed text format, the display area, and the type information recited in claims 1 and 6.

In the final office action, in particular with regard to interactive processing, the examiner did not assert that Cohen teaches interactive processing as argued in the traverse. Instead, the examiner asserted that the language of claims 1 and 6 can be read more broadly than

argued in the traverse by applying the examiner's view of the broadest reasonable interpretation of the claims.

In paragraph 7 of the office action, the examiner notes that the Patent Owner's argument "states that the claim construction set forth in the Markman ruling in the related litigation, which was affirmed by the U.S. Court of Appeals for the Federal Circuit, is utilized in the [Patent Owner's] subsequent remarks." The examiner further notes "that this is not the proper standard for claim construction during examination before the Office... ." In paragraph 8 of the office action, this standard is stated to be "claims must be given their broadest reasonable interpretation consistent with the specification."

The Patent Owner continues to respectfully assert the position argued in its response to the non-final office action that Cohen does not fairly teach or suggest many of the features of the claims, including at least interactive processing, an embed text format, a display area and type information as those features are defined by the unamended language of claims 1 and 6, and that this position is correct in view of the cases cited by the examiner. Claims 1 and 6 have now been amended to include the language that said executable application enables "an end-user to directly interact with said object" in order to obviate the grounds of rebuttal relating to interactive processing asserted by the examiner under his view of the broadest reasonable interpretation of claims 1 and 6. This language is consistent with the claim construction set forth in the Markman ruling as affirmed by the Court of Appeals for the Federal Circuit and therefore this language does not change the scope of claims 1 and 6.

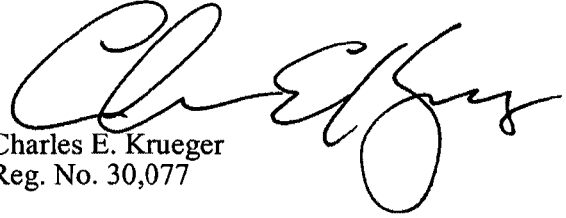
CONCLUSION

In view of the foregoing, all claims now pending in this Reexamination are in condition for confirmation. The mailing of a formal Notice of Intent to Issue Reexamination Certificate (NIRC) at an early date is respectfully requested.

Doyle et al.
Reexamination Control No. 90/007,858
Page 12

If the Examiner believes a telephone conference would expedite prosecution,
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
EMAIL: ckrueger@sbcglobal.net

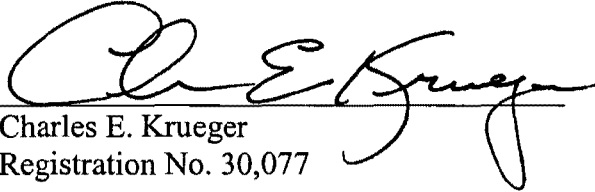
REX Control No. 90/007,858
Attorney Docket No. 006-1-5

CERTIFICATE OF SERVICE

I hereby certify that on this date I caused a true and correct copy of the attached Amendment After Final to be served via first class mail on the following:

Stephen A. Wight
Klarquist Sparkman, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon St.
Portland, OR 97204

6/17/2008
Date


Charles E. Krueger
Registration No. 30,077

Law Office of Charles E. Krueger
P.O. Box 5607
Walnut Creek, CA 94549
(925) 944-3320

858 PH Ex. 11



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 09/10/2008

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

EXAMINER

ART UNIT PAPER NUMBER

DATE MAILED: 09/10/2008

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)).

Notice of Intent to Issue Ex Parte Reexamination Certificate	Control No.	Patent Under Reexamination	
	90/007,858	5838906	
	Examiner	Art Unit	
	JOSEPH R. POKRZYWA	3992	

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

1. Prosecution on the merits is (or remains) closed in this *ex parte* reexamination proceeding. This proceeding is subject to reopening at the initiative of the Office or upon petition. Cf. 37 CFR 1.313(a). A Certificate will be issued in view of
 - (a) Patent owner's communication(s) filed: 23 June 2008.
 - (b) Patent owner's late response filed: _____.
 - (c) Patent owner's failure to file an appropriate response to the Office action mailed: _____.
 - (d) Patent owner's failure to timely file an Appeal Brief (37 CFR 41.31).
 - (e) Other: _____.

Status of *Ex Parte* Reexamination:

 - (f) Change in the Specification: Yes No
 - (g) Change in the Drawing(s): Yes No
 - (h) Status of the Claim(s):
 - (1) Patent claim(s) confirmed: _____.
 - (2) Patent claim(s) amended (including dependent on amended claim(s)): 1-10
 - (3) Patent claim(s) cancelled: _____.
 - (4) Newly presented claim(s) patentable: 11-14.
 - (5) Newly presented cancelled claims: _____.
2. Note the attached statement of reasons for patentability and/or confirmation. Any comments considered necessary by patent owner regarding reasons for patentability and/or confirmation must be submitted promptly to avoid processing delays. Such submission(s) should be labeled: "Comments On Statement of Reasons for Patentability and/or Confirmation."
3. Note attached NOTICE OF REFERENCES CITED (PTO-892).
4. Note attached LIST OF REFERENCES CITED (PTO/SB/08).
5. The drawing correction request filed on _____ is: approved disapproved.
6. 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
 - been received.
 - not been received.
 - been filed in Application No. _____.
 - been filed in reexamination Control No. _____.
 - been received by the International Bureau in PCT Application No. _____.

* Certified copies not received: _____.
7. Note attached Examiner's Amendment.
8. Note attached Interview Summary (PTO-474).
9. Other: _____.

cc: Requester (if third party requester)
U.S. Patent and Trademark Office
PTOL-469 (Rev.08-06)

Notice of Intent to Issue Ex Parte Reexamination Certificate

Part of Paper No 20080808

DETAILED ACTION

Response to Amendment

1. Patent Owner's amendment was received on 6/23/08, and has been entered and made of record. The examiner notes that claims 1-10 originally issued in U.S. Patent Number 5,838,906 (hereafter "the '906 Patent"). With the current amendment dated 6/23/08, claims 1, 4, 5, 6, 9, and 10 were amended and claims 11-14 were newly added. Thus, currently, claims 1-14 are pending, and are the subject of the current reexamination proceeding.

Brief Summary of the Instant Proceedings

2. Within the current reexamination proceeding, an Office action dated 7/30/07 rejected claims 1-10 with the references of "A Brief Overview of the VIOLA Engine, and it's applications", written by Pei Wei, noted as "Viola", and rejected claims 1-3 and 6-8 with the reference of Cohen *et al.* (U.S. Patent Number 5,367,621, noted as "Cohen"), when viewed with "Introducing NCSA Mosaic" (noted as "NCSA Mosaic").

Art Unit: 3992

3. Subsequently, the Patent Owner submitted a Declaration under 37 CFR 1.131 on 10/1/07, which establishes the invention prior to August 16, 1994, being the date utilized as the publication date of the previously cited Viola reference. With that, in the Office action dated 4/18/08, the examiner withdrew the rejection of claims 1-10 as being anticipated by the Viola reference, but maintained the rejection of claims 1-3 and 6-8 as being unpatentable over Cohen in view of NCSA Mosaic. Finally, the Patent Owner submitted the current amendment dated 6/23/08, which amends claims 1 and 6, and places the noted patentable claims 4, 5, 9, and 10 in independent form.

STATEMENT OF REASONS FOR PATENTABILITY AND/OR CONFIRMATION

The following is an examiner's statement of reasons for patentability and/or confirmation of the claims found patentable in this reexamination proceeding:

Claims 1-14 are deemed as patentable, as amended.

With the amendment dated 6/23/08, **claims 1, 4, 5, 6, 9, and 10** are independent.

With respect to independent **claims 1 and 6**, in the examiner's opinion, based on the prior art of record, it would not have been obvious to have the system, as claimed, include the features of an embed text format being parsed by the browser to automatically invoke the executable

Art Unit: 3992

application to execute on the client workstation in order to display the object and enable an end-user to directly interact with the object within a display area created at the first location within the portion of the hypermedia document within the browser controlled window. The examiner notes that the closest prior art, Cohen (U.S. Patent Number 5,367,621), utilizes the IBM BookManager system, whereby Cohen teaches of the AUTOLAUNCH function, which automatically launches an object, whereby the system can automatically invoke multimedia objects, such as “photographic quality graphics, motion video, or sound”, as read in col. 2, lines 50-66.

However, Cohen does not specifically disclose the feature of allowing an end-user to directly interact with the object within the display area of the browser window after the object is automatically invoked. Cohen shows that the graphic 190', as seen in Fig. 4b is automatically invoked. However, there is no indication that an end-user can directly interact with this graphic. Further, the specification of Cohen discusses inserting an audio object “eleph_sound.Audio 1 – Elephant's trumpet” and a movie object “eleph_movie.Motion Picture of African Elephant family”, as seen in Fig. 1b. But these examples are not automatically invoked using the AUTOLAUNCH function, and if they would be set to AUTOLAUNCH, there is no indication in Cohen that would provide the function allowing the end-user to directly interact with the automatically invoked object.

As noted in the specification of the '906 Patent in col. 7, lines 12-15 “Also, the user is able to rotate, scale and otherwise reposition the viewpoint with respect to these images without

Art Unit: 3992

exiting the hypermedia browser software.” There is no indication in Cohen that the BookManager READ program allows the end-user to perform this direct interaction of the object once the multimedia is launched automatically. Further, the examiner can find no other teaching in the prior art of record that would motivate one of ordinary skill in the art to modify the Cohen teachings so as to allow the end-user to directly interact with the automatically invoked object. Therefore, because of this feature that was added in the amendment dated 6/23/08, the invention defined in claims 1 and 6 is rendered as patentable.

With respect to independent *claims 4, 5, 9, and 10*, the examiner believes that it would not have been obvious to one of ordinary skill in the art at the time of the invention to have the method and computer program product, as claimed, further include the features of issuing one or more commands to the network server from the client workstation, executing the one or more instructions on the network server, and sending the information from the network server to the client workstation in response to the executed instructions, and processing the information at the client workstation to interactively control the application.

As discussed above, the prior art of Cohen can be interpreted as teaching of a system that includes an embed text format that specifies a location of at least a portion of the object external to a hypermedia document, which is further utilized to identify and locate an executable application that is external to the hypermedia document. However, Cohen does not explicitly teach if the external application is located at a server, whereby the instructions would be executed at the server, with the client workstation and server performing the process defined in

Art Unit: 3992

claims 4 and 9, respectively. Further, the examiner can find no other teaching that would motivate one of ordinary skill in the art to modify the Cohen teachings so perform these features. Therefore, because of these features, the invention defined in now independent claims 4, 5, 9, and 10 is rendered as patentable.

Any comments considered necessary by PATENT OWNER regarding the above statement must be submitted promptly to avoid processing delays. Such submission by the patent owner should be labeled: "Comments on Statement of Reasons for Patentability and/or Confirmation" and will be placed in the reexamination file.

Art Unit: 3992

Conclusion

4. ALL correspondence relating to this ex parte reexamination proceeding should be directed as follows:

Please mail any communications to:

Attn: Mail Stop "Ex Parte Reexam"
Central Reexamination Unit
Commissioner for Patents
P. O. Box 1450
Alexandria VA 22313-1450

Please FAX any communications to:

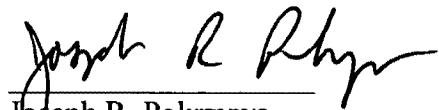
(571) 273-9900
Central Reexamination Unit

Please hand-deliver any communications to:

Customer Service Window
Attn: Central Reexamination Unit
Randolph Building, Lobby Level
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the Reexamination Legal Advisor or Examiner, or as to the status of this proceeding, 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 :

ESK
RGF

Reexamination 	Application/Control No. 90/007,858	Applicant(s)/Patent Under Reexamination 5838906
	Certificate Date	Certificate Number <i>C2</i>

Requester Correspondence Address: <input type="checkbox"/> Patent Owner <input checked="" type="checkbox"/> Third Party
STEPHEN A. WRIGHT KLARQUIST SPARKMAN LLP 121 SW SALMON STREET, SUITE 1600 PORTLAND, OR 97204

LITIGATION REVIEW <input checked="" type="checkbox"/>	J.P. <small>(examiner initials)</small>	9/8/08 <small>(date)</small>
Case Name		Director Initials
Eolas Technologies Inc. et al. v. Microsoft Corp. Case No. 1:99cv626, U.S. Dist. Illinois Northern (Chicago)		<i>Eui Hand for</i> <i>CM</i>

COPENDING OFFICE PROCEEDINGS	
TYPE OF PROCEEDING	NUMBER
1. no copending Office proceedings	
2.	
3.	
4.	

Index of Claims



Application/Control No.

90/007,858

Examiner

JOSEPH R. POKRZYWA

Applicant(s)/Patent under Reexamination

5838906

Art Unit

3992

√	Rejected
=	Allowed

-	(Through numeral) Cancelled
+	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claim		Date
Final	Original	
1	1	6/19/07
2	2	✓ 6/19/07
3	3	✓ 6/19/07
4	4	✓ 6/19/07
5	5	✓ 6/19/07
6	6	✓ 6/19/07
7	7	✓ 6/19/07
8	8	✓ 6/19/07
9	9	✓ 6/19/07
10	10	✓ 6/19/07
11	11	✓ 6/19/07
12	12	✓ 6/19/07
13	13	✓ 6/19/07
14	14	✓ 6/19/07
15	15	
16	16	
17	17	
18	18	
19	19	
20	20	
21	21	
22	22	
23	23	
24	24	
25	25	
26	26	
27	27	
28	28	
29	29	
30	30	
31	31	
32	32	
33	33	
34	34	
35	35	
36	36	
37	37	
38	38	
39	39	
40	40	
41	41	
42	42	
43	43	
44	44	
45	45	
46	46	
47	47	
48	48	
49	49	
50	50	

Claim		Date
Final	Original	
51	51	
52	52	
53	53	
54	54	
55	55	
56	56	
57	57	
58	58	
59	59	
60	60	
61	61	
62	62	
63	63	
64	64	
65	65	
66	66	
67	67	
68	68	
69	69	
70	70	
71	71	
72	72	
73	73	
74	74	
75	75	
76	76	
77	77	
78	78	
79	79	
80	80	
81	81	
82	82	
83	83	
84	84	
85	85	
86	86	
87	87	
88	88	
89	89	
90	90	
91	91	
92	92	
93	93	
94	94	
95	95	
96	96	
97	97	
98	98	
99	99	
100	100	

Claim		Date
Final	Original	
101	101	
102	102	
103	103	
104	104	
105	105	
106	106	
107	107	
108	108	
109	109	
110	110	
111	111	
112	112	
113	113	
114	114	
115	115	
116	116	
117	117	
118	118	
119	119	
120	120	
121	121	
122	122	
123	123	
124	124	
125	125	
126	126	
127	127	
128	128	
129	129	
130	130	
131	131	
132	132	
133	133	
134	134	
135	135	
136	136	
137	137	
138	138	
139	139	
140	140	
141	141	
142	142	
143	143	
144	144	
145	145	
146	146	
147	147	
148	148	
149	149	
150	150	



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



Bib Data Sheet

CONFIRMATION NO. 4371

SERIAL NUMBER 90/007,858	FILING OR 371(c) DATE 12/22/2005 RULE	CLASS 708 715	GROUP ART UNIT 3992	ATTORNEY DOCKET NO. 6620-66570-01	
APPLICANTS 5838906, Residence Not Provided; UNIVERSITY OF CALIFORNIA(OWNER), ALAMEDA, CA; Stephen A. Wight(3rd. Pty. Req.), Portland, OR; Stephen A. Wight, Portland, OR					
** CONTINUING DATA ***** This application is a REX of 08/324,443 10/17/1994 PAT 5,838,906 <i>verified ESK</i>					
** FOREIGN APPLICATIONS ***** <i>none ESK</i>					
Foreign Priority claimed: <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS 18 14	INDEPENDENT CLAIMS 26
35 USC 119 (a-d) conditions met: <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Met after Allowance					
Verified and Acknowledged: <i>Eric P. ...</i> Examiner's Signature: <i>E. Pokrzywa</i> Initials: <i>ESK</i>					
ADDRESS 30080					
TITLE DISTRIBUTED HYPERMEDIA METHOD FOR AUTOMATICALLY INVOKING EXTERNAL APPLICATION PROVIDING INTERACTION AND DISPLAY OF EMBEDDED OBJECTS WITHIN A HYPERMEDIA DOCUMENT					
FILING FEE RECEIVED 2520	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:			<input type="checkbox"/> All Fees <input type="checkbox"/> 1.16 Fees (Filing) <input type="checkbox"/> 1.17 Fees (Processing Ext. of time) <input type="checkbox"/> 1.18 Fees (Issue) <input type="checkbox"/> Other _____ <input type="checkbox"/> Credit	

Issue Classification 	Application/Control No. 90/007,858	Applicant(s)/Patent under Reexamination 5838906
	Examiner JOSEPH R. POKRZYWA	Art Unit 3992

ISSUE CLASSIFICATION										
ORIGINAL					INTERNATIONAL CLASSIFICATION					
CLASS		SUBCLASS			CLAIMED			NON-CLAIMED		
709 715		200 205			G	06	F	28 9	128 46	G 06 F 17 30
CROSS REFERENCES										
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)									
345	419	427	619	638	649	653	654	655	656	/
709	202	218	219							/
715	738	760	777	804						/
718	106									/
719	310	315								/
707	E17.119									
JOSEPH R. POKRZYWA PRIMARY EXAMINER CRU - AU 3992 <i>Joseph R. Pokrzywa</i> 9/12/08 (Primary Examiner) (Date)					Total Claims Allowed: 14					
(Assistant Examiner) (Date)					O.G. Print Claim(s) 1			O.G. Print Fig. 9		
(Legal Instruments Examiner) (Date)										

<input checked="" type="checkbox"/> Claims renumbered in the same order as presented by applicant		<input type="checkbox"/> CPA		<input type="checkbox"/> T.D.		<input type="checkbox"/> R.1.47	
Final	Original	Final	Original	Final	Original	Final	Original
1	①		31		61		91
2	2		32		62		92
3	3		33		63		93
4	④		34		64		94
5	⑤		35		65		95
6	⑥		36		66		96
7	7		37		67		97
8	8		38		68		98
9	⑨		39		69		99
10	⑩		40		70		100
11	11		41		71		101
12	12		42		72		102
13	13		43		73		103
14	14		44		74		104
	15		45		75		105
	16		46		76		106
	17		47		77		107
	18		48		78		108
	19		49		79		109
	20		50		80		110
	21		51		81		111
	22		52		82		112
	23		53		83		113
	24		54		84		114
	25		55		85		115
	26		56		86		116
	27		57		87		117
	28		58		88		118
	29		59		89		119
	30		60		90		120
							121
							122
							123
							124
							125
							126
							127
							128
							129
							130
							131
							132
							133
							134
							135
							136
							137
							138
							139
							140
							141
							142
							143
							144
							145
							146
							147
							148
							149
							150
							151
							152
							153
							154
							155
							156
							157
							158
							159
							160
							161
							162
							163
							164
							165
							166
							167
							168
							169
							170
							171
							172
							173
							174
							175
							176
							177
							178
							179
							180
							181
							182
							183
							184
							185
							186
							187
							188
							189
							190
							191
							192
							193
							194
							195
							196
							197
							198
							199
							200
							201
							202
							203
							204
							205
							206
							207
							208
							209
							210