

# Exhibit E



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, 3<sup>rd</sup> 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, 3<sup>rd</sup> 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**

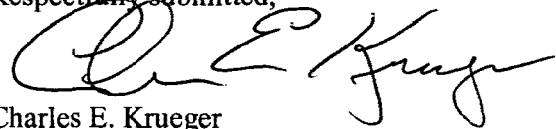
Claims 2-5 depend on claim 1 and are thus allowable for the same reasons as claim 1. Claims 7-10 depend on claim 6 and are thus allowable for the same reasons as claim 6.

**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

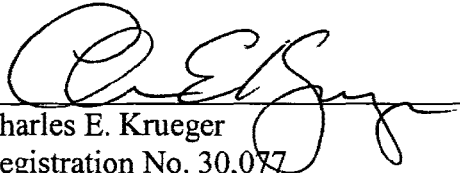
REX Control No. 90/007,858  
Attorney Docket No. 006-1-5

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