

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

RALPH LAUREN CORPORATION,

Petitioner,

v.

LEXOS MEDIA IP, LLC,

Patent Owner.

PTAB Case No. IPR2018-01755

Patent No. 6,118,449

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 6,118,449**

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
P.O. Box 1450
Alexandria, VA 22313-1450

September 18, 2018

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(B)	1
A. Real Party in Interest	1
B. Related Matters	2
C. Notice of Counsel and Service Information	3
III. REQUIREMENTS FOR INTER PARTES REVIEW	3
A. Standing	4
B. Identification of Challenges	4
1. Claims Challenged	4
2. The Prior Art	4
3. Fee for Inter Partes Review (§ 42.15(a))	4
4. Supporting Evidence	5
5. Statutory Grounds	5
6. How Claims Are Unpatentable	5
IV. OVERVIEW OF THE '449 PATENT	5
A. Priority Date	5
B. State of the Art Before the '449 Patent	6
C. Summary of the '449 Patent	8
D. Level of Ordinary Skill in the Art	10
V. PROPOSED CLAIM CONSTRUCTIONS	11
A. “cursor image” / “initial cursor image”	12
B. “specific image” / “specific cursor image”	13
C. “modifying a cursor image” / “modified cursor image”	14
D. “cursor image data”	15

TABLE OF CONTENTS

(continued)

	Page
E. “cursor display code”	16
F. “cursor display instruction”	17
VI. THERE IS A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE ’449 PATENT IS UNPATENTABLE	18
A. Identification and Overview of Key Prior Art References	18
1. Malamud (Ex. 1004)	19
2. Anthias (Ex. 1005)	21
3. Nielsen (Ex. 1006)	23
4. Baker (Ex. 1007)	23
B. Motivation to Combine References	26
C. Ground 1: Claims 1, 7, 15, 27, 33, 41, 53-54, 63, 72-73, 82 Are Rendered Obvious by Malamud and Anthias	30
1. Claim 1 Is Rendered Obvious by Malamud and Anthias	30
2. Claim 7 Is Rendered Obvious by Malamud and Anthias	38
3. Claim 15 Is Rendered Obvious by Malamud and Anthias	38
4. Claim 27 Is Rendered Obvious by Malamud and Anthias	39
5. Claim 33 Is Rendered Obvious by Malamud and Anthias	39
6. Claims 41 and 82 Are Rendered Obvious by Malamud and Anthias	39
7. Claim 53 Is Rendered Obvious by Malamud and Anthias	39
8. Claim 54 Is Rendered Obvious by Malamud and Anthias	44
9. Claim 63 Is Rendered Obvious by Malamud and Anthias	45
10. Claim 72 Is Rendered Obvious by Malamud and Anthias	46
11. Claim 73 Is Rendered Obvious by Malamud and Anthias	46

TABLE OF CONTENTS
(continued)

	Page
D. Ground 2: Claims 12, 14, 38, 40, 60, 62, 79, and 81 Are Rendered Obvious by Malamud, Anthias, and Nielsen	46
1. Claim 12 Is Rendered Obvious by Malamud, Anthias, and Nielsen.....	46
2. Claim 14 Is Rendered Obvious by Malamud, Anthias, and Nielsen.....	47
3. Claims 40, 62, and 81 Are Rendered Obvious by Malamud, Anthias, and Nielsen.....	48
4. Claims 38, 60, and 79 Are Rendered Obvious by Malamud, Anthias, and Nielsen.....	48
E. Ground 3: Claims 2-3, 5-6, 28-29, 31-32, 55-56, 58-59, 74-75, and 77-78 Are Rendered Obvious by Malamud, Anthias, and Baker.....	48
1. Claim 2 Is Rendered Obvious by Malamud, Anthias, and Baker	48
2. Claim 3 Is Rendered Obvious by Malamud, Anthias, and Baker	49
3. Claim 5 Is Rendered Obvious by Malamud, Anthias, and Baker	50
4. Claim 6 Is Rendered Obvious by Malamud, Anthias, and Baker	51
5. Claims 28, 55, and 74 Are Rendered Obvious by Malamud, Anthias, and Baker	51
6. Claim 29, 56, and 75 Are Rendered Obvious by Malamud, Anthias, and Baker	51
7. Claims 31, 58, and 77 Are Rendered Obvious by Malamud, Anthias, and Baker	52

TABLE OF CONTENTS
(continued)

	Page
8. Claims 32, 59, and 78 Are Rendered Obvious by Malamud, Anthias, and Baker	52
F. Ground 4: Claims 13, 39, 61, and 80 Are Rendered Obvious by Malamud, Anthias, Nielsen, and Baker	52
1. Claim 13 Is Rendered Obvious by Malamud, Anthias, Nielsen, and Baker	52
2. Claims 39, 61, and 80 Are Rendered Obvious by Malamud, Anthias, Nielsen, and Baker	53
G. Ground 5: Claims 1, 7, 15, 27, 33, 41, 53-54, 63, 72-73, 82 Are Rendered Obvious by Baker and Anthias	53
1. Claim 1 Is Rendered Obvious by Baker and Anthias	53
2. Claim 7 Is Rendered Obvious by Baker and Anthias	61
3. Claim 15 Is Rendered Obvious by Baker and Anthias	61
4. Claim 27 Is Rendered Obvious by Baker and Anthias	62
5. Claim 33 Is Rendered Obvious by Baker and Anthias	62
6. Claims 41 and 82 Are Rendered Obvious by Baker and Anthias	62
7. Claim 53 Is Rendered Obvious by Baker and Anthias	62
8. Claim 54 Is Rendered Obvious by Baker and Anthias	64
9. Claim 63 Is Rendered Obvious by Baker and Anthias	64
10. Claim 72 Is Rendered Obvious by Baker and Anthias	65
11. Claim 73 Is Rendered Obvious by Baker and Anthias	65
H. Ground 6: Claims 12, 14, 38, 40, 60, 62, 79, and 81 Are Rendered Obvious by Baker, Anthias, and Nielsen	65
1. Claim 12 Is Rendered Obvious by Baker, Anthias, and Nielsen	65

TABLE OF CONTENTS
(continued)

	Page
2. Claim 14 Is Rendered Obvious by Baker, Anthias, and Nielsen.....	65
3. Claims 38, 60, and 79 Are Rendered Obvious by Baker, Anthias, and Nielsen	66
4. Claims 40, 62, and 81 Are Rendered Obvious by Baker, Anthias, and Nielsen	66
VII. CONCLUSION.....	66
CERTIFICATE OF WORD COUNT	68
CERTIFICATE OF SERVICE	69

TABLE OF AUTHORITIES

PAGE(S)

CASES

<i>Black & Decker, Inc. v. Positec USA, Inc.</i> , 646 Fed. App'x 1019 (Fed. Cir. 2016)	11
<i>CCS Fitness, Inc. v. Brunswick Corp.</i> , 288 F.3d 1359 (Fed. Cir. 2002)	12
<i>In re Rambus Inc.</i> , 694 F.3d 42 (Fed. Cir. 2012)	11
<i>KSR Int'l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	26, 27, 29
<i>Lexos Media IP, LLC v. APMEX, Inc.</i> , No. 2:16-cv-00747-JRG-RSP, Early Claim Construction Opinion and Order, Dkt. 86 (E.D. Tex., Mar. 16, 2017) (Ex. 1010).....	2, 3
<i>Lexos Media IP, LLC v. Ralph Lauren Corporation et al.</i> , No. 1:17-cv-01319-LPS (D. Del.)	2
<i>Perfect Web Techs., Inc. v. InfoUSA, Inc.</i> , 587 F.3d 1324 (Fed. Cir. 2009)	27
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	11, 12

STATUTES

35 U.S.C. § 102	5, 19, 22, 23
35 U.S.C. § 103	1, 5, 18, 26
35 U.S.C. § 311	1, 5
35 U.S.C. § 313	5
35 U.S.C. § 314	5
35 U.S.C. § 315	4, 5

35 U.S.C. § 316	5
35 U.S.C. § 317	5
35 U.S.C. § 318	5
35 U.S.C. § 319	5
35 U.S.C. § 325	5

OTHER AUTHORITIES

37 C.F.R. § 42.8	1, 3
37 C.F.R. § 42.10	3
37 C.F.R. § 42.15	4
37 C.F.R. § 42.22	4
37 C.F.R. § 42.73	4
37 C.F.R. § 42.100	1
37 C.F.R. § 42.101	4
37 C.F.R. § 42.102	4
37 C.F.R. § 42.104	4, 5
37 C.F.R. § 42.105	4
37 C.F.R. § 42.106	4

EXHIBIT LIST¹

Ex. 1001	U.S. Patent No. 5,995,102 entitled <i>Server System and Method for Modifying a Cursor Image</i> to James Samuel Rosen et al. (“’102 Patent”)
Ex. 1002	U.S. Patent No. 6,118,449 entitled <i>Server System and Method for Modifying a Cursor Image</i> to James Samuel Rosen et al. (“’449 Patent”)
Ex. 1003	Declaration of Dr. Benjamin B. Bederson Regarding the ’102 and ’449 Patents
Ex. 1004	U.S. Patent No. 6,437,800 to Mark A. Malamud (“Malamud”)
Ex. 1005	U.S. Patent No. 5,920,311 to Taf Anthias (“Anthias”)
Ex. 1006	U.S. Patent No. 5,991,781 to Jakob Nielsen (“Nielsen”)
Ex. 1007	U.S. Patent No. 5,715,416 to Michelle Baker (“Baker”)
Ex. 1008	Chart Listing Full Claim Elements for the Claims of the ’102 Patent
Ex. 1009	Chart Listing Full Claim Elements for the Claims of the ’449 Patent
Ex. 1010	<i>Lexos Media IP, LLC v. APMEX, Inc.</i> , No. 2:16-cv-00747-JRG-RSP, Early Claim Construction Opinion and Order, Dkt. 86 (E.D. Tex., Mar. 16, 2017).
Ex. 1011	File History of the ’102 Patent
Ex. 1012	File History of the ’449 Patent

¹ Given the near complete overlap of the documents relied upon in this IPR Petition and those relied upon in the IPR Petition on the related ’102 Patent, Petitioner has included in this list and with this Petition all documents relied upon in both related IPR Petitions so that the Board need only refer to one set of Exhibits.

- Ex. 1013 Invalidity Claim Charts for the '102 Patent
- Ex. 1014 Invalidity Claim Charts for the '449 Patent
- Ex. 1015 Appendices to the Declaration of Dr. Benjamin B. Bederson

I. INTRODUCTION

Pursuant to 35 U.S.C. § 311 and 37 C.F.R. § 42.100, Ralph Lauren Corporation (“Petitioner”) petitions for *inter partes* review (“IPR”) of Claims 1-3, 5-7, 12-15, 27-29, 31-33, 38-41, 53-56, 58-63, 72-75, and 77-82 (“Challenged Claims”) of U.S. Patent No. 6,118,449 (Ex. 1002).

This Petition shows that there is a reasonable likelihood of invalidity based on at least one of the Challenged Claims based on the following grounds and references.

Grounds	Challenged Claims	References
Pre-AIA 35 U.S.C. § 103(a)	1, 7, 15, 27, 33, 41, 53-54, 63, 72-73, 82	<i>Malamud and Anthias</i>
	12, 14, 38, 40, 60, 62, 79, 81	<i>Malamud, Anthias, and Nielsen</i>
	2-3, 5-6, 28-29, 31-32, 55-56, 58-59, 74-75, 77-78	<i>Malamud, Anthias, and Baker</i>
	3, 39, 61, 80	<i>Malamud, Anthias, Nielsen, and Baker</i>
	1, 7, 15, 27, 33, 41, 53-54, 63, 72-73, 82	<i>Baker and Anthias</i>
	12, 14, 38, 40, 60, 62, 79, 81	<i>Baker, Anthias, and Nielsen</i>

II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(B)

A. Real Party in Interest

Petitioner Ralph Lauren Corporation, its subsidiaries Club Monaco

Corporation, Club Monaco US LLC, Ralph Lauren Media LLC, and PRL USA Holdings, Inc. and Adobe Systems Incorporated are real parties in interest.

B. Related Matters

The '449 Patent and U.S. Patent No. 5,995,102 ("the '102 Patent")
5 (collectively, "Lexos Patents") are asserted against Petitioner in *Lexos Media IP, LLC v. Ralph Lauren Corporation et al.*, No. 1:17-cv-01319-LPS (D. Del.) ("RLC").¹ Assignee Lexos Media IP, LLC ("Lexos") served the complaint on Petitioner on September 19, 2017. Petitioner is contemporaneously filing a
10 petition for IPR of Claims 70-73 of the '102 Patent. The '449 Patent is a continuation of the '102 Patent.² On July 10, 2018, Lexos joined Club Monaco Corporation and Club Monaco US LLC to RLC.

Lexos is also asserting the '449 and '102 Patents against Jos. A. Bank Clothiers, Inc. (Case No. 1:17-cv-01317). Additionally, Lexos has asserted the '449 Patent and/or the '102 Patent in seventeen other cases in the District of

¹ Lexos alleges in the District Court that it is the owner of the '449 Patent. Lexos is also recorded as the current assignee of the Patent.

² For consistency and ease of reference for the Board across both related IPR Petitions, all citations to the specification herein will be made to the column and line numbers of the '102 Patent (Ex. 1001).

Delaware and the Eastern District of Texas, all of which are now terminated. In one of these cases in the Eastern District of Texas, the Court construed the single term “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” as “an
5 image representative of at least a portion of the subject or topic being displayed on the screen.” *Lexos Media IP, LLC v. APMEX, Inc.*, No. 2:16-cv-00747-JRG-RSP (“*APMEX*”), Early Claim Construction Opinion and Order, Dkt. 86, at 12-13 (E.D. Tex., Mar. 16, 2017) (Ex. 1010).

C. Notice of Counsel and Service Information

10 Pursuant to 37 C.F.R. §§ 42.8(b)(3), 42.8(b)(4) and 42.10(a), Petitioner appoints **JAMES F. VALENTINE** (Reg. No. 39,053) as its lead counsel and **DANIEL T. SHVODIAN** (Reg No. 42,148) as its back-up counsel. Both are reachable by mail at Perkins Coie LLP, 3150 Porter Drive, Palo Alto, California 94304; by phone at (650) 838-4300; by fax at (650) 838-4350; and at the following
15 email for service and all communications:

RLC-IPR@perkinscoie.com.

Petitioner consents to electronic service and has executed and is concurrently filing a Power of Attorney appointing the above designated counsel.

III. REQUIREMENTS FOR *INTER PARTES* REVIEW

20 This Petition complies with all statutory requirements, including 37 C.F.R.

§§ 42.104, 42.105, and 42.15, and should be accorded a filing date as the date of filing of this Petition pursuant to 37 C.F.R. § 42.106.

A. Standing

Petitioner certifies that the '449 Patent is available for IPR and that

5 Petitioner is not barred or estopped from requesting IPR of the Challenged Claims.

Petitioner has standing, or meets all requirements, to file this Petition under 35

U.S.C. §§ 315(a)(1), 315(b), and 315(e)(1), and 37 C.F.R. §§ 42.73(d)(1), 42.101, and 42.102.

B. Identification of Challenges

10 Pursuant to 37 C.F.R. §§ 42.104(b) and 42.22, the precise relief requested by Petitioner is that the Board institute trial on and cancel the Challenged Claims because they are invalid.

1. Claims Challenged

15 Claims 1-3, 5-7, 12-15, 27-29, 31-33, 38-41, 53-56, 58-63, 72-75, and 77-82 of the '449 Patent are challenged in this Petition.

2. The Prior Art

The prior art references are: Malamud (Ex. 1004), Anthias (Ex. 1005), Nielsen (Ex. 1006), and Baker (Ex. 1007).

3. Fee for *Inter Partes* Review (§ 42.15(a))

20 Petitioner authorizes the Director to charge any fees required by 37 C.F.R. § 42.15(a) and not submitted with the Petition to Deposit Account No. 50-0665,

charge number 088248.0116.

4. Supporting Evidence

The Declaration of Dr. Benjamin B. Bederson (Ex. 1003) and other supporting evidence are identified in the Exhibit List.

5. Statutory Grounds

Pursuant to 37 C.F.R. § 42.104(b)(2), the review of patentability of the Challenged Claims is governed by 35 U.S.C. §§ 102 and 103 in effect before March 16, 2013. Statutory provisions of 35 U.S.C. §§ 311-319 and 325 that took effect on September 16, 2012 govern this IPR.

6. How Claims Are Unpatentable

Pursuant to 37 C.F.R. § 42.104(b)(4), Section VI of this Petition provides an explanation of how the Challenged Claims are unpatentable.

IV. OVERVIEW OF THE '449 PATENT

A. Priority Date

The '449 Patent issued on September 12, 2000 from U.S. Patent Application No. 09/400,038, filed on September 21, 1999. The '449 Patent claims priority to and is a continuation of U.S. Patent Application No. 08/882,580 (filed June 25, 1997; issued as the '102 Patent on November 30, 1999). Additionally, U.S. Patent Nos. 7,111,254 (issued September 19, 2006) and 6,065,057 (issued May 16, 2000), as divisionals, and 7,975,241 (issued July 5, 2011), as a continuation, claim priority to the application leading up to the '102 Patent.

B. State of the Art Before the '449 Patent

A graphical user interface (“GUI”), first demonstrated by Dougless Engelbart in 1968, is where interactions between humans and computers occur. (Ex. 1003 ¶ 19.) A well-known GUI uses the “desktop metaphor,” where a user
5 uses a pointing device (e.g., mouse) to interact with icons representing computer files and applications on a virtual desktop. (*Id.*)

The operating system (“OS”) is system software that manages computer hardware and software resources, including pointing devices, and display devices and the graphics displayed on them. (*Id.* ¶ 20.) Specifically, the OS uses “driver”
10 programs dedicated to control devices attached to the computer. A “display driver” accepts commands from the OS and generates signals to the display device to render text or images onscreen. Display drivers were only one way that OSs controlled system displays. OSs also used internal functions and other applications to display images. (*Id.*)

15 When a user moves the mouse, an image called a “cursor” moves correspondingly onscreen. (*Id.* ¶ 21.) A cursor’s image file is the actual image drawn by the OS’s chosen display function or application to visually indicate the cursor’s position on the screen. (*Id.*) A single pixel in the cursor, called the “hotspot,” marks the screen location that a mouse click would affect. While
20 applications other than the OS may affect the cursor’s appearance, all images,

including the cursor, are rendered, ultimately, by the OS's chosen display function or application. (*Id.*) An application may trigger modification of graphical elements like fonts and cursors' image files by sending a message to the OS, including specific parameters related to the application's request. (*Id.*)

5 While OSs have for decades provided standard images for cursors, they also allowed applications to customize cursors' appearances. (*Id.* ¶ 23.) Because the user's attention was often focused on or near the cursor onscreen, designers placed additional information around the cursor. (*Id.* ¶ 25.) U.S. Patent No. 5,754,176 (filed October 2, 1995) describes the standard "tooltip" systems built into both
10 Macintosh and Windows 95 to show contextualized help information when a user hovered the cursor in place onscreen. (*Id.* ¶ 26.)

 The client/server architecture is fundamental to computing, and downloading content from servers, including graphical information, has been well-known for decades. (*Id.* ¶ 27.) Client/Server systems are at the heart of the World Wide Web
15 where web browsers allow users (i.e., "clients") to download webpages with graphical information from websites (i.e., "servers). (*Id.*) Given the well-known ability to create custom cursors and the equally well-known ability to transmit information between server and client, it is unsurprising that it was also well-known to combine these practices to transmit content between a server and a client
20 to modify a cursor's image file. (*Id.* ¶¶ 30-31.) This approach was built into the

widely-used “X-Windows” system, which had a separate graphics “server” to provide graphics capability to application clients. (*Id.*) Separating the client application and graphics server meant that applications that wanted to render an image onscreen must transmit all relevant graphical information over a network to the remote graphics server, whether the information to be displayed was text, graphics, or custom cursor images, which the X-Windows system supported. (*Id.* ¶ 33.)

C. Summary of the ’449 Patent

The ’449 Patent is directed toward online advertising, using a cursor that changes according to onscreen content, over a client-server network and in HTML documents (i.e., webpages). These were all concepts well-known in the prior art.

The ’449 Patent discloses “[a] system for modifying a cursor image...to a specific image having a desired shape and appearance” where the cursor image is an image or “icon[] which represent[s] the cursor or pointer.” (Ex. 1001 at

Abstract, 3:38-39.) The Background explains that “there is a need for a simple means to deliver advertising elements, i.e. logos, animations, sound, impressions, text, etc....without the passiveness of ordinary banner and frame advertisements which can be easily ignored.” (*Id.* at 2:27-32.) So, the inventors proposed changing the cursor’s appearance to a “specific image,” e.g., a corporate name, logo, icon, slogan, or animated image, to provide on-screen advertising. (*Id.* at

2:44-47, 2:63-3:3, 3:64-4:3.) The cursor's appearance can also correspond to the content on the screen. (*Id.* at 2:58-62, 7:7-9.) Other examples of cursor modification include rendering the cursor as a baseball bat on a sports website; a pink cursor on Pink Panther's website; a witch on a broomstick for Halloween; and
5 the Statue of Liberty for Fourth of July. (*Id.* at 17:33-37.) The '449 Patent teaches that it was known in the art for pointers and cursors to change shape. (*Id.* at 3:45-49.)

Figures 7-8 below show how the cursor is modified from a standard pointer arrow (designated "44" and with blue highlight added) into the "specific image" of
10 a bottle (designated "44a" and with red highlight added) to advertise a cola:



D. Level of Ordinary Skill in the Art

A person of ordinary skill in the art (“POSITA”) at the time of the claimed invention would have experience in the fields of UI design and OSs. They would have at least a master’s degree in Computer Science, Computer Engineering, or a

related field, or hold a bachelor's degree in Computer Science, Computer Engineering, or equivalent and have at least two years of relevant work experience in the fields of UI design and OSs.

Dr. Bederson qualified as a POSITA by the asserted priority date and remains qualified to testify to what such a person would have understood at the time of the claimed invention. Dr. Bederson holds a Doctorate in Computer Science from New York University, and has been a professor at the University of Maryland since 1998 focusing on human computer interaction, information visualization, and zoomable UIs. (Ex. 1003 ¶¶ 2-11.)

V. PROPOSED CLAIM CONSTRUCTIONS

The '449 Patent has expired. Accordingly, the terms are accorded their ordinary and customary meaning as understood by a POSITA at the time of the invention. *See In re Rambus Inc.*, 694 F.3d 42, 46 (Fed. Cir. 2012); *Black & Decker, Inc. v. Positec USA, Inc.*, 646 Fed. App'x 1019, 1024 (Fed. Cir. 2016) (in IPRs, "[c]laims of an expired patent are given their ordinary and customary meaning in accordance with...*Phillips*") (internal citations omitted).

"[T]he best source for understanding a technical term is the specification from which it arose, informed, as needed, by the prosecution history." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (en banc). "[T]he specification may reveal a special definition given to a claim term by the patentee that differs

from the meaning it would otherwise possess. In such cases, the inventor's lexicography governs." *Id.* at 1316 (citing *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002)).

In *APMEX* where Lexos asserted the '449 Patent, "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" was construed to mean "an image representative of at least a portion of the subject or topic being displayed on the screen." No other claim terms have been construed.

Petitioner proposes construction of the following terms pursuant to *Phillips*.

Each of these terms appears in every Challenged Claim.

A. "cursor image" / "initial cursor image"

"Cursor image" and "initial cursor image" mean "a standard/generic icon controlled by the operating system to show the position of a pointing device's pointer on the user terminal's display," which is the ordinary and customary meaning as understood by a POSITA at the time of invention, considering the claim language, specification, and prosecution history. (Ex. 1003 ¶ 55.)

This construction is consistent with how the '449 Patent uses the phrase to describe ways to modify generic cursors (black arrow, pointing finger, hourglasses, etc.): "[The purported invention] can be used to replace not just the standard arrow but other standard cursors as well." (Ex. 1001 at 17:41-45; *see* Ex. 1001 at 3:57-

64; Ex. 1003 ¶ 56.) The modified cursor can be reverted to a “generic pointer image.” (Ex. 1001 at 18:4-7.)

A user utilizes a pointing device to “move the cursor over objects, buttons, menus, scroll bars, etc. which appear on-screen.” (Ex. 1001 at 3:22-35.) A user terminal’s OS uses “drivers” to access peripheral devices attached to the user terminal, including a display driver that provides the OS access to the displayed cursor image or pointer, and a mouse driver that provides the OS with access to the mouse. (Ex. 1001 at 7:46-55; Ex. 1003 ¶¶ 58-59.) The OS also controls the behavior and appearance of the cursor. (Ex. 1001 at 8:34-37.)

B. “specific image” / “specific cursor image”

“Specific image” and “specific cursor image” mean “a predetermined image, different from the cursor image, that is drawn in its entirety in place of the cursor image,” which is the ordinary and customary meaning as understood by a POSITA at the time of invention, considering the claim language, specification, and prosecution history. (Ex. 1003 ¶ 61.) The invention provides a “server system for modifying a cursor image to a specific image.” (Ex. 1001 at 2:40-43.) The specification contains several examples of a standard cursor image being modified to take on the appearance of a different, predetermined image: on a page advertising “Fizzy Cola,” a bottle; a baseball bat; a witch-on-a-stick; and the Statue of Liberty. (Ex. 1001 at 17:5-14, 17:33-40; Ex. 1003 ¶ 64.)

Moreover, the specific image is drawn in its entirety to modify the cursor image. The specification describes “the invention” and “the present invention” modifying a cursor image “to a specific image” (Ex. 1001 at Abstract, 2:40-57 (emphasis added)), and each independent claim in the Lexos Patents recites either
5 “modifying a cursor image to a specific image” (claims 1, 48, 70, 71, 74, and 75 of the ’102 Patent, and claims 1, 27, 91, and 112 of the ’449 Patent (emphasis added)) or “modifying said cursor image to said cursor image *in the shape and appearance of* said specific image” (claims 28, 72, and 73 of the ’102 Patent, and claims 53 and 72 of the ’449 Patent (emphasis added)). Ex. 1003 ¶ 63.)

10 **C. “modifying a cursor image” / “modified cursor image”**

“Modifying a cursor image” and “modified cursor image” mean “changing the shape and appearance of a cursor image,” which is the ordinary and customary meaning as understood by a POSITA at the time of invention, considering the claim language, specification, and prosecution history. (Ex. 1003 ¶ 66.) Each

15 independent claim in both Lexos Patents recites either modifying or transforming a cursor image “to a specific image” or “to a cursor image in the shape and appearance of [a] specific image.” (Ex. 1001 claims 1, 28, 48, and 70-75; Ex. 1002 claims 1, 27, 53, 72, 91, and 112). The Abstract describes “[a] system for modifying a cursor image...to a specific image having a desired shape and
20 appearance.” Furthermore, a server system “remotely defines and manages the

shape and appearance of the cursor image in accordance with a pre-specified condition.” (Ex. 1001 at 7:5-7.) “The shape and appearance of the cursor image may correspond to the actual content of the data being provided to the user.” (Ex. 1001 at 7:7-9; Ex. 1003 ¶¶ 67-69.)

5 **D. “cursor image data”**

“Cursor image data” means “data corresponding to the specific image that is retrieved by the cursor display code each time the cursor is modified,” which is the ordinary and customary meaning as understood by a POSITA at the time of invention, considering the claim language, specification, and prosecution history.

10 (Ex. 1003 ¶ 70.) “Cursor image data” appears in every independent claim and the Abstract. In each independent claim of the Lexos Patents, “cursor image data” is described as “corresponding to said specific image.” Additionally, each independent claim of the Lexos Patents further recites that the location of the cursor image data is indicated in the cursor display instruction, which is then
15 transmitted as part of the specified content information to the remote user’s terminal. There, the cursor display instruction (indicating the cursor image data’s location) operates with or is processed by the cursor display code to modify the cursor image. (Ex. 1001 claims 1, 28, 48, and 70-75; Ex. 1002 claims 1, 27, 53, 72, 91, and 112).

20 The Abstract describes the cursor image data as “corresponding to the

specific image” and that the “cursor display instruction is operable to modify, in conjunction with...the cursor image data, a cursor image displayed by a display of the remote terminal in the shape and appearance of the specific image.” (Ex. 1001 at Abstract.) So, although the cursor display instruction contains an indication of the location of the cursor image data, the cursor display instruction *in conjunction with* the cursor image data, is operable to modify the cursor image. In other words, not just the location of the cursor image data is needed, but the cursor image data itself is needed, to modify the cursor image. (Ex. 1003 ¶¶ 71-72.) An ActiveX control, i.e. the cursor display code, retrieves data corresponding to the specific image from local memory or a remote server. (Ex. 1001 at 10:47-51, 11:19-27; Ex. 1003 ¶ 72.) The ActiveX control then invokes the OS’s application programming interface (“API”) and causes the cursor image on the user screen to change to a different image. (Ex. 1001 at 11:47-61; Ex. 1003 ¶ 72.)

E. “cursor display code”

“Cursor display code” means “code that, in response to receiving cursor display instruction, interfaces with the operating system to cause it to modify the cursor image,” which is the ordinary and customary meaning as understood by a POSITA at the time of invention, considering the claim language, specification, and prosecution history. (Ex. 1003 ¶ 74.) The cursor display code interacts with the OS’s API to “change, transform[], or swap” the cursor. (Ex. 1001 at 8:52-57,

10:37-42; Ex. 1003 ¶ 75.) When an application (e.g., browser) encounters cursor display instructions, which contain an identifier of the cursor display code, the cursor display code is invoked. (Ex. 1001 at 9:5-11; Ex. 1003 ¶ 75.) Additionally, the cursor display instruction includes information that the cursor display code uses
5 to control the display drivers that provide the OS access to the displayed cursor image. (Ex. 1001 at 8:59-64.) The cursor display code retrieves cursor information from the user terminal's memory or from a remote site and passes the information to display drivers via the OS's API. (Ex. 1001 at 9:7-12; Ex. 1003 ¶ 75.)

10 **F. "cursor display instruction"**

"Cursor display instruction" means "information transmitted by a server computer to a remote terminal which triggers the cursor display code to change the cursor image to the specific image," which is the ordinary and customary meaning as understood by a POSITA at the time of invention, considering the claim

15 language, specification, and prosecution history. (Ex. 1003 ¶ 77.) The cursor display instruction includes information that the cursor display code uses to control the display drivers that provide the OS access to the displayed cursor image. (Ex. 1001 at 7:49-55, 8:59-62; Ex. 1003 ¶ 78.) Thus, the cursor display instruction causes the invocation of an OS function that modifies the cursor image. (Ex. 1001
20 at 11:47-49; Ex. 1003 ¶ 78.)

Information for changing the cursor's appearance ("cursor display instructions") is embedded in webpages fetched from servers and viewed on user terminals. (Ex. 1001 at 4:25-27, 9:21-23, 13:38-41; Ex. 1003 ¶ 78.) A browser or browser extension interprets the cursor display instructions and instructs the OS to display the desired cursor image. (Ex. 1001 at 4:32-49.) Cursor display instructions include information like type of cursor image, location of the specific image, type of modification intended, etc. (Ex. 1001 at 9:36-6, Fig. 4.)

VI. THERE IS A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE '449 PATENT IS UNPATENTABLE

The Challenged Claims of the '449 Patent are directed toward online advertising using concepts well known in the prior art, such as cursor transformation according to onscreen content transmitted over a client-server network and in HTML documents. The Challenged Claims are unpatentable under 35 U.S.C. § 103 for reciting known, predictable, and/or obvious combinations of the prior art.

A. Identification and Overview of Key Prior Art References

Malamud and Baker each teach modifying cursors according to onscreen content. Anthias teaches modifying cursors in a client-server system. Nielsen teaches changing a cursor in an HTML document displayed in a browser.

Combined, these four patents teach the Challenged Claims. Malamud, Baker,

Anthias, and Nielsen were not presented to nor considered by the PTO during prosecution of the application leading up to the issuance of the Lexos Patents.

1. *Malamud* (Ex. 1004)

Malamud describes a cursor that changes according to the content displayed
onscreen. The Malamud patent was filed on October 26, 1994 and issued on
August 20, 2002, making it prior art under pre-AIA 35 U.S.C. § 102(e)(2).

Malamud discloses “information cursors” for use in OSs or applications,
each having a pointing portion and an information portion to display information
about an object to which the pointing portion points. (Ex. 1004 at Abstract.)³ The
information in the information portion may include the object’s name, a preview of
the object’s contents, or property information about the object. (*Id.*) The pointing
portion may indicate that a user selected one of the objects that is displayed on the
UI. The information cursor’s position on the UI changes in response to a user
moving the input device attached to the computer system. (*Id.* at 2:20-24.)

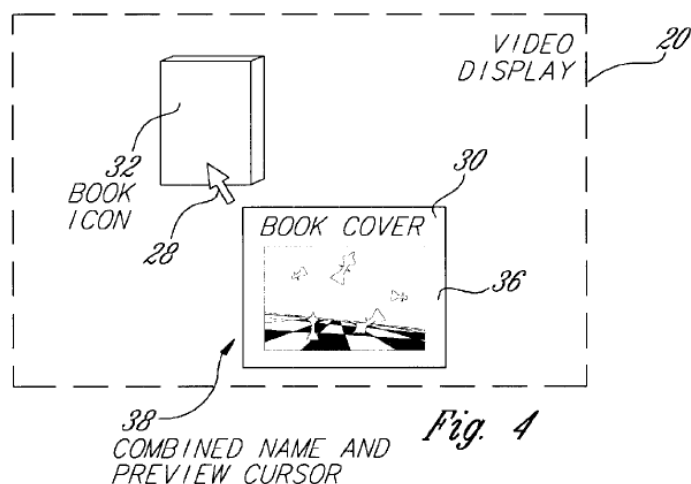
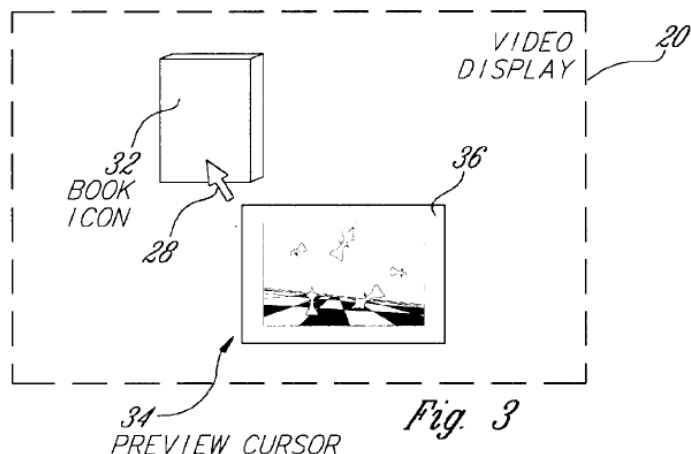
Malamud explains that information cursors may be implemented on and by
OSs such as Windows 3.1. (*Id.* at 3:6-8, 4:34-37.) The OS logically divides the UI
into several program windows, and each window has an associated window

³ Citations to prior art references in this Petition are exemplary. Additional portions of each reference can be found in Exs. 1013-1014.

procedure. (*Id.* at 4:53-55.) The OS maintains a message queue for each program, and when an event occurs (e.g., mouse movement), the event is translated into a message that is put in the program's message queue, and the program retrieves and delivers the message to the proper window procedure for processing (*Id.* at 4:57-
5 65.) When a user positions the information cursor over an object within a program's window, a message specifying the cursor's position travels from the OS to program's message queue, to the program, and on to the program window's window procedure. (*Id.* at 4:66-5:14.) The window procedure uses the information in the message to respond to the cursor's position and other activities,
10 including determining whether the cursor is pointing to an object and, if so, displaying in the cursor's information portion information about the object. (*Id.* at 5:7-10, 5:27-39).

Malamud teaches four types of information cursors: name, preview, combined name and preview, and property cursors. (*Id.* at 3:10-12.) All four have
15 a pointing portion, and each has its own type of information portion that appears when the cursor's pointing portion points at an object onscreen. The name cursor has a name box that displays the object's name. (*Id.* at 3:31-34, 3:54-56.) The preview cursor has a preview portion that displays a graphical preview of the object's contents. (*Id.* at 3:61-63, Fig. 3 (shown below).) For example, when the
20 preview cursor points to the icon of a book about chess, its preview portion

displays an image of chess pieces. (*Id.* at 3:65-4:3.) The property cursor has a property box that displays the object's property information. (*Id.* at 4:23-25.) A combined name and preview cursor has at least one name box and a preview portion. (*Id.* at 4:4-18, Fig. 4 (shown below).)



5

2. *Anthias* (Ex. 1005)

Anthias teaches the management of application windows on a client computer, in a “client server model” (Ex. 1005 at Abstract), including the cursor’s appearance at different positions onscreen. The *Anthias* patent was filed on

December 6, 1993 and issued on July 6, 1999, making it prior art under pre-AIA
35 U.S.C. § 102(e)(2).

Anthias teaches a distributed data processing system in which a “client” runs
an application and uses its presentation system to transmit to the “server” only the
5 minimum information needed to display on the “server” what is dictated by the
application running on the “client” computer. (*Id.* at 2:49-65, 3:2-7.)

Traditionally, in the client-server model, the term “client” refers to a local
computer used by an end user to access information stored at a remote computer
called a “server.” (Ex. 1003 ¶ 101.) In Anthias (and in X-Windows discussed
10 above), however, the user interacts with a local computer called a “server” to
access an application program stored remotely on a “client” computer. (Ex. 1005
at Abstract, Figs. 1-2.) The “client” presentation system associates a particular
cursor type with the “device area” in a window. (*Id.* at 3:2-7.) When “a
pointer/cursor [is] associated with a device area” in a “client” application’s
15 window displayed on the “server” (e.g., “the end user can use [the device area] to
input information to the application that is running on the client computer” (*id.*)),
the “server” “responds to mouse movement by redrawing the associated pointer.”
(*Id.* at 3:29-32, 4:18-22.) The cursor will change shape, color or flashing
frequency as it passes from background window areas to the device area. (*Id.* at
20 4:22-24.) For example, the “client” presentation system requests the “server” to

change the cursor for text input fields that require a special cursor. (*Id.* at 5:24-26.)

3. *Nielsen* (Ex. 1006)

Nielsen teaches a changing cursor in HTML document (i.e., webpage). The Nielsen patent was filed on September 27, 1996 and issued on November 23, 1999,
5 making it prior art under pre-AIA 35 U.S.C. § 102(e)(2).

Nielsen discloses enhancements to defined areas on image maps in HTML documents to aid hearing impaired users to navigate webpages by playing audio clips when the cursor is positioned in those defined areas. (Ex. 1006 at Abstract.)

An information display system includes an image map detection mechanism

10 configured to detect an image map associated with an image area. (*Id.* at 4:59-61.)

When a browser detects that a cursor is positioned over the image of a client-side image map, the browser runs a “Relevant Active Area Hit” procedure to determine whether the cursor is within a selectable area on the image map, and whether that area is associated with an audio attribute. If so, a “hit” has occurred in a relevant

15 active area and an audio clip is played. (*Id.* at 8:24-48.)

4. *Baker* (Ex. 1007)

Baker describes a cursor that changes according to its position onscreen and the content displayed. The Baker patent was filed on September 30, 1994 and issued on February 3, 1998, making it prior art under pre-AIA 35 U.S.C. §

20 102(e)(2).

Baker teaches a pictorial or graphical UI for accessing information in an electronic file system: “[A] user definable graphical interface for a computer OS which utilizes pictorial information and animation.” (Ex. 1007 at Abstract, 1:11-13.) An animated character is overlaid on a background image, which also
5 contains icons representing file objects. (*Id.*) User input causes the animated character to move to different positions on the display to interact with the icons in the background image. (*Id.* at 1:11-13, 13:9-12.)

The animated character “roughly corresponds to the cursor or pointer” in a conventional graphical UI. (*Id.* at 13:12-13, 13:30-42.) Although the inventor
10 served as his own lexicographer by defining “cursor” as only the hotspot on the animated character (*id.* at 13:18-20), Baker’s animated character *plus* its active hotspot is what would traditionally be understood as a “cursor.” (Ex. 1003 ¶ 105.) For this reason, the animated character *plus* hotspot will be referred to herein as a “cursor” and its hotspots will be referred to herein as “hotspots.” The hotspot
15 corresponds to different components of the animated character at different times (e.g., foot as hotspot, left hand at hotspot), and the cursor location is updated after each animation. (*Id.* at 13:14-18.)

When the animated character cursor is moved by the user, it is shown walking. (*Id.* at 13:47-49.) And when the animated character cursor interacts with
20 an icon, both become animated. (*Id.* at 13:49-51.) Information about the character

is stored in the “character_model” list or array. (*Id.* at 23:1-8.) Character_model maintains the location and position of the animated character cursor and its hotspot, and is updated by the play_animation function to match the last frame of each animation played. (*Id.* at 23:20-26.) Character_model is used to calibrate
5 animations and register images of the character across animations, and maintain the list of references to file objects. (*Id.* at 23:23-29.) Additionally, Baker teaches storing pointers to bitmap animation frames and information about the modifications for each animation in the header of the animation. (*Id.* at 24:33-67).

The character cursor may be animated to “jump” into an icon that references
10 a directory file object to open the directory file. (*Id.* at 15:9-11.) When animation playback is terminated, the cursor location stored with the final frame of the animation is used to update the system global cursor. (*Id.* at 15:14-18.) To select an icon, the character is animated to “pick[] up” the icon by issuing the set_selection_arg command. (*Id.* at 16:21-27.)

15 The pictorial UI’s basic processes are grouped around three functional tasks: (1) input interpretation, including cursor control; (2) basic execution unit, including prologue and epilogue animations signaling receipt of a user command and end of an animation, respectively; and (3) system control, which controls execution of input interpretation. (*Id.* at 10:62-11:1, 14:30-38.) While the pictorial
20 UI includes an animated character model (*id.* at 23:1-24:3) and animation scripts

(*id.* at 24:28-25:67), as with all displayed computer images, the animated character cursor and icons are ultimately rendered by functions and applications employed by the OS. (Ex. 1003 ¶ 109.) The animation frames that Baker’s UI displays must therefore be identified or transmitted to those functions and applications for display. (*Id.*)

B. Motivation to Combine References

A patent claim is invalid as obvious if the differences between the claimed subject matter and the prior art are such that the subject matter as a whole would have been obvious to a POSITA at the time the invention was made. 35 U.S.C. §

103. The Supreme Court has given examples of motivations for combining or modifying prior art references, including “interrelated teachings of multiple patents,” and the “background knowledge,” “creativity,” and “ordinary skill and common sense.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418-21 (2007). The Court has further recognized that a combination may be “obvious to try” if “[w]hen there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill [in the art] has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense.” *Id.* at 421. The Federal Circuit has provided similar guidance, noting that “[t]he reason, suggestion, or motivation to combine

may be found explicitly or implicitly: 1) in the prior art references themselves; 2) in the knowledge of those of ordinary skill in the art that certain references, or disclosures in those references, are of special interest or importance in the field; or 3) from the nature of the problem to be solved.” *Perfect Web Techs., Inc. v.*

5 *InfoUSA, Inc.*, 587 F.3d 1324, 1329 (Fed. Cir. 2009). A POSITA would have known to combine the teachings in Malamud, Anthias, Nielsen, and Baker in the manner claimed by the Challenged Claims.

Malamud, Anthias, Nielsen, and Baker are contemporary references from the same field of endeavor: responding to a cursor’s location on the screen. (Ex. 1003 ¶ 111.) Malamud teaches an information cursor, consisting of a pointing
10 portion and an information portion, that displays information about an object to which its pointing portion points. (Ex. 1004 at Abstract.) Anthias teaches modifying a cursor as it moves from one area of the screen to another. (Ex. 1005 at 4:21-26.) Nielsen teaches playing a sound when the cursor is positioned over a
15 defined area on the screen. (Ex. 1006 at 8:24-32.) And, Baker teaches a character cursor that is animated in different ways depending on its location on and movement across the screen. (Ex. 1007 at 15:1-22.) *See KSR*, 550 U.S. at 402.

A POSITA in the mid- to late-1990s looking to adapt custom cursors of Malamud or Baker for websites would have looked to Anthias and Nielsen, which
20 disclose using a web-based server for tracking the location of and modifying the

cursor on a remote computer. (Ex. 1003 ¶ 112.) A POSITA desiring to provide graphical UI elements, such as a cursor, over a network would consult references such as Anthias, which teaches “a client server model [that] includes an application executing on a client system with graphics being drawn on the server system for the end user” (Ex. 1005 at Abstract; Ex. 1003 ¶ 112.), and Nielsen, which teaches playing an audio clip when a cursor is positioned over a predefined area on an image map of an HTML page viewed in a browser application. (Ex. 1006 at Abstract, 4:8-19; Ex. 1003 ¶ 112.) Anthias discloses externalizing certain graphics functions to another computer, one of which is the modification of the cursor on the screen. (Ex. 1005 at 5:26-30 (“The client presentation system then...request[s] the server to change the cursor for text input fields which require a special cursor.”).) A POSITA would have recognized that Anthias’s system provides the benefit of limiting network traffic and storage and processing overhead in the server computer. (Ex. 1005 at 2:30-34; Ex. 1003 ¶ 112.) While Anthias teaches use of client-server architecture, a POSITA would have looked to a publication like Nielsen that explains tracking the cursor on an HTML page in a browser and generating a response based on the cursor location. (Ex. 1006 at 9:43-47, 9:64-67; Ex. 1003 ¶ 112.) A POSITA would have recognized that the client-server architecture of Anthias and the use of HTML in Nielsen could be combined with Malamud and Baker to customize cursors for websites. (Ex. 1003 ¶ 112.)

Malamud, Anthias, Nielsen, and Baker disclose pre-existing building blocks for delivering customized cursors for websites that when put together without any modification performed all steps of the Challenged Claims. Where a combination “yields no more than one would expect from such an arrangement,” such a
5 combination would have been obvious for a person of ordinary skill in the art to try. *See KSR*, 550 U.S. at 417. The disclosures of Malamud, Anthias, Nielsen, and Baker were well within the grasp of a person of ordinary skill in the art because applications for all four patents were filed in a three-year period.

There is no reason why the references could not have been combined
10 according to known methods to yield predictable results with a reasonable expectation of success. *See id.* at 416. The combination of modifying a cursor when it points to an object (Malamud) or moves across the screen (Baker) for advertising purposes (Baker), distributing graphics functions across client and server machines (Anthias), tracking cursor movement on an HTML page in a
15 browser (Nielsen) did “no more than one would expect from such an arrangement.” *Id.* at 417. Moreover, the combination represented the obvious path for UI developers, who were faced with a transition from developing interfaces for applications on non-networked personal computers to designing webpages viewable via the Internet. (Ex. 1003 ¶ 112.)

C. Ground 1: Claims 1, 7, 15, 27, 33, 41, 53-54, 63, 72-73, 82 Are Rendered Obvious by *Malamud* and *Anthias*

1. Claim 1 Is Rendered Obvious by *Malamud* and *Anthias*

**Element 1[Preamble]: A server system for modifying a cursor
5 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 teaches an “information cursor” that includes a pointing portion
and an information portion. (Ex. 1004 at Abstract.) The information cursor is
10 “generally manipulable by an input device, such as a keyboard or a mouse,” and its
pointing portion is a standard/generic cursor (e.g., arrow pointer) (*Id.* at 1:16-18,
Figs. 2a-5, 5:66-6:17 (“conventional cursor”)) (Ex. 1003 ¶ 116). When its pointing
portion points to an on-screen object, the information cursor is modified to display
its information portion (“*modifying a cursor image to a specific image*”⁴), as drawn
15 by a function or application employed by the computer OS (“*displayed on...a
remote user’s terminal’s*”). (Ex. 1004 at 3:59-6:3; Ex. 1003 ¶ 116.) In the cases of
the preview and combined name and preview cursors, the cursor’s preview portion
displays a graphical preview of the contents of an object. (Ex. 1004 at 3:59-4:18.)⁵

⁴ Hereinafter, language from the Challenged Claims is italicized.

⁵ Because the combined name and preview cursor includes a preview portion
that operates in the same manner as the preview cursor’s preview portion, this

It would have been obvious to a POSITA to combine Malamud and Anthias, which teaches a data processing system implemented on a client-server model in which an application runs on a “client” (traditionally, server) system and graphics data for the application is sent to and drawn by a “server” (traditionally, client) system. (Ex. 1005 at 1:24-33; Ex. 1003 ¶ 117.) A particular cursor type is associated with “device areas” onscreen and, when the cursor moves from a background window area into a “device area,” the client presentation system instructs the server to redraw the cursor by changing its shape, color, or flashing frequency to match the particular cursor type. (Ex. 1005 at 4:16-25; Ex. 1003 ¶ 117.)

Combining Malamud and Anthias, a POSITA would have recognized that the computer OS in Malamud corresponds to the “*remote user’s terminal*” in Claim 1 and the application program’s window procedure corresponds to the “*server.*” (Ex. 1003 ¶ 118.)

Petition focuses on the operation of the preview cursor for the sake of simplicity. All discussion of how Malamud’s disclosure of a preview cursor invalidates the ’449 Patent applies similarly to Malamud’s disclosure of a combined name and preview cursor.

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

When a preview cursor points to a UI object, a graphical preview of the object appears in the cursor's preview portion. (Ex. 1004 at 3:63-67; Ex. 1003 ¶ 122.) The OS stores bitmaps ("*specific image*") that are to be displayed in the preview portion and other information corresponding to the preview cursor, including pointers to the bitmaps ("*cursor image data*"). (Ex. 1004 at 5:16-18, 5:59-62; Ex. 1003 ¶ 122.) These bitmaps are predetermined, static images that are different from the "conventional cursor" initially displayed, and that are displayed in their entirety in the information portion of Malamud's information cursor. (Ex. 1003 ¶ 122.) Each time that the information cursor's pointing portion points at an object, the bitmap and other information corresponding to the preview cursor are retrieved by the OS to check that the information cursor is turned on and to modify the cursor by displaying its preview portion ("*cursor display code*"). (Ex. 1003 ¶ 122.)

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

Malamud teaches the use of information cursors in conjunction with a conventional OS (e.g., Windows), which a POSITA would understand includes functions or applications to display and modify graphics on the UI ("*cursor display*

code”) including cursors in their standard forms as well as any subsequent modifications (“*cursor display code operable to modify said cursor image*”). (Ex. 1004 at 3:6-8, 4:33-38, 5:47-53, Fig. 6; Ex. 1003 ¶ 125.) The OS divides the UI into program windows, each having an associated window procedure. (Ex. 1004 at 3:53-55; Ex. 1003 ¶ 126.) When the cursor is moved into the window of an application program, the OS sends a message (specifying the cursor’s position) to the program, which forwards the message to the window procedure (“*first server computer*”) associated with the window. (Ex. 1004 at 5:24-28; Ex. 1003 ¶ 127.) The window procedure then determines whether the position corresponds to the position of an object (i.e., whether the cursor is pointing to an object). If so, “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 5:28-53; Ex. 1003 ¶ 126.) The OS then employs a function or application (“*cursor display code*”) to modify the information cursor to display its preview portion.

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

Malamud, in view of Anthias, discloses an application program’s window procedure (“*first server computer*”) and an OS (“*remote user terminal*”) on a client-server network. Malamud teaches that the window procedure transmits a

message (“*specified content information*”) to the OS, which employs functions or applications (“*cursor display code*”) to display its information portion. (Ex. 1004 at 4:53-54, 5:53-57; Ex. 1003 ¶¶ 129-130.) (See section VI.C.1, Element 1[Preamble] regarding Anthias.)

5 **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**
10 **appearance of said specific image,**

Malamud teaches that information cursors are invoked when an application’s window procedure sends a message (“*specified content information*”) to invoke a preview cursor and the OS employs a function or application (“*cursor display code*”) to check whether the information cursor is “on”, and if so, instructing the
15 OS to invoke functions or applications to display a certain type of cursor that includes text or graphical previews when pointed to a displayed object. (Ex. 1004 at 3:6-8, 4:53-55, 5:47-57, Fig. 6; Ex. 1003 ¶ 134.) In the case of preview cursors, using a portion of the message (“*specified content information*”) transmitted by the window procedure (“*server*”) to the OS (“*remote terminal*”) and that includes a
20 pointer to the location of the bitmap of an image (“*cursor display instruction indicating location of said cursor image data*”), the function or application that the

OS uses (“*cursor display code*”) displays the bitmap image in the preview cursor’s preview portion (“*cursor display instruction and said cursor display code operable...shape and appearance of said specific image*”). (Ex. 1004 at 5:57-62; Ex. 1003 ¶ 136.) For example, when the preview cursor points to an icon for a book about chess that is displayed on the screen, the preview portion displays a graphic of chess pieces (“*specific image includes content corresponding to at least a portion of said information*”). (Ex. 1004 at 3:32-35, 3:59-4:3, Fig. 3-4; Ex. 1003 ¶ 136.)

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,

Malamud, in view of Anthias, discloses that an application program’s window procedure (“*first server computer*”) transmits a message (“*specified content information*”) to the OS (“*remote user terminal’s*”), which then sends it to the functions and applications the OS employs (“*cursor display code*”) for displaying the cursor’s information portion on the screen. (Ex. 1004 at 4:53-54, 5:53-57; Ex. 1003 ¶ 140.) Modifying an information cursor to display its information portion initiates when “the cursor position is moved by the mouse

or other input device to point within the window,” and the OS generates and sends a message to the program which forwards the message (with new cursor position) to the program’s window procedure. (Ex. 1004 at 5:22-28.) The application window procedure then responds with its own message to the OS that “tells the operating system what type of cursor to display....” (Ex. 1004 at 5:47-53; Ex. 1003 ¶ 140.) This process of sending new cursor positions to the program’s window procedure allows the OS (“*remote user terminal*”) to request that the program’s window procedure (“*first server computer*”) send a message back identifying the type of cursor to be displayed (“*specified content information*”). (Ex. 1004 at 5:7-10.) The message may further comprise information to be displayed on the user’s terminal because, in the case of a combined name and preview cursor, the application program’s window procedure sends a message to the OS and the functions and applications it employs that includes the name of the object (“*specified content information further comprises information to be displayed*”). (Ex. 1004 at 4:4-17; Ex. 1003 ¶ 140.)

(See section VI.C.1, Element 1[Preamble] regarding Anthias.

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
5 **a portion of said information to be displayed on said display of said remote user's terminal.**

Malamud teaches the use of information cursors in conjunction with a conventional OS (e.g., Windows), which uses functions or applications to display and modify graphics on the UI (“*cursor display code*”) including cursors in their
10 initial, standard forms as well as any subsequent modifications (“*cursor display code operable...shape and appearance of said specific image*”). (See section VI.C.1, Element 1[b] regarding Malamud.) In the case of preview cursors, the message (“*specified content information*”) transmitted by the window procedure (“*server*”) to the OS (“*remote terminal*”) and the functions and applications it
15 employs containing a pointer to the bitmap (“*cursor display instruction*”) is processed by those functions and applications (“*cursor display code*”), which then display the cursor’s preview portion (“*modify said cursor image to said...specific image*”). (Ex. 1004 at 5:59-62; Ex. 1003 ¶¶ 144-145.) Receipt of the message containing a pointer to the bitmap causes the OS to invoke the function or
20 application for display, which then retrieves the bitmap for display on the user screen: “[I]f the cursor to be displayed is a preview cursor 34 (FIG. 3), a message

specifying that a preview cursor is required is sent [to the OS]. The message includes a pointer to a bitmap of graphical information that the operating system 22 should use in the preview portion 36.” (Ex. 1004 at 5:57-62; Ex. 1003 ¶ 146.)

When a preview cursor points to an object displayed on the screen, the
5 cursor is modified so that the preview portion appears displaying graphical data depicting the object’s contents. (Ex. 1004 at 3:32-35, 3:61-63.) (See, for example, section VI.C.1, Element 1[c][ii] regarding Malamud.)

2. Claim 7 Is Rendered Obvious by *Malamud* and *Anthias*

**Claim 7: The server system in accordance with claim 1, wherein
10 said specific image has a shape and appearance corresponding to said information to be displayed on said display of said user’s terminal.**

Malamud discloses that the information portion of an information cursor displays text or graphics related to the UI object on the screen to which the
15 information cursor is pointing. (See section VI.C.1, Elements 1[Preamble] and 1[c][iv] regarding Malamud and Anthias.)

3. Claim 15 Is Rendered Obvious by *Malamud* and *Anthias*

**Claim 15: The server system in accordance with claim 1, said
20 cursor display instruction further comprising an image identifier indicating said cursor image data corresponding to said specific image.**

Malamud discloses, in view of Anthias, that, in case of preview cursors, an

application program's window procedure message to the OS and the functions and applications it employs for displaying the cursor's information portion on the screen includes a pointer to the location of the bitmap of an image ("*cursor display instruction further comprising an image identifier*"). (Ex. 1004 at 4:53-54, 5:53-62; Ex. 1003 ¶¶ 153-154.) (See section VI.C.1, Element 1[c][ii] regarding Malamud.)

4. Claim 27 Is Rendered Obvious by *Malamud* and *Anthias*

See claim 1. Claim 27 is substantially identical to claim 1 except that claim 27 recites modification of the cursor image in response to "movement of said cursor image over a specified location" rather than in response to "movement of said cursor image over a display of said at least a portion of said information to be displayed." (See Ex. 1003 at ¶¶ 156-160.) Malamud and Anthias satisfy claim 27 for the same reasons as claim 1.

5. Claim 33 Is Rendered Obvious by *Malamud* and *Anthias*

See claim 7.

6. Claims 41 and 82 Are Rendered Obvious by *Malamud* and *Anthias*

See claim 15.

7. Claim 53 Is Rendered Obvious by *Malamud* and *Anthias*

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:

As explained in Element 1[Preamble] (Section VI.C.1), Malamud, in view of Anthias, discloses an application program's window procedure ("*server*") and an OS ("*user terminal*") on a client-server network. Malamud teaches an "information cursor" that includes a pointing portion and an information portion.

5 (Ex. 1004 at Abstract.) The information cursor is "generally manipulable by an input device, such as a keyboard or a mouse," and its pointing portion is a standard/generic cursor (e.g., arrow pointer). (*Id.* at 1:16-18, Figs. 2a-5, 5:66-6:17 ("conventional cursor"); Ex. 1003 ¶ 167.) When the information cursor's pointing portion points at an on-screen object, the information cursor is modified to display
10 its information portion ("*modifying an initial cursor image*"), as drawn by the OS via a function or application employed by the OS ("*displayed on...a remote user's terminal*"). In the case of preview and combined name and preview cursor types, the cursor includes a pointing portion and a preview portion (showing a graphical preview of an object's contents). (Ex. 1004 at 3:59-4:18.)

15 It would have been obvious to a POSITA to combine Malamud and Anthias, which teaches a data processing system implemented on a client-server model in which an application runs on a client system and graphics data for the application is being drawn on a server system. (Ex. 1005 at 1:24-33; Ex. 1003 ¶ 168.) (See section VI.C.1, Element 1[Preamble] regarding Anthias.)

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

Combining Malamud and Anthias, a POSITA would have recognized that the computer OS in Malamud corresponds to the client “*user’s terminal*” and the application program’s window procedure in Malamud corresponds to the “*server*” of Claim 53. Malamud discloses that modifying an information cursor to display its information portion initiates when the cursor is moved within an application window and the OS sends a message to the program which forwards the message (with new cursor position) to the program’s window procedure. (Ex. 1004 at 5:22-28.) The window procedure responds to the OS and the functions and applications it employs and “tells the operating system what type of cursor to display...” (Ex. 1004 at 5:47-53; Ex. 1003 ¶ 172.) Sending new cursor positions to the application’s window procedure allows the OS (“*user terminal*”) to request that the program’s window procedure (“*server*”) send a message back identifying the type of cursor to be displayed (“*request at said at least one server to provide specified content information*”). (*Id.* at 5:7-10.)

(See section VI.C.1, Element 1[c][iii] regarding Anthias.)

**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**

Malamud combined with Anthias discloses that an application program's window procedure ("*first server computer*") transmits a message ("*specified content information*") to the OS ("*user terminal's*") and the functions and applications it employs ("*cursor display code*") identifying the type of cursor to be displayed. (See section VI.C.1, Elements 1[c][ii] and [iii] regarding Malamud.)

**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**

Combining Malamud and Anthias, a POSITA would have recognized that the computer OS in Malamud corresponds to the client "user's terminal" and the application program's window procedure in Malamud corresponds to the "server" of Claim 53. (Ex. 1003 ¶ 180.) In Malamud, when the preview cursor's pointing portion points to an object displayed on the screen, the cursor is modified to display its information portion ("*transforming said initial cursor image...into the shape and appearance of said specific image*"). (Ex. 1004 at 3:59-6:3; Ex. 1003 ¶

181.) Using a portion of the message (“*specified content information*”) transmitted by the window procedure (“*server*”) to the operating system (“*remote terminal*”) and that includes a pointer to the location of the bitmap of an image (“*cursor display instruction*”), the OS users functions and applications (“*cursor display code*”) to display the bitmap image in the cursor’s preview portion. (Ex. 1004 at 5:57-62; Ex. 1003 ¶ 182.) In the case of a combined name and preview cursor, the message includes the name of the object to which the cursor points (“*specified content information further comprises information to be displayed*”). (*Id.* at 4:4-17, 5:54-62.)

(See section VI.C.1, Element 1[c][ii] regarding Malamud.)

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 wherein said specific image has a shape and appearance relating to said information to be displayed.

When the cursor is moved into the window of an application program (“*movement of said cursor image...on said display of said user’s terminal*”), the OS sends a message (specifying cursor position) to the program, which forwards

the message to the window procedure associated with the window, which then
sends a message to the OS identifying the type of cursor to display. (Ex. 1004 at
5:28-53; Ex. 1003 ¶ 186.) Receipt of the message (“*specified content
information*”) containing a pointer to graphical data (“*cursor display instruction*”)
5 causes the OS to invoke a function or application, which then retrieves the
graphical data for display on the user screen. A POSITA would have understood
that identifying the type of cursor (preview) would have indicated that the cursor’s
information portion would be displaying an image, which would may in turn
indicate that certain functions or applications are needed to retrieve and display the
10 retrieved image (“*cursor display instruction indicates a cursor display code
operable to process said cursor display instruction*”). (Ex. 1003 ¶ 188.) When the
preview cursor points to an icon for a book about chess that is displayed on the
screen, the cursor’s preview portion displays an image of chess piece shapes
 (“*specific image has a shape and appearance relating to said information to be
15 displayed*”). (*Id.* at 3:32-35, 3:59-4:3, Fig. 3-4; Ex. 1003 ¶ 188.)

(See section VI.C.1, Elements 1[b] and 1[c][ii] regarding Malamud.)

8. Claim 54 Is Rendered Obvious by *Malamud* and *Anthias*

Claim 54: The method in accordance with claim 53, wherein said
transforming further comprises executing said cursor display
20 code so as to display said specific cursor image while at least a

portion of said information to be displayed is displayed on said display of said user's terminal.

Malamud discloses that when a preview cursor points to an object displayed on the screen, the cursor is modified so that the cursor's preview portion ("*specific cursor image*") appears displaying graphical data depicting the object's contents while the object continues to be displayed on the screen. (Ex. 1004 at 3:32-35, 3:61-63; Ex. 1003 ¶¶ 191-192.)

9. Claim 63 Is Rendered Obvious by *Malamud* and *Anthias*

Claim 63: The method in accordance with claim 53 wherein said specified content information comprises an image identifier that corresponds to the location of data representing said specific image.

Malamud discloses preview cursors, the message ("*specified content information*") includes "what type of cursor to display." (Ex. 1004 at 4:47-53; Ex. 1003 ¶ 195.) Often, images are stored in a file location specifically for images (as opposed to other files, like text documents). A POSITA would have understood that identifying the type of cursor (preview) would have indicated that the cursor's information portion would be displaying an image, which would correspond to the location of the image itself ("*image identifier*").

(See section VI.C.1, Element 1[c][ii] regarding Malamud.)

10. Claim 72 Is Rendered Obvious by *Malamud and Anthias*

See claim 53. Claim 72 is substantially identical to claim 53 except that claim 72 recites modification of the cursor image in response to “movement of said cursor image over a specified location” rather than in response to “movement of said cursor image over a display of said at least a portion of said information to be displayed.” (See Ex. 1003 at ¶¶ 197-201.) Malamud and Anthias satisfy claim 72 for the same reasons as claim 53.

11. Claim 73 Is Rendered Obvious by *Malamud and Anthias*

See claim 54.

D. Ground 2: Claims 12, 14, 38, 40, 60, 62, 79, and 81 Are Rendered Obvious by *Malamud, Anthias, and Nielsen*

1. Claim 12 Is Rendered Obvious by *Malamud, Anthias, and Nielsen*

Claim 12: The server system in accordance with claim 1, wherein said specified content information is transmitted in the form of HTML files that define a web page.

It would have been obvious to a POSITA to combine Malamud, Anthias, and Nielsen, which teaches a changing cursor in a HTML document (i.e., webpage) viewable by a browser. Nielsen teaches transmitting HTML webpages over a network from a server to a client computer. (Ex. 1006 at 2:16-20; Ex. 1003 ¶ 210.) Nielsen teaches that images in HTML files are divided into image maps and positioning the cursor within a certain image map causes the browser to invoke

a procedure to play an associated audio clip. (Ex. 1006 at 8:36-43; Ex. 1003 ¶ 210.) For example, when the cursor is placed in a browser window displaying Sun Microsystems's Home Page, an audio clip of the title is played. (Ex. 1006 at 8:24-32.)

5 2. **Claim 14 Is Rendered Obvious by *Malamud, Anthias, and Nielsen***

**Claim 14: The server system in accordance with claim 1, wherein
said user terminal includes a browser application responsive to
said cursor display instruction, said browser application executing
10 said cursor display code using parameters defined in said cursor
display instruction.**

It would have been obvious to a POSITA to combine Malamud, Anthias, and Nielsen, which teaches a changing cursor in a HTML document (i.e., webpage) viewable by a browser. Nielsen teaches such a browser that detects a
15 cursor's movement within the image of a client-side image map. (Ex. 1006 at 8:36-39; Ex. 1003 ¶ 214.) Upon detecting movement, the browser runs a "Relevant Active Area Hit" procedure to determine whether the cursor is within a selectable area defined by the image map and whether that area is associated with an audio attribute. (Ex. 1006 at 8:39-43.) The cursor's coordinates are compared
20 to the areas defined for the image and, if the cursor is within a defined area having an associated audio attribute, a hit has occurred and the browser plays the associated audio. (Ex. 1006 at 8:44-48, 9:64-10:6; Ex. 1003 ¶ 214.)

3. Claims 40, 62, and 81 Are Rendered Obvious by *Malamud, Anthias, and Nielsen*

See claim 14.

4. Claims 38, 60, and 79 Are Rendered Obvious by *Malamud, Anthias, and Nielsen*

See claim 12.

E. Ground 3: Claims 2-3, 5-6, 28-29, 31-32, 55-56, 58-59, 74-75, and 77-78 Are Rendered Obvious by *Malamud, Anthias, and Baker*

1. Claim 2 Is Rendered Obvious by *Malamud, Anthias, and Baker*

Claim 2: The server system in accordance with claim 1, wherein said specific image comprises advertising material related to at least a portion of said information to be displayed on said display of said user's terminal.

Malamud teaches an “information cursor” that includes a pointing portion and an information portion containing information about an object displayed on the screen and to which the pointing portion points. (Ex. 1004 at Abstract.) The information displayed in the information portion may include the object’s name, a preview of its contents, or the object’s property information. (*Id.*) For example, when the information cursor points to a book about chess, the information portion may display the name of the book, graphic depicting chess pieces, or both. (*Id.* at 3:31-34, 3:61-4:3.) It would have been obvious to a POSITA that Malamud discloses “advertising material” because the contents of a book have been used

traditionally as advertising for the book itself. (Ex. 1003 ¶ 230.)

It would have been obvious to a POSITA to combine Malamud, Anthias, and Baker, which teaches a cursor that changes according to its position onscreen and the content displayed. Baker discloses modifying a cursor to convey

5 advertising material, specifically that pictures and objects that serve as icons in the pictorial UI may represent a vendor's goods or services. (Ex. 1007 at 11:53-56.)

The animated character cursor is modified so that it is depicted picking up or otherwise interacting with objects on the screen. (Ex. 1007 at 16:21-23.) In this manner, the depiction of the animated character cursor interacting with an object

10 on the screen comprises advertising material related to the vendor's goods or services. (Ex. 1003 ¶¶ 232-233.)

2. Claim 3 Is Rendered Obvious by *Malamud, Anthias, and Baker*

Claim 3: The server system in accordance with claim 2, wherein
15 **said advertising material further comprises a brand logo.**

Malamud discloses that the information cursor's information portion displays text or a graphical preview related to the UI object on the screen to which the cursor's pointer portion is pointing. (Ex. 1004 at 3:32-35, 3:61-63; Ex. 1003 ¶ 235.) For example, when the preview cursor points to a book icon, the cursor's
20 preview portion shows a graphical scene representing the book's contents, in this case chess pieces to represent a book about chess. (Ex. 1004 at 3:65-4.) It would

have been obvious to a POSITA that Malamud discloses a “brand logo” because traditional advertising has long utilized an image of a product to symbolize the product’s brand. (Famous examples include and Auntie Anne’s, specializing in pretzels, uses a pretzel as its logo and Planters, a company specializing in peanuts, uses an anthropomorphic peanut as its logo.) (Ex. 1003 ¶ 236.)

3. Claim 5 Is Rendered Obvious by *Malamud*, *Anthias*, and *Baker*

Claim 5: The server system in accordance with claim 2, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed on said display of said user's terminal.

Malamud discloses that the information portion of an information cursor displays text or a graphical preview related to the UI object on the screen to which the information cursor is pointing. (Ex. 1004 at 3:32-35, 3:61-63.) For example, when the preview cursor points to a book icon, the cursor’s preview portion shows a graphic representing a “goods,” in this case chess pieces to represent a book about chess. (Ex. 1004 at 3:65-4:3; Ex. 1003 ¶ 240.)

It would have been obvious to a POSITA to combine Malamud, Anthias, and Baker, which teaches a cursor that changes according to its position onscreen and the content displayed. Baker teaches that pictures and objects that serve as icons in the pictorial UI may represent a vendor’s goods or services. (See section

VI.E.1 above regarding Baker.)

4. Claim 6 Is Rendered Obvious by *Malamud, Anthias, and Baker*

**Claim 6: The server system in accordance with claim 2, wherein
said advertising material further comprises messages relating to
said information to be displayed on said display of said user's
terminal.**

Malamud discloses that the information portion of an information cursor
displays text related to the UI object displayed on the screen and to which the
information cursor is pointing. (Ex. 1004 at 3:32-35, 3:61-63; Ex. 1003 ¶ 246.)

For example, when the “name cursor” points to a book icon, the name box displays
a description of the icon, in this the phrase “Book Cover.” (Ex. 1004 at 3:39-41,
Figs. 2a-2c.) Additionally, the property cursor has a property box in which it
displays text-based property information regarding the UI object to which the
pointer portion of the cursor points. (Ex. 1004 at 4:23-28, Fig. 5.) Property
information may include help information for the object. (Ex. 1004 at 1:46-48.)

5. Claims 28, 55, and 74 Are Rendered Obvious by *Malamud, Anthias, and Baker*

See claim 2.

6. Claim 29, 56, and 75 Are Rendered Obvious by *Malamud, Anthias, and Baker*

See claim 3.

7. Claims 31, 58, and 77 Are Rendered Obvious by *Malamud, Anthias, and Baker*

See claim 5.

8. Claims 32, 59, and 78 Are Rendered Obvious by *Malamud, Anthias, and Baker*

See claim 6.

F. Ground 4: Claims 13, 39, 61, and 80 Are Rendered Obvious by *Malamud, Anthias, Nielsen, and Baker*

1. Claim 13 Is Rendered Obvious by *Malamud, Anthias, Nielsen, and Baker*

Claim 13: The server system in accordance with claim 12, wherein said cursor image data includes at least in part an advertisement for goods or services contained in said web page.

See claims 5 and 12. Malamud teaches that information cursors are invoked when an application's window procedure sends a message to the OS and the chosen functions and applications it employs instructing it to display a certain type of cursor. (Ex. 1004 at 5:47-57; Ex. 1003 ¶ 275.) In the case of preview cursors, the message to the OS and the functions and applications it employs includes a pointer to the location of the bitmap of the image to be displayed in the preview portion. (Ex. 1004 at 5:57-62; Ex. 1003 ¶ 276.) The image in the preview portion relates to the UI object on the screen to which the preview cursor is pointing. (Ex. 1004 at 3:32-35, 3:61-63.) For example, when the preview cursor points to a book icon, the cursor's preview portion shows graphics representing a "goods," in this

case chess pieces to represent a book about chess. (*Id.* at 3:65-4:3.)

2. Claims 39, 61, and 80 Are Rendered Obvious by *Malamud, Anthias, Nielsen, and Baker*

See claim 13.

5 G. Ground 5: Claims 1, 7, 15, 27, 33, 41, 53-54, 63, 72-73, 82 Are Rendered Obvious by *Baker and Anthias*

1. Claim 1 Is Rendered Obvious by *Baker and Anthias*

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

Baker discloses an animated character (“*cursor image*”) corresponding to the
cursor or pointer in a graphical UI (GUI). (Ex. 1007 at 13:12-13; Ex. 1003 ¶ 290.)
The user can control the cursor’s position onscreen by moving the user computer’s
15 pointing device (mouse) (*id.* at 5:18-23), and the cursor is, initially, a “standard
cursor” (*id.* at 13:22-30). The animated character adopts a different pose (“*specific
image*”) and the cursor’s hotspot occupies a different part of the character
depending on the character’s onscreen location. (Ex. 1007 at 13:47-52, 14:59-67.)
For example, the cursor’s hotspot may appear as the character’s foot or hand. (Ex.
20 1007 at 13:25-30.) Moving the cursor animates the character cursor to walk in the
selected direction. (Ex. 1007 at 13:47-52, 21:6-8, Figs. 3a-3c.) When the
character cursor is moved over a “quit” sign, the character cursor “wave[s]

goodbye, or, alternatively, wipe[s] his brow in relief.” (Ex. 1007 at 20:16-24.)

It would have been obvious to a POSITA to combine Baker and Anthias, which teaches a data processing system implemented on a client-server model. Anthias teaches a data processing system implemented on a client-server model in which an application runs on a client system and graphics data for the application is being drawn on a server system. (Ex. 1003 ¶ 291.) (See section VI.C.1, Element 1[Preamble] regarding Anthias.)

Combining Baker and Anthias, a POSITA would have recognized that Baker’s computer OS corresponds to the client “*remote user’s terminal*” of Claim 1 and the pictorial UI system in Baker corresponds to the “*server*” of Claim 1. Client-server systems interact according to a request-response model: the client sends a request to the server, which performs some action and sends a response back to the client. (Ex. 1003 ¶ 292.) To display content on the screen of a user terminal, a POSITA would understand that the OS uses functions and applications to convey commands to the display hardware. Baker’s pictorial UI system (“*server*”) transmits to the OS (“*remote user terminal*”) and the functions and applications it employs information that the functions and applications may use to modify the animated character cursor (“*cursor image*”), including the background animation.

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

Baker teaches storing information about the cursor's size and its specific position ("*cursor image data*") on the animated character (shown as the "*specific image*") in a character_model list/array ("*cursor display instruction*"). (Ex. 1007 at 23:1-29; Ex. 1003 ¶ 299.) Character_model also contains the location of where animation frames are stored ("*cursor image data*") because character_model is used to "register images of the character across animations." (Ex. 1007 at 23:24-26; Ex. 1003 ¶ 299.) Additionally, Baker teaches storing pointers to bitmap animation frames ("*cursor image data*") (Ex. 1007 at 24:33-59) and information about the modifications for each animation in the header of the animation script ("*cursor image data*"). (Ex. 1007 at 24:61-67; Ex. 1003 ¶ 299.)

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

Baker discusses OSs that employ a graphical UI and thus discloses those OS's functions and applications to display and modify graphics on the UI ("*cursor display code*"), including cursors in their initial, standard forms as well as any subsequent modifications ("*cursor display code operable to modify said cursor image*"). (Ex. 1007 at 2:32-56; Ex. 1003 ¶ 303.) Baker discloses rendering a cursor on the screen, tracking the cursor, and modifying its appearance. (Ex. 1007

at 13:47-52, Appendix E; Ex. 1003 ¶ 305.)

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

Combining Baker and Anthias, a POSITA would have recognized that the
5 computer OS in Baker corresponds to the client “remote user’s terminal” of Claim 1 and Baker’s pictorial UI system corresponds to the “server” of Claim 1.

Baker’s pictorial UI system (“*first server computer*”) transmits to an OS (“*remote user terminal’s*”) and the functions and applications employed by the OS (“*cursor display code*”) information that the functions and applications may use to
10 modify the animated character cursor (“*cursor image*”), including the background animation and the character_model list or array (collectively, “*specified content information*”). (Ex. 1003 ¶¶ 308-311.)

(See section VI.C.1, Element 1[Preamble] regarding Anthias.)

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,

15

20 When the user positions the animated character cursor (“*cursor image*”) over a portion of an icon (“*information to be displayed*”), Baker teaches requesting

information (“*specified content information*”) from the pictorial UI (“*first server computer*”), including the background animation (“*information to be displayed*”) and the character_model data structure (“*cursor display instruction*”). (Ex. 1007 at Appendix E; Ex. 1003 ¶ 315.) Baker also discloses maintaining information about the character cursor’s location (“*cursor image data*”) in character_model. (Ex. 1007 at 23:20-22; Ex. 1003 ¶ 315.) Character_model is used to “register images of the character across animations,” which means character_model necessarily contains the location of where animation frames (“*specific image*”) are stored (“*cursor display instruction indicating a location of said cursor image data*”). (Ex. 1007 at 23:24-26; Ex. 1003 ¶ 314.) The play_animation procedure, which is responsible for displaying the animated character overlaid on the background image, reads and updates character_model when modifying the animated character cursor’s location, size, and appearance. (Ex. 1007 at 23:20-22, Appendix E; Ex. 1003 ¶ 315.) Additionally, Baker teaches storing pointers to bitmap animation frames (“*cursor image data*”) and information about the modifications for each animation in the header of the animation. (Ex. 1007 at 24:33-67; Ex. 1003 ¶ 315.) Baker also teaches storing information about the cursor’s last location (“*cursor image data*”). (*Id.*)

Baker’s pictorial UI communicates with the OS by providing the information in character_model (“*cursor display instruction*”) to the OS and its

functions and applications (“*cursor display code*”) in order to execute the prologue and epilogue animations (“*cause said user terminal to display a modified cursor image...in the shape and appearance of said specific image*”). (Ex. 1007 at Abstract, 23:1-29; Ex. 1003 ¶ 316.)

5 **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**
10 **display of said user's terminal,**

Combining Baker and Anthias, a POSITA would have recognized that the computer OS in Baker corresponds to the client “remote user’s terminal” of Claim 1 and Baker’s pictorial UI system corresponds to the “server” of Claim 1. Baker’s basic processes start with input interpretation to detect meaningful input signals.

15 (*Id.* at 14:30-34, 10:53-63.) Inputs are transmitted from devices (such as “a gamepad controller, a mouse, or by using a limited number of keys on a normal keyboard” (*id.* at 10:29-31)) to the OS, then to Baker’s pictorial UI.

This process of sending inputs to the pictorial UI allows the OS (“*remote user terminal*”) to request that the pictorial UI (“*first server computer*”) send a
20 message back identifying the appropriate background animation (“*information to be displayed*”) and cursor animation, as indicated in the character_model (“*cursor*

display instruction”) (collectively, “*specified content information*”). (Ex. 1003 ¶ 321.)

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.

When the user positions a cursor so that it hovers over a portion of an icon (“*information to be displayed*”), Baker teaches requesting content information, including the background animation and the `character_model` data structure (“*cursor display instruction*”). (Ex. 1007 at Appendix E; Ex. 1003 ¶ 325.)

`Character_model` is used to track the cursor location as the animated character is moved on the pictorial UI (Ex. 1007 at 15:1-22, 23:20-29; Ex. 1003 ¶ 328.)

“Internal procedures” used to implement the system functions, including the “`process_input` procedure” that uses the cursor location in `character_model` as an input signal to determine the next animation: “For example, the `process_input` procedure processes input signals and cursor location to determine the command

code, the icon, the animation_selector, and the duration, if applicable for the next command. ...The process_input procedure takes input from...character_model ...and returns the command_code, icon, duration, and move_vector.” (Ex. 1007 at 31:56-59, 32:16-18.) In this manner, when the animated character cursor is moved
5 to a location with an associated animation (e.g., “quit” sign), the cursor location stored in character_model causes the pictorial UI to send a message to the OS and the functions and applications it employs (“*cursor display code*”) to display the correct animation. (Ex. 1003 ¶ 328; Ex. 1007 at 20:12-28.) The functions or applications then processes the information in or linked from the character_model
10 list or array and executes the prologue and epilogue animations (“*cursor display code is operable...in the shape and appearance of said specific image*”).

Baker teaches several examples of the character cursor including content corresponding to icons on the screen. The animation for the set_selection_arg command shows the character cursor picking up the selected icon, which means
15 the cursor at that time has a shape and appearance corresponding to the icon displayed on the screen (“*specific image including content corresponding to at least a portion of said information to be displayed*” and “*specific image relates to at least a portion of said information to be displayed*”). (Ex. 1007 at 16:21-23; Ex. 1003 ¶ 326.) The character_model data structure stores the list of icons collected

by the animated character in `character_model.collected_objects`. (Ex. 1007 at 23:18-29; Ex. 1003 ¶ 326.)

2. Claim 7 Is Rendered Obvious by *Baker* and *Anthias*

**Claim 7: The server system in accordance with claim 1, wherein
said specific image has a shape and appearance corresponding to
said information to be displayed on said display of said user's
terminal.**

Baker teaches several examples of the animated character cursor having a shape and appearance corresponding to icons on the screen. The animation for the `set_selection_arg` command shows the character cursor picking up the selected icon. (Ex. 1007 at 16:21-23; Ex. 1003 ¶ 332.) In this manner, the animated character cursor picking up the icon have a shape and appearance corresponding to the icon displayed on the screen.

3. Claim 15 Is Rendered Obvious by *Baker* and *Anthias*

**Claim 15: The server system in accordance with claim 1, said
cursor display instruction further comprising an image identifier
indicating said cursor image data corresponding to said specific
image.**

See claim 1. Baker discloses maintaining information about the location and position of the character cursor ("*cursor image data*") in the `character_model` data structure ("*cursor display instruction*"). (See section VI.G.1, Element 1[c][ii] regarding Baker.)

4. Claim 27 Is Rendered Obvious by *Baker and Anthias*

See claim 1. Claim 27 is substantially identical to claim 1 except that claim 27 recites modification of the cursor image in response to “movement of said cursor image over a specified location” rather than in response to “movement of said cursor image over a display of said at least a portion of said information to be displayed.” (See Ex. 1003 at ¶¶ 337-360.) Baker and Anthias satisfy claim 27 for the same reasons as claim 1.

5. Claim 33 Is Rendered Obvious by *Baker and Anthias*

See claim 7.

6. Claims 41 and 82 Are Rendered Obvious by *Baker and Anthias*

See claim 15.

7. Claim 53 Is Rendered Obvious by *Baker and Anthias*

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:

(See section VI.G.1, Element 1[Preamble], regarding Baker.)

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

Combining Baker and Anthias, a POSITA would have recognized that the computer OS in Baker corresponds to the client “remote user’s terminal” of Claim 53 and Baker’s pictorial UI corresponds to the “server” of Claim 53.

Inputs are transmitted from input devices (id. at 10:29-31)) to the OS, then to Baker's pictorial UI system. The OS ("*remote user terminal*") then requests that the pictorial UI system ("*first server computer*") send a message back identifying the appropriate background and cursor animation, as indicated in the

5 character_model (collectively, "*specified content information*"). (Ex. 1003 ¶ 355.)

(See section VI.C.1, Element 1[c][iii] regarding Anthias.)

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
10 **least one indication of cursor image data corresponding to a specific image; and**

(See section VI.G.1, Elements 1[c][ii] and [iii] regarding Baker.)

Element 53[c][i]: transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance
15 **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**
20 **display of said user's terminal, and**

(See section VI.G.1, Element 1[c][iv] regarding Baker.)

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 wherein said specific image has a shape and appearance relating to said information to be displayed.

(See section VI.G.1, Element 1[c][iv] regarding Baker.)

8. Claim 54 Is Rendered Obvious by *Baker* and *Anthias*

Claim 54: The method in accordance with claim 53, wherein said transforming further comprises executing said cursor display code so as to display said specific cursor image while at least a portion of said information to be displayed is displayed on said display of said user's terminal.

Baker teaches several examples of the character cursor having a shape and appearance corresponding to icons on the screen. The animation for the set_selection_arg command shows the character cursor picking up the selected icon. (Ex. 1007 at 16:21-23.) In this manner, the animated character cursor picking up the icon have a shape and appearance corresponding to the icon on the UI. (Ex. 1003 ¶ 378.)

9. Claim 63 Is Rendered Obvious by *Baker* and *Anthias*

Claim 63: The method in accordance with claim 53 wherein said specified content information comprises an image identifier that corresponds to the location of data representing said specific image.

(See section VI.G.1, Element 1[c][ii] regarding Baker.)

10. Claim 72 Is Rendered Obvious by *Baker and Anthias*

See claim 53. Claim 72 is substantially identical to claim 53 except that claim 72 recites modification of the cursor image in response to “movement of said cursor image over a specified location” rather than in response to “movement of said cursor image over a display of said at least a portion of said information to be displayed.” (See Ex. 1003 at ¶¶ 383-386.) Baker and Anthias satisfy claim 72 for the same reasons discussed above regarding claim 53.

11. Claim 73 Is Rendered Obvious by *Baker and Anthias*

See claim 54.

H. Ground 6: Claims 12, 14, 38, 40, 60, 62, 79, and 81 Are Rendered Obvious by *Baker, Anthias, and Nielsen*

1. Claim 12 Is Rendered Obvious by *Baker, Anthias, and Nielsen*

Claim 12: The server system in accordance with claim 1, wherein said specified content information is transmitted in the form of HTML files that define a web page.

To the extent Baker does not explicitly disclose that content information is transmitted in the form of HTML files, Nielsen teaches transmitting HTML webpages over a network from a server to a client computer. (See section VI.D.1 above regarding Nielsen.)

2. Claim 14 Is Rendered Obvious by *Baker, Anthias, and Nielsen*

Claim 14: The server system in accordance with claim 1, wherein

**said user terminal includes a browser application responsive to
said cursor display instruction, said browser application executing
said cursor display code using parameters defined in said cursor
display instruction.**

5 To the extent Baker does not explicitly disclose a browser application,
Nielsen teaches a browser that detects a cursor's movement within the image of a
client-side image map. (See section VI.D.2 above regarding Nielsen.)

3. Claims 38, 60, and 79 Are Rendered Obvious by *Baker, Anthias, and Nielsen*

10 See claim 12.

4. Claims 40, 62, and 81 Are Rendered Obvious by *Baker, Anthias, and Nielsen*

See claim 14.

VII. CONCLUSION

15 Petitioner has demonstrated a reasonable likelihood that it will prevail in its
challenge of patentability for the Challenged Claims and respectfully requests,
therefore, that a trial on the '449 Patent be instituted.

Dated: September 18, 2018

Respectfully submitted,

/James Valentine/

Lead Counsel

James F. Valentine, Reg. No. 39,053

PERKINS COIE LLP
3150 Porter Drive
Palo Alto, California 94304
(650) 838-4300 (phone)
(650) 838-4350 (fax)

Backup Counsel

Daniel T. Shvodian, Reg. No. 42,148

Attorneys for Petitioner

Ralph Lauren Corporation

CERTIFICATE OF WORD COUNT UNDER 37 CFR § 42.24(d)

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,971, excluding the parts exempted by 37 C.F.R. § 42.24(a). This word count was made by using the built-in word count function tool in the Microsoft Word software Version 2016 used to prepare the document.

Dated: September 18, 2018

Respectfully submitted,

/James Valentine/

Lead Counsel
James F. Valentine, Reg. No. 39,053

PERKINS COIE LLP
3150 Porter Drive
Palo Alto, California 94304
(650) 838-4300 (phone)
(650) 838-4350 (fax)

Backup Counsel
Daniel T. Shvodian, Reg. No. 42,148

Attorneys for Petitioner
Ralph Lauren Corporation

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of the foregoing
PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 6,118,449 and
supporting materials (Exhibits 1001-1015 and Power of Attorney) have been
served in their entirety this 18th day of September, 2018, by FEDERAL EXPRESS
on Patent Owner at the correspondence address for the attorney of record for
the '449 Patent shown in USPTO PAIR:

Robert Haroun and Joseph Sofer
Weingarten Schurgin Gagnebin & Hayes LLP
Ten Post Office Square
Boston, MA 02109

and the attorneys of record for Plaintiff in the concurrent litigation matter:

Stamatios Stamoulis (stamoulis@swdelaw.com)
Richard Charles Weinblatt (weinblatt@swdelaw.com)

Dated: September 18, 2018

Respectfully submitted,

/James Valentine/

Lead Counsel
James F. Valentine, Reg. No. 39,053

PERKINS COIE LLP
3150 Porter Drive
Palo Alto, California 94304
(650) 838-4300 (phone)
(650) 838-4350 (fax)

Back-Up Counsel
Daniel T. Shvodian, Reg. No. 42,148

Attorneys for Petitioner
Ralph Lauren Corporation