

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EBAY, INC¹.
Petitioner,

v.

Lexos Media IP, LLC
Patent Owner.

Case IPR2023-01001
Patent 6,118,449

**PATENT OWNER LEXOS MEDIA IP, LLC'S
RESPONSE PURSUANT TO 37 C.F.R. § 42.120**

¹ The Board joined eBay, Inc. as a party (Petitioner) to this proceeding in IPR2024-00337. Original Petitioner Amazon.com, Inc. has been terminated in this proceeding.

TABLE OF CONTENTS

I. INTRODUCTION	1
II. THE COMMON SPECIFICATION OF THE ‘449 PATENT.....	4
III. LEVEL OF SKILL IN THE ART AT THE RELEVANT TIME.....	12
IV. CLAIM CONSTRUCTION	12
V. CLAIMS 1, 38, AND 53 ARE NOT UNPATENTABLE	14
A. Legal Standards	14
B. Ground 1: Malamud and Knowledge of POSITA Does Not Render Obvious Claim 1 or 53	16
C. Ground 2: The Combination of Malamud and Nakagawa Do Not Render Obvious Claims 1 or 53	49
D. Ground 3: Nielsen and Malamud Do Not Render Obvious Claims 1, 38, or 53.....	52
VI. CONCLUSION.....	64

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Cumberland Pharms. Inc. v. Mylan Institutional LLC</i> , 846 F.3d 1213 (Fed. Cir. 2017)	15
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966)	15
<i>In re Kahn</i> , 441 F.3d 977 (Fed. Cir. 2006)	15
<i>In re Magnum Oil Tools Int’l, Ltd.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	15
<i>Intelligent Bio-Sys., Inc., v. Illumina Cambridge Ltd.</i> , 821 F.3d 1359 (Fed. Cir. 2016)	15
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007)	14, 15

I. INTRODUCTION

Patent Owner Lexos Media IP, LLC (“Patent Owner,” or “Lexos Media”) respectfully submits this response to eBay’s Petition for *inter partes* review (Paper 2 in IPR2024-00337, now joined with this proceeding, “Petition”) filed by Petitioner eBay, Inc. (“Petitioner” or “eBay”) against Claims 1, 27, 38, and 53 of U.S. Patent No. 6,118,449 (the “’449 Patent”) and the Board’s Decision Granting Institution (“Institution Decision”) (Paper 12²).

The Board instituted review on three grounds that challenged Claims 1, 38, and 53 of the ‘449 Patent. Patent Owner does not seek to amend or disclaim these claims, which each remain separate and pending claims in this proceeding. As discussed in detail below, Petitioner has not met its burden of proving unpatentability of the challenged claims. Pursuant to 35 U.S.C. § 316(e), Petitioner has the burden to prove their proposition of unpatentability by a preponderance of the evidence.

As discussed in greater detail below, eBay’s Petition fails to show Claims 1, 38, and 53 of the ‘449 Patent are rendered obvious by any of the disclosures

² Reference will also be made to Paper 9, as that original institution decision is heavily referenced in the Institution Decision (Paper 12).

concerning (1) Malamud in light of the knowledge of a POSITA, (2) Malamud when combined with Nakagawa, and (3) Nielsen when combined with Malamud.

The Petition improperly attempts to transform Malamud from what is actually disclosed, *i.e.*, a local data processing system based on an operating system extension using locally stored data to change a cursor, to a server-based system that allegedly renders claims 1, 38 and 53 of the '449 Patent obvious. In doing so, Petitioner largely reads out the claim language which provides both context and meaningful limitations to the claims which are not disclosed or even suggested by Malamud. At the most basic level, Malamud has no disclosure nor hint of a suggestion of its local data processing system being “connected to at least one server,” “receiving a request at said at least one server...” or the server “providing said specified content information to said user terminal in response to said request...” each of which is expressly set forth in these claims. Indeed, Malamud does not even disclose any type of network interface that would facilitate this function. Petitioner glosses over this fundamental difference in operation arguing that a POSITA would understand that an “application” in Malamud could be downloaded from a server and this non-disclosed hypothetical download of an application would not only be made in response to a “request to said at least one server” for “specified content information” but would also include the claimed “content information” and “cursor image data

corresponding to a specific image,” despite Malamud only disclosing that such content is stored locally by the operating system. These positions are factually and legally wrong. Petitioner is not properly using the knowledge of a POSIA simply to understand the art and make an “obvious” modification thereto. Instead, under the guise of the knowledge of a POSIA and aided by hindsight, Petitioner is fundamentally altering the disclosure of Malamud in its effort to reconstruct the limitations of claims 1, 38 and 53 of the ’449 patent.

The Petitioner argued, and the Board accepted for the sake of institution, that a Windows application program invoking Malamud’s window procedure may be generating an icon for a stored file and therefore could meet the “information to be displayed” limitation of all Challenged Claims. This argument was based solely upon a general discussion within Malamud regarding how operating systems create one or more windows for Windows-based application programs. Because Malamud, as a factual matter, fails to expressly or inherently disclose that the application program (rather than the operating system) generates the icon, the Petition fails on Grounds 1-3.

The Petitioner also argued, and the Board accepted for the sake of institution, that downloading a Windows application program that invoked Malamud’s window procedure from a server would result in the practice of Challenged Claims and

thereby render them obvious. Because downloading such an application program does not result in the practice of the Challenged Claims, including for failure to meet the construction of “specified content information” required by all Challenged Claims, the Petition also must fail on Grounds 1-2.

The Petitioner further argued, which the Board did not accept, that even if downloading a Windows application program that invoked Malamud’s window procedure from a server did not result in the practice of the Challenged Claims, Nielsen’s tooltip being modified to an information cursor would so. Because Petitioner does not demonstrate that it was within the skill of a POSITA to combine Malamud’s operating system-based technology with Nielsen’s browser-based technology and a POSITA would also not have been motivated to make such a modification of Nielsen because it would not improve Nielsen, the Petition fails on Ground 3.

For these reasons, among others explained below, Claims 1, 38, and 53 are not unpatentable.

II. THE COMMON SPECIFICATION OF THE ‘449 PATENT

The ’102 and ‘’449 Patent share a common specification having a priority date of June 25, 1997 (“Common Specification”).³ At the time of the patents’ invention,

³ For ease of reference, patent citations to the Common Specification will be with

more and more companies were starting to advertise on the Internet. ‘102 Patent, 1:21-23. Three typical forms of advertising at the time of the invention were “banner ads,” ‘102 Patent, 1:24-25, “frames,” ‘102 Patent, 1:55-56, and “self-appearing windows.” ‘102 Patent, 2:4-5. Each form of advertising had its drawbacks and limitations, as explained in the specification. See ‘102 Patent, 1:24 to 2:32. Consequently, the inventors recognized “a need for a simple means to deliver advertising elements . . . without the annoyance of totally interrupting and intrusive content delivery, and without the passiveness of ordinary banner and frame advertisements which can be easily ignored.” ‘102 Patent, 2:27-32. Thus, it was a specific objective of the inventions “to provide a server system for modifying a cursor image to a specific image displayed on a video monitor of a remote user’s terminal for the purposes of providing on-screen advertising.” ‘102 Patent, 2:44-47. The inventions claimed in the Asserted Patents provide for server systems that enable a cursor or pointer displayed on a remote client’s computer to change appearance to a specific image that is related to the content being transmitted to and displayed on that computer. ‘102 Patent, 4:4-12. The modified cursor image can be used by the provider of displayed web content to advertise or promote the displayed content. ‘102 Patent, 3:51 to 4:3.

reference to the column and line numbering of the ’102 Patent.

These customized webpage cursor images could include multiple components, including components that were spaced apart from the primary cursor image and were not of a normal size or in a normal cursor location and which had their own movement that tracked the movement of the modified cursor image. *See* EX. 2021, Expert Declaration of Samuel H. Russ, Ph.D. (“Russ Decl.”), ¶ 44.

For example, the specification provides an exemplary embodiment detailing a webpage advertising a product, Fizzy Cola, wherein the webpage displays an image of a Fizzy Cola bottle as part of its content and the webpage’s cursor image is modified to a dynamic cursor image showing a person holding a straw wherein the straw always points toward the top of the Fizzy Cola bottle, regardless of where the cursor moves on the screen. *See* ‘102 Patent at 17:15-31. As the specification explains in this passage, the straw may either be part of the same cursor image or could be a separate "satellite" image, a "sprite," whose movement on the screen is related to the movement of the cursor. *Id.* The patent also explains that the sprites may appear and disappear as desired and “enhance the invention by enabling the use of graphical elements which are associated with the cursor but which reside outside the limited cursor "space" (which in some systems may be, at maximum, 32 by 32 pixels). For the purposes of the invention, however, there should be no limitation to the size of the cursor.” *Id.* at 17:25-31. Importantly, the cursor’s features are tied to

the content being displayed by the browser. (*Ibid.*) *See also*, Russ Decl., ¶ 45. This teaching of non-traditional cursor images used for promotional purposes is reinforced throughout the specification. For example, Cursor Display Instructions that include parameters that control the type of cursor image for the modification, how long the changed image should remain before reverting, the changed image's trajectory path, where the image should change based upon its screen location and what satellite images should be used that track the cursor's movement. 9:39-65. *See also*, Russ Decl., ¶ 46. Large portions of the Common Specification are directed to teaching how to implement non-standard cursors on webpages, and the specification devotes all of columns 9 through 13 to this task, explaining to a POSITA that browser extensions and plug-ins such as ActiveX, JavaBeans, JavaScript or VBScript would be used to implement the desired cursor modifications. *See also*, Russ Decl., ¶ 47.

As the specification explains, cursor display instructions are initially embedded inside of an HTML document, *e.g.* a webpage. Then the browser processes the instructions, retrieves cursor display code (ActiveX) and cursor information to display the specific image via display drivers. The specification in FIG. 4 provides exemplary cursor display instructions written for ActiveX. '102 Patent, EX. 1001, at 9:1-63. *See also*, Russ Decl., ¶ 48.

Further detail regarding implementing a cursor modification on a webpage is provided throughout the specification, and particularly through columns 10-13. This level of detail regarding the implementation of the cursor modification on a webpage is necessary for a POSITA in 1997 to accomplish the modification without undue experimentation, particularly given that the Common Specification expressly discloses dynamically downloadable and executable code, such as ActiveX or JavaScript, would be needed to implement a system that could respond dynamically both to user cursor movements and to Web-based content to be displayed. *See also*, Russ Decl., ¶ 49.

The object of the invention is also clearly stated in the specification. At the time of the invention, more and more companies were advertising on the Internet. '102 Patent, 1:21-23. Three typical forms of advertising at the time of the invention were “banner ads,” 1:24-25, “frames,” 1:55-56, and “self-appearing windows.” 2:4-5. Each form of advertising had its drawbacks and limitations, as explained in the specification. 1:24- 2:32. Consequently, the inventors recognized “a need for a simple means to deliver advertising elements . . . without the annoyance of totally interrupting and intrusive content delivery, and without the passiveness of ordinary banner and frame advertisements which can be easily ignored.” 2:27-32. Thus, achieving webpages that utilize customized cursor images to provide “on-screen

advertising” was the fundamental objective of the disclosure. 2:44-47. Therefore, how to implement the browser-based cursor modification of the claimed inventions had to be carefully taught because browsers did not have such specific functionality already included as part of the HTML protocol. This is in contrast with tooltips, wherein all the web page developer had to do was enter HTML tags in order to achieve a tooltip. Russ Decl., ¶ 50.

Thus, the claimed inventions provide for server systems that enable a cursor or pointer displayed on a remote client’s computer to change its appearance to a “specific image” that is representative of the content being displayed on that client computer. 4:4-12. The modified cursor image can be used by the provider of displayed web content to advertise or promote the displayed content. 3:51-4:3. As the Common Specification explains, “[t]he present invention provides a means for enabling cursors and pointers to change color, shape, appearance, make sounds, display animation, etc., when the user’s terminal or computer, known as the ‘client’ or ‘user’ terminal, which has a network connection, receives certain instructions from a remote or ‘server’ computer attached to the network” (3:51-57) whereby “the generic cursor or pointer icons used in many networking applications, such as black arrows, hands with a pointing finger, spinning wheels, hourglasses, wristwatches, and others, will change appearance, and in some cases may incorporate sound or

animation, in a way that is linked and related to the content, such as a webpage, which is being transmitted to and displayed on the client computer.” `102 Patent, 3:57-64. As a result, the cursor “convey[s] a message that relates to the advertising content within the webpage being transmitted and displayed.” 3:67-4:3. *See also*, Russ Decl., ¶ 51.

As extensively described in the Common Specification (*e.g.* 7:41-14:64), when a browser requests a webpage from a server, it receives back an HTML file that includes content to displayed and/or pointers thereto, formatting instructions on how to display the content, and instructions and/or code specifying how the webpage is to interact with the user, including how the cursor modification takes place and specific images are displayed. The browser then uses the received information to generate the webpage displayed to the user, including the shape and appearance of the modified cursor image. *See also*, Russ Decl., ¶ 54.

As the Common Specification also describes, the content of the specific image may be dynamically updated 17:55-61 (“It may also be desirable in certain cases to put alphanumeric data in the cursor ‘space’ to convey information to users, such as stock prices, baseball game scores, the temperature in Florida, etc. The data can be static, semi-static (i.e. updated periodically), or dynamic (updated frequently --

possibly incorporating available streaming-data and data compression technologies.”) *See also*, Russ Decl., ¶ 55.

Thus, the Asserted Patents disclose and enable systems and methods in which cursor images are dynamically modified for promotional and advertising purposes. In the ’102 and ’449 Patents, the specific shape and appearance of the modified cursor image corresponds to other content being displayed on the webpage to promote that content. The modified cursor image may include a dynamic visual image that tracks the movement of a cursor. The various examples disclosed in the Common Specification of promotional cursors include that dynamically update in response to the cursors’ movements and associated images that are spaced apart from with the modified cursor image and track the movement of a cursor. Russ Decl., ¶ 56.

Through these teachings, the Common Specification of the ’102 and ’449 Patents provided an important benefit to Internet retailers who used these inventions to promote internet sales at a time when most shoppers were not familiar with buying products over the internet rather than in a brick-and-mortar environment. Russ Decl., ¶ 57.

III. LEVEL OF SKILL IN THE ART AT THE RELEVANT TIME

The Board adopted the following level of ordinary skill in the art in 1997: “A person of ordinary skill in the art (“POSITA”) at the claimed priority date would have had experience in the fields of human factors engineering or human computer interaction. The POSITA would have at least a bachelor’s degree in computer science, computer engineering, human factors engineering, or a related field and would have had at least two years of relevant work experience in the fields of UI design, or equivalent experience.” Paper 9 at 5. For purposes of this proceeding only, Patent Owner does not dispute the proffered definition for a POSITA. *See also* Russ Decl., ¶ 32.

IV. CLAIM CONSTRUCTION

As detailed further herein, Petitioner did not give proper effect to the meaning of the claim terms. Patent Owner applies in this response the following constructions that were either agreed upon by Original Petitioner and Patent Owner or issued by the United States District Court for the Eastern District of Texas:

CLAIM TERM	PARTIES' AGREED CONSTRUCTION
“said specific image includes/including content corresponding to at least a portion of said information that is to be displayed on said display of said user’s terminal” See ‘102 Patent, Claim 72; ‘449 Patent, Claims 1, 38, 53.	“an image representative of at least a portion of the subject or topic being displayed on the screen” <i>Lexos Media IP, LLC v. APMEX, Inc.</i>, No. 216CV00747JRGRSP, 2017 WL 1021366, at *5–6 (E.D. Tex. Mar. 16, 2017).
“specified content information” See ‘102 Patent, Claim 72; ‘449 Patent, Claims 1, 38 and 53.	The term “specified content information” / “content information” means “information provided to a user’s terminal for use in the display, such as a web page.”

See EX. 1008.

CLAIM TERM	THE COURT’S CONSTRUCTION
“cursor display code” (‘102 Patent, Claim 72; ‘449 Patent, Claims 1, 27, 53)	“computer code for modifying the display of the cursor image”
“cursor display instruction” (‘102 Patent, Claim 72; ‘449 Patent, Claims 1, 27, 53; ‘241 Patent, Claim 35)	“an instruction operable to modify the display of a cursor image”

“cursor image,” “initial cursor image,” “modified cursor image” (‘102 Patent, Claim 72; '449 Patent, Claims 1, 27, 53; '241 Patent, Claim 35)	“a movable image on a display screen whose position can be controlled through a user interface”
---	---

See EX. 2008 at 22-23. See also, Russ Decl., ¶¶ 33-34.

V. CLAIMS 1, 38, AND 53 ARE NOT UNPATENTABLE

Petitioner raised three grounds in its petition, each of which were instituted.

The status of the grounds is as follows:

Ground	35 U.S.C. §	References	Challenged Claim	Status
1	Pre-AIA 103	Malamud	1,27 ⁴ ,53	Disputed
2	Pre-AIA 103	Malamud, Nakagawa	1,27,53	Disputed
3	Pre-AIA 103	Nielsen, Malamud	1,27,38,53	Disputed

A. Legal Standards

An invention “composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope

⁴ As noted by the Board, Claim 27 was previously canceled, so argument and reference is made only due to Claim 38’s dependency thereupon. See Paper 12 at 6.

and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) objective evidence of nonobviousness. *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17–18 (1966); *see also* Institution Decision (Paper 9) at 13-14.

An obviousness determination “cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *KSR*, 550 U.S. at 418 (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)); *see In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016). “[I]t can be important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does.” *KSR*, 550 U.S. at 418.

The Federal Circuit has been clear that motivation to combine references and reasonable expectation of success are separate and distinct elements of the obviousness analysis: “one must have a motivation to combine accompanied by a reasonable expectation of achieving what is claimed in the patent-at-issue.” *Intelligent Bio-Sys., Inc., v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1367 (Fed. Cir. 2016) (emphasis added); *see also id.*, 1365 (describing the “requirement for a reasonable expectation of success” in the context of obviousness); *Cumberland*

Pharms. Inc. v. Mylan Institutional LLC, 846 F.3d 1213, 1221 (Fed. Cir. 2017) (describing motivation to combine and reasonable expectation of success as separate requirements for obviousness analysis).

B. Ground 1: Malamud and Knowledge of POSITA Does Not Render Obvious Claim 1 or 53

With respect to **Ground 1**, Petitioner asserts that Claims 1, 27, and 53 are unpatentable as allegedly being obvious over Malamud in light of the knowledge of a POSITA, and the Board found that Petitioner had established a reasonable likelihood it would prevail in showing Claim 1 and 53 were unpatentable thereunder. As demonstrated below, Claim 1 and 53 are not obvious and the Petition does not establish Ground 1. The Petition does not include Claim 38 in its allegations of obviousness under Ground 1. *See*, Pet. at 6; Paper 12 at 8.

1. Overview of Malamud (EX. 1004)

Malamud is a Microsoft patent that extends the existing Windows 3.1 operating system (or a comparable window-based operating system) by adding a proposed new window procedure. A Windows Application can invoke the new window procedure so that metadata about a named file stored on the computer (Fig. 1, “data processing system 10”) that would normally be displayed in the status bar of a window could instead be displayed in a text or image box of an “information cursor” when the user moves the cursor over the file’s icon in a desktop type

window. *See, e.g.*, Malamud, EX. 1004; Background of the Invention at col. 1, lines 15-31; Summary of the Invention at col. 1, lines 44-57; Figs 3-5 (book icon 32). *See also*, Russ Decl., ¶ 59.

Malamud expressly describes that the text file associated with book icon 32 is a pre-existing file stored on the computer, with a pre-existing icon:

FIG.2b shows an example of how the name cursor 26 is used. **Suppose that a book icon 32, representing a book of text stored in the system 10 (FIG. 1),** is displayed on the video display 20. When the pointing portion 28 of the name cursor 26 points to the book icon 32, the name box 30 of the name cursor displays the phrase "Book Cover".

EX. 1004 at 3:35-38 (emphasis added); *see also* Russ Decl., ¶ 60.

Malamud's specification explains that the function of the new window procedure is to "pass[a] message to operating system specifying info[rmation] about [the] object." (Box 58 in Fig. 7). It describes how this window procedure operates:

After the window procedure has determined what is at the specified cursor position, the procedure passes a message to the operating System 22 (FIG. 1) that tells the operating system what type of cursor to display and sets forth the contents and type of information to be displayed in the cursor (step 58 in FIG. 7).

EX. 1004 at 5:46-53; *see also* Russ Decl., ¶ 61.

Pointedly, Malamud describes this proposed window procedure separately from the program responsible for creating the window. The latter sends a message to the former for processing. As Malamud explains:

When an event occurs, the event is translated into a message that is put into the message queue for the program. The program retrieves and delivers the message to the proper window procedure by executing a block of code known as the “message loop”. The window procedure that received the message then processes the message.

EX. 1004 at 4:59:-65; *see also* Russ Decl., ¶ 62.

In other words, the “window procedure” is a built-in operating system function, with an API to an application program to allow the latter to invoke this window procedure. Russ Decl., ¶ 63.

A POSITA would understand from Malamud’s specification as a whole that the window procedure does not generate the icon or even the metadata to be inserted into the information cursor. Rather, a POSITA would understand that the Windows 3.1 operating system maintains metadata about a stored file and generates a “desktop” window that displays icons representing files and applications that the user can select. *See also*, EX. 1003-1, Declaration of Dr. Rosenberg (“Rosenberg Decl.”) at ¶ 40 (“a popular type of GUI is the “desktop metaphor,” where the computer screen represents a virtual desktop with icons representing files and applications that users can select.) Russ Decl., ¶ 64.

For example, the Windows 3.1 operating system by itself generates various user configurable “desktop” windows for “accessories,” “games,” “startup” or any other category that the user wishes to have. These desktop-type windows display the icons for application and data files (such as text files) stored on the computer.



Russ Decl., ¶ 65.

As these above screenshots of Windows 3.1 generated windows reflect, Windows 3.1 is itself a program that generates user configurable windows that display icons associated with stored application and text files. A POSITA understands that operating systems such as Window 3.1 maintain metadata about a stored file that they can displayed in the status bar or, as posited by Malamud's proposed extension to Windows 3.1, in an information cursor. Russ Decl., ¶ 66.

In sum, Malamud proposes an extension to Windows 3.1 or similar operating system to provide a new method of displaying metadata ***maintained by the operating system*** for a file stored on the computer that is running that operating system. Instead of the operating system displaying file information (aka "metadata") regarding that file in the status bar on the periphery of the window, the file's metadata can be displayed in an information box of an information cursor when the operating system or an application program invokes the new window procedure. Russ Decl., ¶ 67.

2. Errors in Dr. Rosenberg's Analysis of Malamud

Importantly, Dr. Rosenberg does not express any opinion that Malamud's application program provides the icon of the stored file. This alone is fatal to his analysis. See EX. 1003-1 at ¶ 102 (only testifying that "that the application program would contain cursor display instructions that instruct the performance of the

functionality discussed below.”); ¶ 104 (only testifying that “the application program would include information to be displayed on the display of the user’s terminal.”); ¶ 105 (only citing to graphical image displayed in the preview cursor---not the icon.); ¶ 108 (again not reading the “information to be displayed” element onto the icon) Russ Decl., ¶ 148; *see also* Russ Decl., ¶¶ 111, 114.

The Petitioner’s sole reliance for the proposition that Malamud’s application program displays the icon is a general statement that when a Windows operating system opens a Windows application program, it creates a new window and assigns a window procedure for that window. *See*, EX. 1004 at 5:53-59 (explaining the operating system “logically divides the user interface into a number of windows” wherein “each window has a separate window procedure associated with it” and “a message queue for each program that generates windows.”) But this passage does not describe or state that the Windows application program supplies the icon or create the content displayed in the window and such a reading is contrary to Malamud’s specification. Malamud describes a new window procedure that is part of the Windows 3.1 operating system, and not the window program. Moreover, a POSITA would understand that an application program does not create or contain all the content in a window that it generates. Rather, a POSITA understands that within the content within the generated window may be supplied by the application

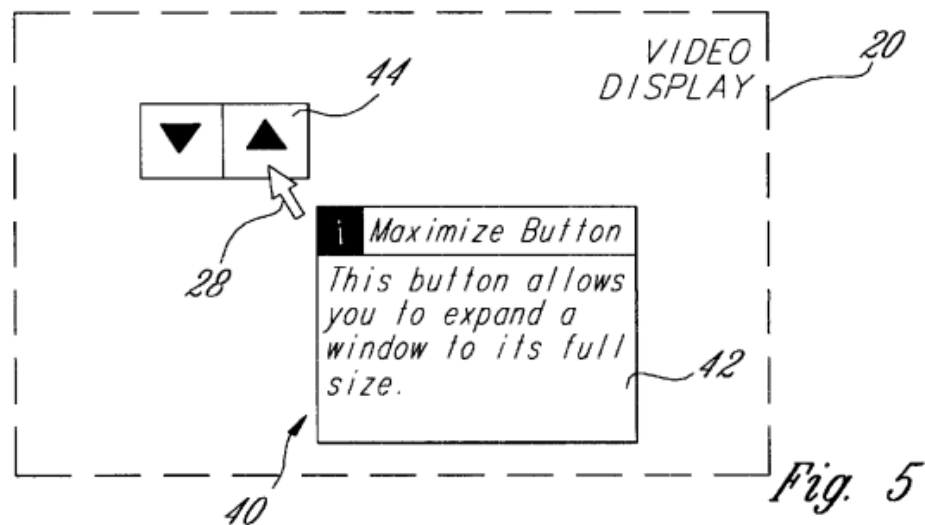
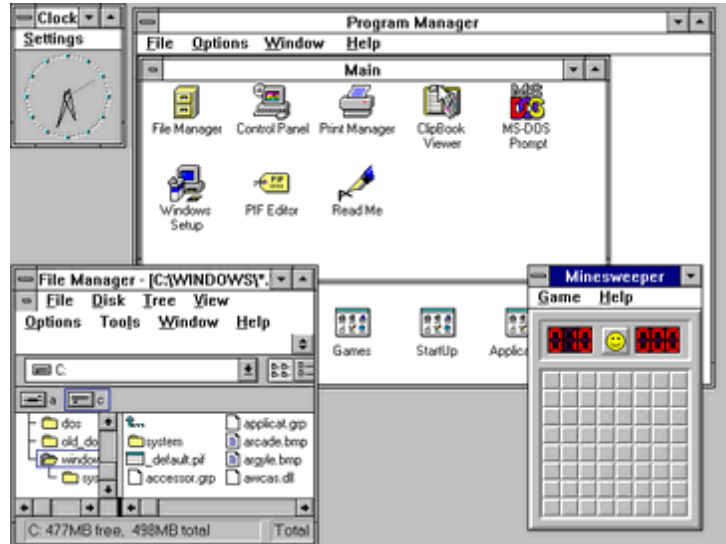
program or by the operating system. For example, icons and metadata for windows functions or stored files are supplied by the operating system. The sole passage relied upon by Petitioner for this element therefore provides no express or inherent support for the proposition that the icon was generated by the application program. Russ Decl., ¶ 150. Moreover, for reasons further discussed herein, a POSITA would not consider the request for such an application to be a request for the claimed “specified content information.” Russ Decl., ¶ 151.

Dr. Rosenberg’s testimony is contrary to the understanding of a POSITA reading Malamud’s specification. Russ Decl., ¶ 68. Malamud specifically describes a proposed extension to the Windows 3.1 operating system that adds a new window procedure to support the use of information cursors in order to display metadata for a named entity (*e.g.*, a file or object for which there is metadata and an icon).

First, a POSITA reading Malamud understands that the window procedure is part of a Windows operating system and can be invoked by the operating system itself or by a Windows application program. *See* Ex. 1004 at 5:65-6:17 (explaining that whether the information cursor is displayed depends on whether the information cursor is designated as “on” or “off” may be an option given the user or “may be

programmed by the application program *or the operating system.*”) Russ Decl., ¶ 69 (emphasis supplied.)

Moreover, a POSITA would understand that Windows operating systems, such as Windows 3.1, create desktop windows that display icons of stored files. Compare, Malamud Fig. 5 below with this Windows 3.1 generated desktop window:



Russ Decl., ¶ 71.

As can be seen from the comparison, the up down arrows 44 of Fig. 5 are from a Windows 3.1 generated window. Moreover, Malamud explains that the bitmap of the information cursor (28 and 40) is also provided by the operating system.

Likewise, compare the Windows 3.1 generated desktop window above with Malamud's Fig. 4 (showing a portion of the video display screen of the user's computer):

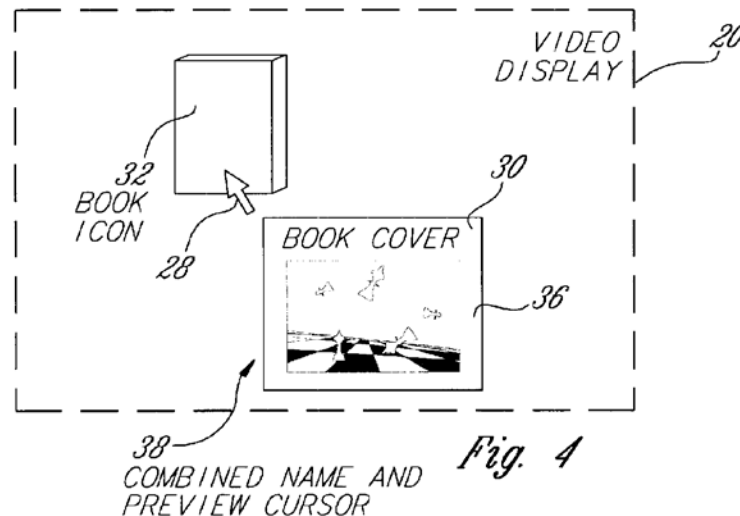


Fig. 4 is consistent with the Windows 3.1 desktop window and further explains to a POSITA that Microsoft, the assignee and developer of the Malamud application, was concerned with upgrading its Windows 3.1 operating system so that the user has the ability to “mouse over” an already displayed icon (32) in the desktop window in order to see in the information portion (38) of an information cursor (28,38) the metadata (30, 36) regarding the associated file rather than in a status bar. The Malamud specification is tied very tightly to the Windows 3.1 operating system (or a comparable windows-based operating system) with a new proposed window procedure which determines whether an icon for a stored file is a “named object” (*i.e.*, one that has associated metadata in the operating system), selects the

appropriate bitmap for the information associated with that metadata, and sends the command to the display drivers to display the bitmap populated with the associated metadata. Russ Decl., ¶¶ 72-74.

Specifically, a POSITA would understand that when an Windows-based application program, such as Corel's WordPerfect program (a popular windows-based text editor in 1997) or Microsoft's Word program (also a popular text editor both then and today) has been opened, such as by double-clicking on its icon displayed on a Windows 3.1 desktop window, Windows 3.1 opens a new window for that program with the command buttons and other features displayed therein. The user can create a new text file in the open window and use the application's functionality to store the text file. Furthermore, a POSITA would understand that when the file is stored, Windows 3.1 will assign and manage an icon in association with that file. In this case, because the file was created in Word, Windows 3.1 will associate a Word icon with the file as part of its metadata. Russ Decl., ¶ 75.

Second, a POSITA reading Malamud would understand that the Windows-based program merely invokes the window procedure to turn on the information cursor functionality, with the two available options regarding when the information cursor is displayed. For example, let's assume that the user or Microsoft configured the Word program to turn on the Windows 3.1's information cursor extension.

When a user is later trying to open the Word file they created from within Word's window, the application has no need to send the icon associated with the file or its metadata to the window procedure because the operating system already has the icon and other file metadata from the time that it was first stored thereupon. Hence, Malamud does not describe the application program sending the file's icon or metadata for display in the information portion of the information cursor to the window procedure. Russ Decl., ¶ 76.

Therefore, the Petition is incorrect in its assumption that if a POSITA chose to download an application that invoked the information cursor window procedure of Malamud, the application would necessarily include either the icon for the file or the file's metadata to be displayed within the information cursor. Russ Decl., ¶ 77.

Malamud's window procedure has a very narrow functionality, namely to receive the cursor position, determine whether the icon of a named entity is located at the cursor position and, if so, convey to the display driver which type of information cursor is to be displayed and what file metadata is to be displayed therein. In other words, the proposed Windows extension simply has no need to create file icons or other file metadata for stored files because the operating system already has and manages that information for a stored file. Russ Decl., ¶ 78.

In paragraph 84 of his declaration, Dr. Rosenberg conflates functions of the proposed Windows 3.1 information cursor ***extension*** with functions of the application program ***using that extension***. Malamud describes that the operating system sends a message providing the cursor location that is routed by the application program to the window procedure. The proposed window procedure decides based upon the cursor position as to “whether a named entity is present at the specified cursor position.” EX. 1004 at 5:35-36. “After the window procedure has determined what is at the specified cursor position, the procedure passes a message to the operating System 22 (FIG. 1) that tells the operating system what type of cursor to display and sets forth the contents and type of information to be displayed in the cursor (step 58 in FIG. 7).” *Id.* at 5:46-53; *see also*, Russ Decl., ¶ 80.

The Windows application program, on the other hand, merely invokes the window procedure and sets whether an information cursor is to be displayed or not. If set to “on”, the information box of the information cursor is always displayed but is blank when the cursor position is not over icon and only populated when the cursor position is over the icon. If set to “off,” then the information cursor is only displayed (and with its content/metadata) when the cursor position is over the icon. *See*, EX. 1004, at 5:65-6:17; *see also*, Russ Decl., ¶ 81.

Every active function is carried out by the Windows operating system; the icons are managed by Windows, and the information in the cursor is metadata for the information cursor which is managed by Windows. Indeed, even “the appearance of the information cursors...is dictated by bitmaps stored within the operating system 22” (5:15-17); *see also*, Russ Decl., ¶ 82.

In sum:

a. Malamud only describes that the Windows application program invoking the information cursor window procedure can only turn the information cursor “on” or “off.” There is no description in Malamud of the window procedure being designed to receive either icons or the metadata for display within the information portion of the information cursor from the Windows application program invoking it.

b. The Windows 3.1 operating system did not need the Malamud extension to display icons or metadata for a file. The Malamud extension is merely for allowing the operating system to display these preexisting items in an information cursor, rather than in a status bar. The only API to the windows procedure described in Malamud is to turn the information cursor window procedure “off” or “on.” Therefore, the API would have to be modified to allow a Windows-based program to provide an icon or metadata

to the windows procedure. This change would require approval and implementation by Microsoft, or the development by a POSITA of their own operating system designed to interact with the Windows application program in this manner. Neither of these options would have involved any reasonable amount of effort and without undue experimentation or have reasonable or predictable chance of success.

c. Downloading a Windows-based application program from a server that invokes the Malamud window procedure extension would not result in the download of any icon or any metadata about a file for display in the information box of the information cursor. These icons and metadata are preexisting and stored by the operating system itself, as are the files that create the information cursor outline and shape. There is simply no description or disclosure of the extension itself having any pre-loaded icons, information cursors, or content whatsoever. Nor would it make sense for Malamud to create icons, information cursors, or content for files that already exist within the operating system. Being the architect of both Windows 3.1 and Malamud, Microsoft understood this and provided no description within Malamud concerning the creation of items that already exist. And a POSITA runs into

the same problem when trying to modify the extension to receive this information for the same reasons as described in the paragraph above.

Russ Decl., ¶ 83.

The Original Petitioner and the Patent Owner stipulated that the term “specified content information” / “content information” means “information provided to a user’s terminal for use in the display, such as a web page.” EX. 1008. This construction is consistent with the Common Specification, which is directed to serving online content to be displayed, such as a webpage. *See, e.g.*, EX. 1001 at Background of the Invention (discussing the problems with on-line advertising being addressed by the invention); 3:15-21 (the invention is directed to “providing a system for delivering advertising elements online”); Figs. 7-9 (showing the webpages with an initial cursor image and transformed specific cursor image); 5:58-65 (discussing the browser and that the “server 12 functions as a so called ‘web site’”) 6:7-18 (explaining browsers working dynamically at “run-time” versus statically at “compile-time”); 8:38-11:60 (discussing how the browser is used to transform the cursor image on the webpage using plug-ins such as Active X). Russ Decl., ¶ 100.

A POSITA would not consider a downloaded application to be any of the claimed “specified content information” (webpage file), “information to be

displayed” (webpage content over which the initial cursor is moved to trigger transformation thereof into the shape and appearance of the specific image), “cursor display code” (Active X or JavaScript), or cursor display instructions (URL for Active X or JavaScript file). Rather, the POSITA would see it as information to be installed because a Windows application program cannot run to invoke Malamud’s window procedure without first being installed. An application program is not like a webpage or other online content that a browser can dynamically render, and it is therefore unsuitable to the purpose of serving webpages with promotional cursors. Russ Decl., ¶ 101. Also, downloading the Windows application program described in Malamud would not result in the downloading of either the specific image (information cursor content) or the information for display (icon) that triggers that transformation of initial cursor image to the corresponding specific image because the associated icon and metadata for a file stored at the computer are created at the time the file is stored and already managed by the Windows 3.1 operating system. Consequently, downloading Malamud’s application program from a server would not comprise a request to a server to provide specified content information to a user terminal. Russ Decl., ¶ 102; see also, Russ Decl., ¶¶ 85-86. Serving a Windows application program from a server therefore would not address Malamud’s shortcomings. Russ Decl., ¶ 87.

No motivation exists to modify Malamud to add the claimed server/client architecture. Malamud is designed to function using a pre-existing operating system; serving the Windows application program from the server is simply not required to achieve the goals of Malamud. It is also for this reason that it would not have been “obvious to try” adding a server/client architecture to Malamud, as a POSITA would never have been inclined to attempt to modify Malamud in the first instance, let alone try several different modifications. Furthermore, Windows application programs were/are large files in comparison to HTML webpage files and the cost for internet bandwidth in 1997 was too high to make downloading these files feasible. Indeed, Fig. 1 of Malamud does not include any network interface by which such downloading could take place. It simply would not make sense to a POSITA to modify Malamud to download an application program in some attempt to implement Claim 72 because of the problems such modifications would cause and because there would be no motivation to do so. Russ Decl., ¶¶ 88-90.

3. Petitioner Fails to Establish Obviousness using Malamud and the Knowledge of a POSITA for Claims 1 and 53

As an initial matter, obviousness of the claim cannot properly be assessed by breaking a claim apart into its constituent claim elements and making some argument why modification of each particular claim element would be obvious. Rather one must look at the claim as a whole and the prior art as a whole and then comparing

the entirety of its teachings against the entire claim when comparing their similarities and differences in arriving at a conclusion. In determining the differences between the prior art and the claims, the question under 35 U.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); emphasis by MPEP § 2141.

When Claim 1 and 53 are viewed as a whole and compared to the totality of Malamud's teachings, Claim 1 and 53 of the '449 Patent are not rendered obvious in view of Malamud and the knowledge of a POSITA. Russ Decl., ¶¶ 91-97.

Element 1[Preamble]: A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

Malamud does not disclose or render obvious a server system for modifying the display of an initial cursor image on a remote terminal into the shape and appearance of a specific image on a remote user terminal as required by element 1[Preamble] for at least all the reasons discussed above.

Malamud does not disclose any server system. Malamud is a Microsoft patent that discloses a new window procedure for the Windows 3.1 operating system (or a comparable system). Either the operating system itself or a Windows application program can invoke the window procedure to display the metadata of a stored file in

an information cursor instead of in a status bar. The operating system or Windows application program can set the procedure to always display the information cursor or to only display it (and the metadata to be displayed within it) when the cursor position is over the icon of a stored file. The Windows application program does not communicate to the windows procedure (a) any specific image to display in the information cursor; (b) an icon; or (c) the information cursor. All of this data is understood to a POSITA to be maintained by the operating system for every stored file having such associated data.

A POSITA would not have been motivated to modify Malamud in order for the Windows application file to supply the icon and metadata for a stored file provided by the Windows application itself, because any such change would necessarily involve changing the design of the API to the window procedure, and therefore the operating system. Such a change would have to be approved and made by the operating system's manufacturer, which is not something that a POSITA could predictably or reliably accomplish. Alternatively, a POSITA would have to design their own operating system to effectuate this modification of Malamud, which would also be beyond the capability of a POSITA without undue effort.

Additionally, even if a POSITA could effectuate the change to the Windows operating system and provide a Windows application program that contained the data to

create the information cursor, a file icon, and the metadata to be displayed in the information cursor, this would still not result in the practice of the claimed invention because there would still be no request for “specified content information,” as further discussed below, and downloading an application program would not work to practice the claim. Also, , unlike the claimed specified content information, such as a webpage, a Windows application program would typically be a large file and not as prone to download in 1997 because of internet costs and bandwidth constraints, making the argument even more far-fetched. *See*, Russ Decl., ¶¶ 122-126.

Element 1[a]: cursor image data corresponding to said specific image;

This element is considered in detail below with Element 1[c][ii]. That said, Malamud does not disclose or render obvious Element 1[a] (when read in connection with the rest of the claim’s requirements), because Malamud does not disclose or suggest to a POSITA a server to provide specified content information to the user’s terminal wherein said specified content information cursor image data corresponding to a specific image.

A Windows application program that invoked (or even contained) Malamud’s extension to Windows 3.1 sent from a server in response to a request for the program would not meet this claim element for the reasons previously stated. The application program is not “information to be used in the display, such as a webpage” per the

agreed upon construction. A POSITA understands that an application program does not operate like a webpage or banner ad or other online content for display that is suitable for the claimed integrated display method which can be achieved by a browser and webpage server because a Windows application program would have to be installed in order to communicate with Malamud's extension to Windows 3.1 operating system, making it unsuitable for purpose of the claimed server system.

Also, Malamud does not describe any interface in which the application program communicates to the window procedure a specific image (*i.e.*, the metadata to be displayed in the information cursor) for a file stored on the computer. Indeed, because the window procedure is part of the operating system that maintains such information, even a Windows application that included the windows extension would not contain such information for any stored file. *See*, Russ Decl., ¶¶ 127-133.

Element 1[b]: cursor display code, said cursor display code operable to modify said cursor image; and

This element is considered in detail below with Element 1[c][iii].

Element 1[c][i]: a first server computer for transmitting specified content information to said remote user terminal,

This element would not have been obvious to a POSITA in view of Malamud because Malamud fails to disclose transmitting specified content information to a remote user terminal. This claim element highlights the need for a server to be

connected to the user terminal to provide the specified content information, such as a webpage, to a user terminal so that the browser can render the webpage and follow the cursor display instructions (in the HTML file for the webpage) to transform the initial cursor image displayed on the user's terminal into the shape and appearance of the specific image. Russ Decl., ¶ 135.

The Original Petitioner and the Patent Owner stipulated that the term “specified content information” / “content information” means “information provided to a user's terminal for use in the display, such as a web page.” EX. 1008. This construction is consistent with the Common Specification, which is directed to serving online content to be displayed, such as a webpage. *See, e.g.*, EX. 1001 at Background of the Invention (discussing the problems with on-line advertising being addressed by the invention); 3:15-21 (the invention is directed to “providing a system for delivering advertising elements online”); Figs. 7-9 (showing the webpages with an initial cursor image and transformed specific cursor image); 5:58-65 (discussing the browser and that the “server 12 functions as a so called ‘web site’”) 6:7-18 (explaining browsers working dynamically at “run-time” versus statically at “compile-time”); 8:38-11:60 (discussing how the browser is used to transform the cursor image on the webpage using plug-ins such as Active X). Russ Decl., ¶¶ 136-137, 139.

A POSITA would not consider a Windows application program as “information for use in the display, such as a webpage.” Rather, the POSITA would see it as information to be installed because a Windows application program cannot run to invoke Malamud’s window procedure without first being installed. An application program is not like a webpage or other online content that a browser can dynamically render, and it is therefore unsuitable to the purpose of serving webpages with promotional cursors. Russ Decl., ¶¶ 101, 138. Also, downloading the Windows application program described in Malamud would not result in the downloading of either the specific image (information cursor content) or the information for display (icon) that triggers that transformation of initial cursor image to the corresponding specific image because the associated icon and metadata for a file stored at the computer are created at the time the file is stored and already managed by the Windows 3.1 operating system. Consequently, downloading Malamud’s application program from a server would not comprise a request to a server to provide specified content information to a user terminal. Russ Decl., ¶¶ 102, 140

Dr. Rosenberg states: “[t]he application program comprises ‘specified content information’ because it includes content to be displayed on the screen when the program is run, and it includes instructions regarding the modified cursor, such as whether to display an information cursor, the type of information cursor to display,

and an indication of the content of the information cursor.” EX. 1003-1, ¶ 96. His explanation for this statement is that the book icon 32 relates to the application running in that window because there is a discussion in the specification that windows displayed on the screen are generated by various programs. But this conclusion is inconsistent with the totality of Malamud’s specification because the Windows application program only invokes the window procedure and there is no description that it passes the icon to the latter. Because Dr. Rosenberg’s analysis of how the Windows application program interacts with Malamud’s window procedure is incorrect, his conclusion that it passes icon information or file metadata is flawed. Because his obviousness analysis is also premised on this faulty conclusion, his conclusion that all a POSITA must do to achieve the claimed method is to serve the Windows application program from a server is incorrect. Russ Decl., ¶ 103.

Downloading Malamud from a server would therefore not meet this element. A POSITA in 1997 would also not have been motivated download a Windows application program from a server, because (a) there would be no need to introduce a server just to download one Windows application program, (b) Windows application programs are large in size in comparison to webpages, and (c) internet bandwidth in 1997 was both limited and expensive. Russ Decl., ¶¶ 104, 140-141.

For all of the above reasons, Malamud fails to disclose this element.

Element 1[c][ii]: said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image,

Malamud neither discloses nor renders obvious this element. This claim element captures the teachings of the Common Specification that in order to effectuate the cursor modification in a webpage rendered by a browser, some additional cursor display code such as ActiveX or JavaScript is necessary to effectuate the desired modification. The cursor display instruction must indicate cursor image data whereby the cursor display instruction and code are able to cause the use terminal to display a modified cursor image on the user's display in the shape and appearance of the specific image. Russ Decl., ¶ 143.

There is no description in Malamud of the Windows application program communicating to the windows procedure the any specific image to display in the information cursor or any instruction that would affect the shape or appearance of the specific image. Rather, all that is disclosed is that the Windows application can set an on/off toggle whereby either the information cursor's information box is always displayed and is blank when the cursor is not over the icon, or that the information cursor box is only displayed with the metadata contained in it (*i.e.*, not blank) when the cursor is over the icon. But these options do not change the shape

or appearance of the specific image because same the specific image is always displayed regardless of which setting the Windows application program set for Malamud's window procedure. Russ Decl., ¶ 144.

While Malamud does disclose that the window procedure can communicate the bitmap of an image to be displayed in the information box of the information cursor, as well as the bitmap of the information cursor type to be displayed for a stored file, Malamud describes not state the source either bitmap. Nor does it have to because Malamud describes that the operating system already has both bitmaps. EX. 1004 at 5:16-18 ("The appearance of the cursor on the video display 20 (FIG. 1) is dictated by bitmaps stored within the operating system 22."); 5: 59-62 ("The message includes a pointer to a bitmap of graphical information that the operating system 22 should use in the preview portion.") This description is consistent with Malamud's overall disclosure that it is directed to a window procedure that is part of the operating system and therefore already has access to the metadata for stored files to display in the information cursor. Russ Decl., ¶ 145.

Element 1[c][iii]: wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information further comprises information to be displayed on said display of said user's terminal.

Element 1[c][iv]: said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image in response to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal, and wherein said specific image relates to at least a portion of said information to be displayed on said display of said remote user's terminal.

Element 1[c][iii] and [iv] are not disclosed by and would not have been obvious to a POSITA in view of Malamud. These claim elements serve to highlight the that a server needs to provide the specified HTML webpage file (specified content information) to the user terminal so that the browser on the user terminal can display the webpage content (information to be displayed) on the user terminal wherein the movement of the cursor over that particular information to be displayed will trigger the cursor modification into the shape and appearance of the corresponding specific image. Malamud's application program does not generally comprise "specified content information" for the reasons previously stated. Therefore, even if a Windows application program were served from a server, it would not meet this claim element by virtue this fact alone. Russ Decl., ¶ 146.

The Petitioner reads the "information be displayed" element onto the icon 32 for file stored in the computer system, But Malamud does not describe the application program as having the book icon 32 for the stored file, nor providing it

to the window procedure. Nor does Malamud describe the window procedure providing or identifying the icon for display. Nor would there be any need for the Windows application program to do so because Malamud shows that the icon is pre-existing and is already displayed before the window procedure has begun actively monitoring the cursor position. Russ Decl., ¶ 147.

Importantly, Dr. Rosenberg does not express any opinion that Malamud’s application program provides the icon of the stored file and discloses this entire element. This alone is fatal to his analysis. *See* EX. 1003-1 at ¶ 102 (only testifying that “that the application program would contain cursor display instructions that instruct the performance of the functionality discussed below.”); ¶ 104 (only testifying that “the application program would include information to be displayed on the display of the user’s terminal.”); ¶ 105 (only citing to graphical image displayed in the preview cursor--- not the icon.); ¶ 108 (again not reading the “information to be displayed” element onto the icon) Russ Decl., ¶ 148; *see also* Russ Decl., ¶¶ 111, 114.

But the claim requires both the “specific image” element and the “information to be displayed” element, wherein the latter is what the initial cursor image must move over in order to trigger the transformation of the initial cursor image into the shape and appearance of the specific image. Because the Petitioner cannot establish

that the application program would send both the icon and the image to be displayed in the information cursor, this alone is fatal to this ground. Russ Decl., ¶ 149.

The Petitioner’s sole reliance for the proposition that Malamud’s application program displays the icon is a general statement that when a Windows operating system opens a Windows application program, it creates a new window and assigns a window procedure for that window. *See*, EX. 1004 at 5:53-59 (explaining the operating system “logically divides the user interface into a number of windows” wherein “each window has a separate window procedure associated with it” and “a message queue for each program that generates windows.”) But this passage does not describe or state that the Windows application program supplies the icon or create the content displayed in the window and such a reading is contrary to Malamud’s specification. Malamud describes a new window procedure that is part of the Windows 3.1 operating system, and not the window program. Moreover, a POSITA would understand that an application program does not create or contain all the content in a window that it generates. Rather, a POSITA understands that the content within a generated window may be supplied by the application program or by the operating system. For example, icons and metadata for windows functions or stored files are supplied by the operating system. The sole passage relied upon by Petitioner for this element therefore provides no express or inherent support for the

proposition that the icon was generated by the application program. Russ Decl., ¶ 150.

Malamud does not teach these limitations, even if the Windows application program was downloaded from a server and was designed to invoke the proposed Malamud extension. The metadata and icon associated with a stored file already exist and are managed by the operating system. The Windows application that takes advantage of Malamud's proposed extension to Windows 3.1 only invokes the extension and sets which mode of information cursor display it wants. The Windows application does not contain the extension. As a POSITA would understand, the extension is part of the operating system, and not the program. However, if the extension was somehow part of the program, it would not contain metadata and icons of any files stored on the computer. And a request for such a program would not be a request for specified content information, as previously discussed. Indeed, this would be so even if the Windows application contained the file, the file's icon, the information cursor bitmap, and the data to be included in the information portion of the information cursor because it would still not operate like the integrated server system for displaying content and a POSITA would not consider the request for such an application to be a request for the claimed "specified content information." Russ Decl., ¶ 151.

Thus, Claim 1 would not have been obvious in view of Malamud for at least the reasons set forth herein.

Element 53[Preamble]: A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

Element 53[Preamble] is not rendered obvious in view of Malamud because Malamud does not disclose or suggest to a POSITA a method for modifying a cursor image displayed on a computer connected to a server for the reasons discussed above regarding Claim 1, particularly Element 1[Preamble]. Russ Decl., ¶¶ 155-155.

Element 53[a]: receiving a request at said at least one server to provide specified content information to said user terminal;

Element 53[a] is not rendered obvious in view of Malamud, because Malamud it does not disclose or suggest to a POSITA requesting a server to provide specified content information to the user's terminal, for the reasons discussed above in connection with Claim 1, particularly element 1[a]. Russ Decl., ¶ 156.

Element 53[b]: providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

Element 53[b] is not rendered obvious in view of Malamud, because Malamud does not disclose or suggest to a POSITA providing specified content information to said user terminal in response to a request, where the specified content information

includes at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image, for the reasons provided above in connection with Claim 1, particularly element 1[b]. Russ Decl., ¶ 157.

Element 53[c][i]: transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information that is to be displayed on said display of said user's terminal, and

Element 53[c][ii]: wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image in response to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal, and

Element 53[c][iii]: wherein said specific image has a shape and appearance relating to said information to be displayed.

Element 53[c][i]-[iii] is not rendered obvious in view of Malamud, because Malamud does not disclose or suggest to a POSITA transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information that is to be displayed on said

display of said user's terminal, wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image in response to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal, and wherein said specific image has a shape and appearance relating to said information to be displayed for the reasons provided above in connection with Claim 1, particularly Element 1[c][i]-[iv]. Russ Decl., ¶ 158.

C. Ground 2: The Combination of Malamud and Nakagawa Do Not Render Obvious Claims 1 or 53

With respect to **Ground 2**, Petitioner asserts that Claims 1, 27, and 53 are unpatentable as allegedly being obvious over the combination of Malamud and Nakagawa, and the Board found that Petitioner had established a reasonable likelihood it would prevail in showing that Claims 1 and 53 are unpatentable over Malamud and Nakagawa. See Paper 12. As a reminder, Claim 27 was previously canceled before the filing of the Petition.

1. Overview of Nakagawa (EX. 1005)

Nakagawa is not directed to online advertising, the requesting and service of webpages, or the modification of cursors or the display of online content. It is

directed to a system for automatically requesting and distributing application programs and updates thereto to be installed on client computers so that the downloaded software is properly maintained. As Nakagawa describes its system and method allows for “distributing and maintaining software...where a number of various software users and a number of vendors can systematically manage the distribution and maintenance of plural sets of various object software.” *See*, EX. 1005 at 5:5-10; *see also* Russ Decl., ¶¶ 159-161.

There are no shortcomings with Malamud that are fixed via the addition of a downloading a Windows application program from Nakagawa’s server for the reasons previously discussed with regard to Ground 1, namely:

- a. It does not result in a server system for display in which there is a request from the user’s computer for “information to be used in the display, such as a webpage (“specified content information” that the server can send to effectuate the claimed method/system.
- b. A Windows application program does not include metadata invokes the extension include “information to be displayed” (putatively, the stored file’s associated icon) over which the cursor has to be moved in order to trigger the transformation of the initial cursor image into the shape and appearance of a

corresponding “specific image” (putatively, the stored file’s associated metadata displayed in the information portion of the information cursor). The operating system maintains and displays the icons associated with the stored files.

- c. A POSITA would not be motivated to develop a Windows application program that would communicate a file icon to Malamud’s proposed window procedure because it would involve a change to the design of the Windows 3.1 operating system, which is not feasible to a POSITA with predictability or undue experimentation, if at all.

Russ Decl., ¶¶ 163-165.

2. Petitioner Fails to Establish Obviousness of Claims 1 and 53 using Malamud and Nakagawa

As mentioned above, there are no shortcomings with Malamud that are fixed via the addition of downloading a Windows application program from Nakagawa’s server for the reasons previously discussed with regards to Ground 1. Additionally, it would not have been obvious to a POSITA in 1997 to try to download such a Windows application from a server per Nakagawa because of bandwidth and cost issues and because doing so still would not result in practice of Claim 1 or Claim 53

for the reasons explained in Ground 1. Because Nakagawa does not correct any of the previously identified failings of Malamud, the combination of Malamud and Nakagawa fails to establish obviousness of Claims 1 and 53. Russ Decl., ¶¶ 159-199.

D. Ground 3: Nielsen and Malamud Do Not Render Obvious Claims 1, 38, or 53.

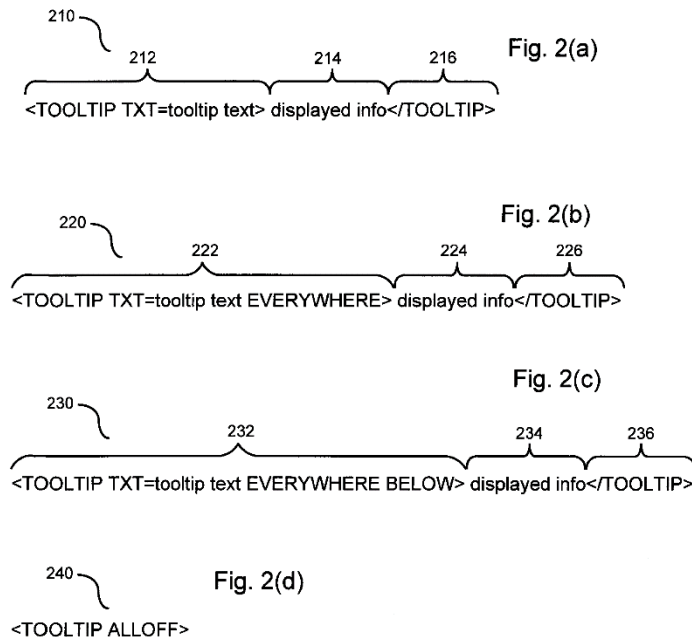
With respect to **Ground 3**, Petitioner asserts that Claims 1, 27, 38, and 53 are unpatentable as allegedly being obvious over the combination of Nielsen and Malamud. The Board did not determine whether a likelihood of success had been shown for this ground, but provided its initial views for the benefit of the parties and made its decision to institute based upon Grounds 1 and 2. Paper 12 at 9-10. As a reminder, Claim 27 was previously canceled before the filing of the Petition.

1. Overview of Nielsen (EX. 1006)

Nielsen is directed to a specific extension of the HTML standard that added a feature called “tooltips” to the standard. A webpage developer can place into an HTML file, adjacent to any webpage content they wish to display, a tooltip tag and enter content they wish to be displayed in a tooltip box when a cursor is positioned over particular webpage content for a specified period of time. Nielsen at 3:34-45.

Because Nielsen’s specification is directed to explaining this new feature of HTML, and its use requires proper use of HTML language, a great deal of Nielsen’s

specification is directed to the various valid HTML formats for tooltip tags, as shown in FIGS. 2(a)-2(d):



As shown, the webpage developer can insert into the webpage's HTML file a pair of tooltip tags around the portion of webpage text for which a tooltip is desired to be displayed, specify the tooltip text, and specify whether this tooltip text should be displayed over every instance of the webpage text.

In a primary example given, shown in FIG. 4, the tooltip text "London Heathrow" will appear over the webpage text "LHR" when a cursor has been positioned somewhere over "LHR" for a sufficient length of time. Likewise, the

tooltip text “United Airlines” will appear above the webpage text “UA” and the tooltip text “San Francisco” will appear over the webpage text “SFO.”

Because browsers that support HTML include code to execute these HTML instructions, the webpage developer does not have to provide code to the browser to make a tooltip appear on the user’s display, but only supplies data surrounded by tags. See also, Russ Decl., ¶¶ 200-203.

2. Petitioner Fails to Establish Obviousness of Claims 1, 38, or 53 using Nielsen and Malamud

Malamud is deficient for all the reasons previously identified, and unless otherwise stated Nielsen does not correct the deficiencies of Malamud for any element of Claim 1, 38, or 53.

Motivation to Combine

As an initial matter, the Petition has failed to show a motivation to combine Nielsen and Malamud. Nielsen’s method is directed to serving and displaying webpage but is not directed to modifying a cursor image on the served webpage. Rather, it describes an addition to the HTML protocol to display a tooltip. Malamud is directed to an addition to the Windows 3.1 (and similar) operating systems in which a Windows application program can invoke a new windows procedure that allows metadata for a stored file to be displayed in an information cursor rather than

in a status bar. Malamud is not directed to serving a webpage or other online content for display. *See*, Russ Decl., ¶¶ 204-213.

A POSITA would not have been motivated to modify Nielsen to display a tooltip in the information box an information cursor. First, a POSITA would have no need to substitute the tooltip with an information cursor “to ensure that the user sees the supplemental information or graphic” because, as Nielsen explains, the tooltip is already automatically displayed placed adjacent to the associated webpage content. EX. 1006 at 7:16-32. *See also*, Russ Decl., ¶¶ 217, 226. A POSITA would also not have been motivated to make such a change because it would also display the tooltip whether the user wished to see it or not. *Id.*

Second, a POSITA would not have been motivated to try to make a change to the HTML protocol. The HTML protocol is controlled by the Internet Engineering Task Force (the “IETF”), the committee that decides what changes will be implemented to the HTML protocol. A POSITA would have to get the IETF to approve any proposed change to the HTML protocol. Then the IETF would have to design new HTML code to implement the proposed change. Then browsers around the world would have to be updated to accomplish the change. Because a POSITA would know that they would not have any predictable success to accomplish the

change with any reasonable amount of effort, it would likely never even cross their mind to try to modify Nielsen. Russ Decl., ¶¶ 207, 218-219.

It would be implausible for a POSITA to combine the event-driven aspects of Malamud's application with the HTML-tag system disclosed in Nielsen. Any attempt at doing so would not have had a reasonable chance of success and would have required undue experimentation (namely, re-writing either Malamud in HTML or Nielsen into an event-driven system). The details of how this combination would function are not explained by Dr. Rosenberg, who merely concludes it would be obvious to extend Nielsen's tooltip features to include modified cursors and graphical images. Nielsen's HTML-based tooltips do not contain the grammar or syntax to implement event-driven programming. In contrast, the '102 Patent teaches using JavaScript to implement the event driver portion of the claims, which provided a POSITA in 1997 with guidance of how to build a webpage that could have had instructions sent via server that would implement the event-driven portions of the claims. The concept of augmenting the HTML standard to include event driven aspects is not a motivation a POSITA would have had back in 1997, as a POSITA would not need to know event-driven programming in order to implement Nielsen's tooltip tags. Malamud relies on Windows for events driven programming. Neither references how to implement events-driven programming on a webpage, and

Petitioner does not account for this failure in the teachings in combining Nielsen with Malamud. Thus, it would have not been obvious for a POSITA to modify the teachings of Nielsen in order to display a tooltip in an information cursor. Russ Decl., ¶¶ 212-228.

Element 1[Preamble]: A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

Element 1[Preamble] would not have been obvious to a POSITA in view of Nielsen and Malamud for at least the same reasons as explained regarding Ground 1. *See* Section V[B], re Element 1[Preamble]. *See also*, Russ Decl., ¶¶ 235-241.

Element 1[a]: cursor image data corresponding to said specific image;

Element 1[a] would not have been obvious to a POSITA in view of Nielsen and Malamud because the tooltip metadata does not comprise cursor image data, a POSITA would not have been motivated to make display a tooltip in an information cursor because a tooltip was already displayed near the cursor and displaying the tooltip in an information cursor would have created a problem of tooltips being inadvertently displayed, because such a change would have required modifying the HTML protocol or using techniques that the Petitioner has not even alleged to be

obvious to a POSITA in 1997 without the extensive teachings of the Common Specification. *See* Russ Decl., ¶¶ 242-243.

Element 1[b]: cursor display code, said cursor display code operable to modify said cursor image; and

Element 1[b] would not have been obvious to a POSITA in view of Nielsen and Malamud. Nielsen implements no cursor display code whatsoever. Malamud, to the extent that it teaches cursor display code at all, teaches it as an undisclosed part of a Windows operating system and does not discuss how the cursor display code operates. A POSITA would not know how to implement the claimed cursor display code. *See also* Russ Decl., ¶¶ 244-244.

In order to combine these references, a POSITA would have to eliminate Nielsen's requirement for the user to hover their cursor over the associated webpage content. A POSITA would not wish to make this change because it would lead to tooltips being displayed when the user did not want them being displayed, and also being often displayed in the wrong place since the tooltip's location would move rather being statically displayed adjacent to the associated webpage content. A POSITA would have no reason to introduce such confusion and would not think of displaying a tooltip in the body of an information cursor. In contrast, allowing the initial cursor to be modified to the specific image in response to movement over the associated webpage content is desired in the claimed inventions because the goal of

the claimed inventions disclosed in the Common Specification is to serve a webpage with browser instructions that can be implemented in real-time to modify the cursor image to a specific image used on the webpage to promote the sale of the product being advertised or offered on the webpage. *Id.*

Element 1[c][i]: a first server computer for transmitting specified content information to said remote user terminal,

Element 1[c][i] is not disclosed in Nielsen or Malamud or rendered obvious by the combination, because, at a minimum, the claimed specified content information must include the cursor image data and cursor display instructions (recited in further elements) that are not present anywhere in Nielsen. Nielsen does not teach the use of cursor display code or any cursor modification as a result of the use of cursor display instructions, cursor display code and cursor image data as the claim requires. Therefore, it does not transmit the claimed specified content information. *See*, Russ Decl., ¶¶ 246-248.

As previously stated, Malamud does not disclose a request to a server for specified content information nor the transmission back of specified content information. As Nielsen does not correct this failure of Malamud, the combination also fails to disclose this element. Moreover, it was not within the skill of a POSITA to combine Windows operating system-based technology with and implement in it

a browser- based environment in 1997, and a POSITA would not have been motivated to do so for the reasons previously stated. See Russ Decl., ¶¶ 249-251.

Element 1[c][ii]: said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image,

Element 1[c][ii] is not disclosed in Nielsen because, at a minimum, the claimed specified content information must include the cursor image data and cursor display instructions that are not present anywhere in Nielsen. Nielsen does not teach the use of cursor display code or any cursor modification. Therefore, it does not transmit the claimed specified content information. Malamud is deficient for the reasons previously stated with regards to Ground 1. As Nielsen does not correct this failure of Malamud, the combination also fails to disclose this element. See Russ Decl., ¶¶ 252-257.

Element 1[c][iii]: wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information further comprises information to be displayed on said display of said user's terminal.

Element 1[c][iii] is not disclosed in Nielsen because, at a minimum, the claimed specified content information must include the cursor image data and cursor display instructions that are not present anywhere in Nielsen. Nielsen does not teach

the use of cursor display code or any cursor modification as a result of the use of cursor display instructions, cursor display code and cursor image data as the claim requires. Therefore, it does not transmit the claimed specified content information. Malamud is deficient for the reasons previously stated with regards to Ground 1. As Nielsen does not correct this failure of Malamud, the combination also fails to disclose this element. *See* Russ Decl., ¶¶ 258-263.

Element 1[c][iv]: said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image in response to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal, and wherein said specific image relates to at least a portion of said information to be displayed on said display of said remote user's terminal.

Element 1[c][iv] is not disclosed in Nielsen because, at a minimum, the claimed specified content information must include the cursor image data and cursor display instructions that are not present anywhere in Nielsen. Nielsen does not teach the use of cursor display code or any cursor modification as a result of the use of cursor display instructions, cursor display code to modify the cursor into the shape and appearance of said specific image in response to movement of said cursor image over a display of said at least a portion of said information to be displayed on said

display of said user's terminal. Malamud is deficient for the reasons previously stated with regards to Ground 1. As Nielsen does not correct this failure of Malamud, the combination also fails to disclose this element. *See Russ Decl.*, ¶¶ 264-269.

Element 53[Preamble]: A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

Element 53[Preamble] would not have been obvious to a POSITA in view of Nielsen and Malamud for at least the same reasons set forth in the discussion of Element 1[Preamble] and Ground 1 with regard to Malamud's lack of disclosure of Claim 53.

Element 53[a]: receiving a request at said at least one server to provide specified content information to said user terminal;

Element 53[a] would not have been obvious to a POSITA in view of Nielsen and Malamud for at least the same reasons as set forth in the discussion of Element 1[a] and Ground 1 with regard to Malamud's lack of disclosure of Claim 53.

Element 53[b]: providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

Element 53[b] would not have been obvious to a POSITA in view of Nielsen and Malamud for the same reasons as set forth in the discussion of Element 1[b] and Ground 1 with regard to Malamud's lack of disclosure of Claim 53.

Element 53[c][i]: transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information that is to be displayed on said display of said user's terminal, and

Element 53[c][ii]: wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image in response to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal, and

Element 53[c][iii]: wherein said specific image has a shape and appearance relating to said information to be displayed.

Element 53[c][i]-[iii] would not have been obvious to a POSITA in view of Nielsen and Malamud for the same reasons set forth in the discussion of Elements 1[c][i]-[iv] and Ground 1 with regard to Malamud's lack of disclosure of Claim 53.

Claim 27

Claim 27 was previously canceled before the Petition was filed and is only addressed herein because Claim 38, which depends therefrom, is a Challenged Claim. For the reasons set forth above regarding Claims 1 and 53, Claim 27 would not have been obvious to a POSITA in view of Malamud and Nielsen (because Nielsen cures no deficiency in Malamud and vice versa.)

Claim 38: The server system in accordance with Claim 27, wherein said specified content information is transmitted in the form of HTML files that define a web page.

Nielsen teaches an HTML file, but not with the claimed specified content information for all the reasons set forth regarding Claim 1 and 53 because it does not disclose any “cursor display instructions,” “cursor image data”, “cursor display code,” or even the claimed “information to be displayed” (because no initial cursor image displayed on the user terminal is transformed responsive to its movement over any served webpage content in Nielsen) and therefore does not disclose any request for “specified content information” or the server having responded with “specified content information.” Therefore, Claim 38 is not obvious in view of the combination of Nielsen and Malamud for all the same reasons as explained with regard to Claims 1 and 53.

In conclusion, the combination of Nielsen and Malamud does not render any of the challenged claims obvious.

VI. CONCLUSION

For these reasons, Patent Owner respectfully requests that the Board find that Petitioner has not shown that the challenged claims are unpatentable and confirm the validity of these claims.

Dated: April 17, 2024

Respectfully submitted,

By /s/ Sandeep Seth
Sandeep Seth (Reg No. 37,537)
SethLaw
Two Allen Center
1200 Smith Street, Ste. 1600
Houston, TX 77002
Telephone: (713) 244-5017
Facsimile: (713) 244-5018
Email: ss@sethlaw.com

CERTIFICATE OF COMPLIANCE

Patent Owner certifies that Patent Owner's Response contains 13,992 words exclusive of a table of contents, a table of authorities, mandatory notices under § 42.8, a certificate of service or word count, or appendix of exhibits or claim listing, according to Microsoft Word.

Dated: April 17, 2024

Respectfully submitted,

/s/ Sandeep Seth
Sandeep Seth (Reg No. 37,537)
SethLaw
ss@sethlaw.com

§ 42.6(e)—CERTIFICATION OF SERVICE

In accordance with § 42.6(e)(1), the undersigned certifies that on the 6th day of March 2024, the above PATENT OWNER RESPONSE was served, via electronic mail upon the following counsel for Patent Owner,

- Heath J. Briggs - BriggsH@gtlaw.com
- Joshua L. Raskin - Joshua.Raskin@gtlaw.com
- Brian J. Prew - prewb@gtlaw.com
- Vimal M. Kapadia - kapadiav@gtlaw.com
- Kathryn E. Albanese - Katie.Albanese@gtlaw.com

Dated: April 9, 2024

Respectfully submitted,

/s/ Sandeep Seth
Sandeep Seth (Reg No. 37,537)
SethLaw
ss@sethlaw.com