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. IPR2018-01749

Patent No. 5,995,102

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 5,995,102

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

September 18, 2018

TABLE OF CONTENTS

Page

EXHIBIT LIST vii

I. INTRODUCTION 1

II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(B) 1

A. Real Party in Interest 1

B. Related Matters 1

C. Notice of Counsel and Service Information 3

III. REQUIREMENTS FOR *INTER PARTES* REVIEW 3

A. Standing 3

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 4

5. Statutory Grounds 4

6. How Claims Are Unpatentable 5

IV. OVERVIEW OF THE '102 PATENT 5

A. Priority Date 5

B. State of the Art Before the '102 Patent 5

C. Summary of the '102 Patent 8

D. Level of Ordinary Skill in the Art 10

V. PROPOSED CLAIM CONSTRUCTIONS 11

A. "cursor image" / "initial cursor image" (all Challenged Claims) 12

B. "specific image" / "specific cursor image" (all Challenged Claims) 13

C. "modifying a cursor image" / "modified cursor image" (all Challenged Claims) 14

D. "cursor image data" (all Challenged Claims).....15

E. "cursor display code" (all Challenged Claims).....17

F. "cursor display instruction" (all Challenged Claims) 17

VI. THERE IS A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE '102 PATENT IS UNPATENTABLE 19

A. Identification and Overview of Key Prior Art References 19

1. *Malamud* (Ex. 1004) 19

2. *Anthias* (Ex. 1005) 22

3. *Baker* (Ex. 1007).....24

B. Motivation to Combine References 27

C. Ground 1: Claims 70-73 Are Rendered Obvious by *Malamud* and *Anthias*. 31

1. Claim 70 Is Rendered Obvious by *Malamud* and *Anthias* 31

2. Claim 71 Is Rendered Obvious by *Malamud* and *Anthias* 44

3. Claim 72 Is Rendered Obvious by *Malamud* and *Anthias* 44

4. Claim 73 Is Rendered Obvious by *Malamud* and *Anthias* 49

D. Ground 2: Claims 70-73 Are Rendered Obvious by *Baker* and *Anthias*. 49

1. Claim 70 Is Rendered Obvious by *Baker* and *Anthias* 49

2. Claim 71 Is Rendered Obvious by *Baker* and *Anthias* 59

3. Claim 72 Is Rendered Obvious by *Baker* and *Anthias* 59

4. Claim 73 Is Rendered Obvious by *Baker* and *Anthias* 61

VII. CONCLUSION.....61

CERTIFICATE OF WORD COUNT UNDER 37 CFR § 42.24(d).....63

CERTIFICATE OF SERVICE 64

TABLE OF AUTHORITIES

Page(s)

CASES

Belden Inc. v. Berk-Tek LLC,
805 F.3d 1064 (Fed. Cir. 2015) 28

Black & Decker, Inc. v. Positec USA, Inc.,
646 Fed. App'x 1019 (Fed. Cir. 2016) 11

CCS Fitness, Inc. v. Brunswick Corp.,
288 F.3d 1359 (Fed. Cir. 2002) 12

ClassCo v. Apple, Inc.,
838 F.3d 1214 (Fed. Cir. 2016) 28

Graham v. John Deere Co.,
383 U.S. 1 (1966).....27

In re Rambus Inc.,
694 F.3d 42 (Fed. Cir. 2012) 11

KSR Int'l Co. v. Teleflex Inc.,
550 U.S. 398 (2007).....27, 28, 29, 30

Lexos Media IP, LLC v. APMEX, Inc.,
No. 2:16-cv-00747-JRG-RSP, Early Claim Construction Opinion
and Order, Dkt. 86 (E.D. Tex., Mar. 16, 2017) (Ex. 1010) 3, 12

Lexos Media IP, LLC v. Ralph Lauren Corporation et al.,
No. 1:17-cv-01319-LPS (D. Del.) 1, 2

Perfect Web Techs., Inc. v. InfoUSA, Inc.,
587 F.3d 1324 (Fed. Cir. 2009) 28

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) (en banc) 11, 12

STATUTES

35 U.S.C. § 102.....4, 19, 23, 24

35 U.S.C. § 103 ..... 1, 4, 19, 27

35 U.S.C. § 311 ..... 1, 4

35 U.S.C. § 312 4

35 U.S.C. § 313 4

35 U.S.C. § 314 4

35 U.S.C. § 315 ..... 3, 4

35 U.S.C. § 316 4

35 U.S.C. § 317 4

35 U.S.C. § 318 4

35 U.S.C. § 319 4

35 U.S.C. § 325 4

OTHER AUTHORITIES

37 C.F.R. § 42.8 ..... 1, 3

37 C.F.R. § 42.10 3

37 C.F.R. § 42.15 ..... 3, 4

37 C.F.R. § 42.22 4

37 C.F.R. § 42.24 1

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 ..... 3, 4, 5

37 C.F.R. § 42.105 3

37 C.F.R. § 42.106 3

EXHIBIT LIST[1]

Ex. 1001	U.S. Patent No. 5,995,102 entitled *Server System and Method for Modifying a Cursor Image* to James Samuel Rosen et al. ("'102 Patent")
Ex. 1002	U.S. Patent No. 6,118,449 entitled *Server System and Method for Modifying a Cursor Image* to James Samuel Rosen et al. ("'449 Patent")
Ex. 1003	Declaration of Dr. Benjamin B. Bederson Regarding the '102 and '449 Patents, and Appendices
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	*Lexos Media IP, LLC v. APMEX, Inc.*, No. 2:16-cv-00747-JRG-RSP, Early Claim Construction Opinion and Order, Dkt. 86 (E.D. Tex., Mar. 16, 2017).

[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'449 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. 1011 File History of the ’102 Patent

Ex. 1012 File History of the ’449 Patent

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 70-73 ("Challenged Claims") of U.S. Patent No. 5,995,102 (Ex. 1001).

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)	Claims 70-73	*Malamud and Anthias*
	Claims 70-73	*Baker and Anthias*

The Board should accordingly institute a trial and cancel the Challenged Claims.

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 '102 Patent and U.S. Patent No. 6,118,449 ("the '449 Patent") (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 petition for IPR of 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. 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 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 image representative of at least a portion of the subject or topic being displayed on

[1] Lexos alleges in the District Court that it is the owner of the '102 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).

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

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 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

This Petition complies with all statutory requirements and requirements under 37 C.F.R. §§ 42.104, 42.105, and 42.15 and thus 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 '102 Patent is available for IPR and the 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

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

1. Claims Challenged

Claims 70-73 of the '102 Patent are challenged in this Petition.

2. The Prior Art

The prior art references are in the Exhibit List: Malamud (Ex. 1004), Anthias (Ex. 1005), and Baker (Ex. 1007).

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

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 evidence supporting the Petition 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 '102 PATENT

A. Priority Date

The '102 Patent issued on November 30, 1999 from U.S. Patent Application No. 08/882,580 filed on June 25, 1997. 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 the application leading up to the '102 Patent.

Additionally, U.S. Patent Nos. 7,111,254 ("the '254 Patent"), 6,065,057 ("the '057 Patent"), and 7,975,241 ("the '241 Patent") claim priority to the application leading up to the '102 Patent. The '254 Patent issued on September 19, 2006 and is entitled "System for Replacing a Cursor Image in Connection with Displaying the Contents of a Web Page." The '057 Patent issued on May 16, 2000 and is entitled "Method for Authenticating Modification of a Cursor Image." The '241 Patent issued on July 5, 2011 and is entitled "System for Replacing a Cursor Image in Connection with Displaying the Contents of a Web Page."

B. State of the Art Before the '102 Patent

A graphical user interface ("GUI"), first demonstrated by Dougless

Engelbart in 1968, is where interactions between humans and computers occur. (Ex. 1003 at ¶ 19.) A well-known example of a GUI uses the "desktop metaphor," where graphical icons represent computer files and applications on a virtual desktop and where the user interacts with those icons using a pointing device (e.g., mouse, touchpad, stylus pen). (*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*. at ¶ 20.) Specifically, the OS uses "driver" programs dedicated to control devices attached to the computer. The driver provides a software interface to hardware devices. (*Id*.) A "display driver" accepts commands from the OS and generates signals to the display device to render the desired text or image, including GUI elements, on the display device's screen. Display drivers were only one way that OSs controlled system displays. OSs also used internal function and other applications to display images. (*Id*.)

When a user moves the mouse, an image called a "cursor" moves correspondingly onscreen. (*Id*. at ¶ 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. (*Id*.)

While 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 application windows, fonts, and cursors' image files by sending a message to the OS, including specific parameters related to the application's request. Message parameters can contain integers, bit flags, or pointers to additional data. (*Id*.)

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

The client/server architecture is fundamental to computing, and downloading content from servers, including graphical information, has been well-known for decades. (*Id*. at ¶ 27.) Client/Server systems are at the heart of the World Wide Web 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 along with 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 to modify a cursor's image file. (*Id*. ¶¶ 30-31.) This approach was built into the very widely-used "X Windows" system, which had a separate graphics "server" to provide graphics and input 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 graphics server. This is true whether the information to be displayed is text, graphics, or custom cursor images, which the X Windows system supported. (*Id*. at ¶ 33)

C. Summary of the '102 Patent

The '102 Patent is directed toward online advertising, using a cursor that changes according to onscreen content. The invention operates over a client-server network and in HTML documents (i.e., webpages). These were all concepts well-known in the prior art.

The '102 Patent discloses "[a] system for modifying a cursor image … to a specific image having a desired shape and appearance" (Ex. 1001 at Abstract) where the cursor image is an image or "icon[] which represent[s] the cursor or pointer as it may appear in some cases" (*id.* at 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 annoyance of totally interrupting and intrusive content delivery, and 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 or logo, a brand logo, an advertising or marketing icon or slogan, or animated advertising 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 (*id*. at 17:33-34); a pink cursor on a website about Pink Panther (*id*. at 17:34-35); a witch on a broomstick for Halloween (*id*. at 17:35-36); and the Statue of Liberty for Fourth of July (*id*. at 17:36-37). The '102 Patent teaches that it was known in the art for pointers and cursors to change shape. (*Id*. at 3:45-49.)

Figures 7 and 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 a bottle (designated "44a" and with red highlight added) to advertise a cola drink:

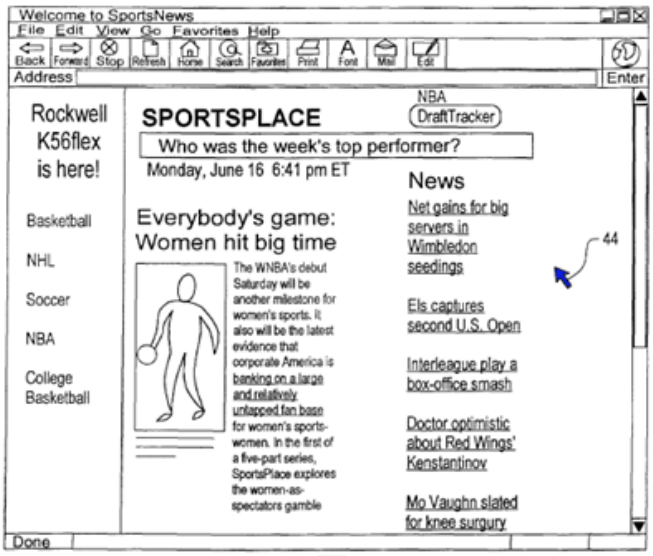

FIG. 7

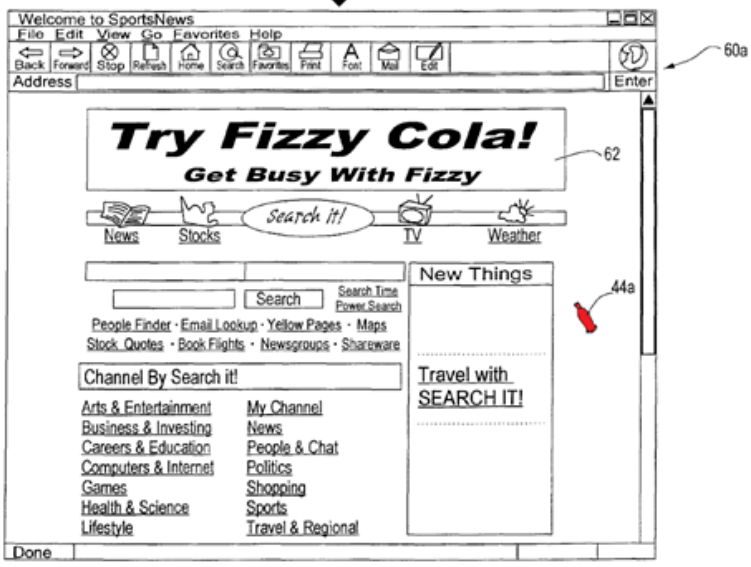

FIG. 8

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 user interface ("UI") design and operating systems ("OSs"). He or she 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 he 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 '102 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 '102 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 certain claim terms below pursuant to *Phillips*.

A. "cursor image" / "initial cursor image" (all Challenged Claims)

"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, taking into account the claim language, specification, and prosecution history. (Ex. 1003 ¶ 55.) This construction is consistent with how the '102 Patent uses the phrase to describe ways to modify a generic cursor (black arrow, pointing finger, spinning wheels, hourglasses, and wristwatches): "[The purported invention] can be used to replace not just the standard arrow but other standard cursors as well, such as the generic

hand with pointing index finger." (Ex. 1001 at 17:41-45; *see* Ex. 1001 at 3:57-64, Figs. 7-9; Ex. 1003 ¶ 56.) The modified cursor can be reverted back 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.) The user can also click or double-click the pointing device to activate a screen or task or launch an application or other function. (*Id.*) A user terminal's OS uses "drivers" to access peripheral devices that are part of the user terminal, including a display driver that controls and 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's functions also control the behavior and appearance of the cursor. (Ex. 1001 at 8:34-37.)

B. "specific image" / "specific cursor image" (all Challenged Claims)

"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, taking into account 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 displayed on a video monitor of a remote user's terminal." (Ex. 1001 at 2:40-43, Figs. 7-9.) 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 "Fizzy Cola" bottle, "Fizzy Cola's logo, its corporate mascot, images of its products or services, slogans, icons, brand images, advertising messages (the word "Thirsty?", for example), abstract suggestions (such as a straw or glass), etc."; a baseball bat; a pink but otherwise standard-shaped pointer; 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 "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.)

C. "modifying a cursor image" / "modified cursor image" (all Challenged Claims)

"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, taking into account

the claim language, specification, and prosecution history. (Ex. 1003 ¶ 66.) Each 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 appearance." (Ex. 1001 at Abstract.) 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.)

D. "cursor image data" (all Challenged Claims)

"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, taking into account the claim language, specification, and prosecution history. (Ex. 1003 ¶ 70.) "Cursor image data" appears in every independent claim and in 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 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).

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 (Ex. 1001 at 10:47-51; Ex. 1003 ¶ 72), retrieves data corresponding to the specific image from local memory or a remote server. (Ex. 1001 at 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" (all Challenged Claims)

"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, taking into account 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 displayed. (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 to control drivers (e.g., display driver) that provide the OS access to the displayed cursor image. (Ex. 1001 at 8:59-64; Ex. 1003 ¶ 75.) 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.) The display drivers use the cursor information to modify the cursor displayed. (Ex. 1001 at 9:12-20; Ex. 1003 ¶ 75.)

F. "cursor display instruction" (all Challenged Claims)

"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, taking into account the claim language, specification, and prosecution history. (Ex. 1003 ¶ 77.) The cursor display instruction includes information that the cursor display code uses to control drivers (e.g., display driver) 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 at 11:47-49; Ex. 1003 ¶ 78.)

Information for changing the cursor's appearance ("cursor display instructions") were 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, specification of type of modification intended, etc. (Ex. 1001 at 9:36-6, Fig. 4; Ex. 1003 ¶ 78.)

VI. THERE IS A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE '102 PATENT IS UNPATENTABLE

The Challenged Claims of the '102 Patent are directed toward online advertising using concepts well known in the prior art, such as a cursor that changes according to onscreen content transmitted over a client-server network and in HTML documents (i.e., webpages). 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. Combined, these three patents teach the Challenged Claims. Malamud, Anthias, and Baker 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). Originally assigned to Microsoft Corporation, Malamud was not presented to or considered by the PTO during prosecution of the application leading up to the issuance of the Lexos Patents.

Malamud discloses "information cursors" for use in OSs or application programs. (Ex. 1004 at Abstract.)[3] Each information cursor includes a pointing portion and an information portion to display information about an object to which the pointing portion points. (*Id*.) The information displayed 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. (*Id*. at 2:22-24.) Objects, including information cursors, are displayed on the video display of a data processing system having an input device (e.g., mouse). (*Id*. at 1:35-37.) The information cursor's position on the UI changes in response to a user moving the input device. (*Id*. at 2:20-22.)

Malamud explains that information cursors may be implemented on and by OSs such as Microsoft Windows 3.1. (*Id*. at 3:6-8, 4:34-37.) Each program running on the OS may have one or more associated windows. (*Id.* at 4:57-59.) The OS logically divides the UI into a number of program windows, and each window has an associated window 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 message queue

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

for the program. (*Id*. at 4:57-61.) The program retrieves and delivers the message to the proper window procedure by executing a block of code known as the "message loop." (*Id*. at 4:61-63.) The window procedure that receives the message then processes the message. (*Id*. at 4:63-65.) When a user positions the information cursor over an object within a program's window, a message specifying the cursor's position within the window 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 mouse activities (*id*. at 5:7-10), including determining whether the cursor is pointing go an object and, if so, modifying the information cursor to display in the cursor's information portion information about the object (*id*. at 5:27-39).

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

the 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 property information regarding the object. (*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.) A preview cursor is shown in Figure 3 below, and a combined name and preview cursor is shown in Figure 4.

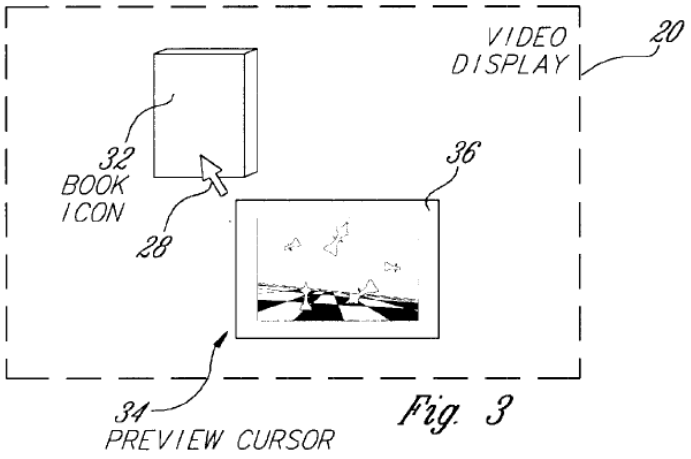

Fig. 3

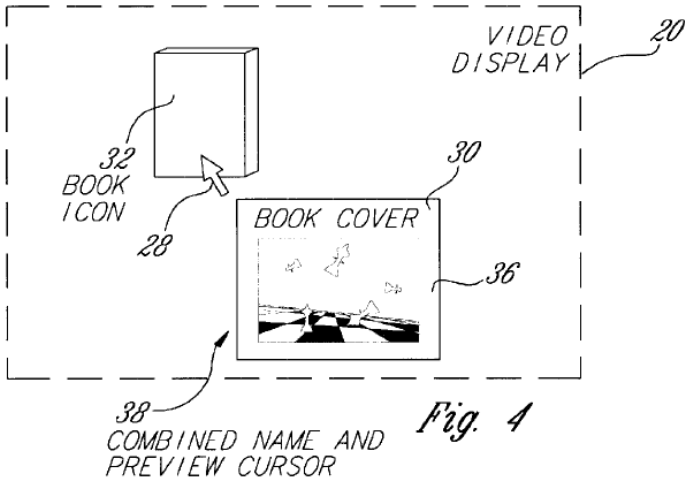

Fig. 4

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 how the cursor appears 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). Originally assigned to IBM Corporation, Anthias was not presented to or considered by the PTO during prosecution of the application leading up to the issuance of the Lexos Patents.

Anthias teaches a distributed data processing system in which a "client" runs an application and then uses its presentation system to transmit to the "server" only the 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 similarly in X Windows as discussed 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 and 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 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 (or color or flashing frequency) as it passes from background window areas to the device area. (*Id*. at 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. *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. § 102(e)(2). Originally assigned to the inventor and then to IBM, Baker was not presented to or considered by the PTO during prosecution of the application leading up to the issuance of the Lexos Patents.

Baker teaches a pictorial or graphical UI for accessing information in an electronic file system. (Ex. 1007 at Abstract.) In particular, Baker discloses "a user definable graphical interface for a computer operating system which utilizes pictorial information and animation as well as sound." (*Id*. at 1:11-13.) Icons in the pictorial background image are linked to file objects, and an animated character is overlaid on the background image. (*Id*.) User input causes the animated character to move, including to different positions on the screen 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 served as his own lexicographer by defining "cursor" as used in Baker 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." In Baker, the hotspot 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 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 (*id*. at 23:20-26), and is updated by the play_animation function to match the last frame of each animation played (*id*. at 23:22-24). Character_model is used to calibrate animations and register images of the character across animations (*id*. at 23:24-26), and maintain the list of references to file objects (*id*. at 23:26-29). Additionally, Baker teaches storing pointers to bitmap animation frames (*id.* at

24:33-59) and information about the modifications for each animation in the header of the animation (*id.* at 24:61-67).

Baker also teaches that the character cursor may be animated to "jump" into an icon that references 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, whereby the character is animated to hold the icon until a reset_selection_arg command is issued. (*Id*. at 16:21-27.)

The basic processes of the pictorial UI in Baker are grouped around three functional tasks: input interpretation, basic execution unit, and system control. (*Id*. at 14:30-34.) Input interpretation includes cursor control; basic execution unit includes prologue animation (signaling that the pictorial UI received the user's commands), a command execution, and an epilogue animation (signaling that the animation has completed); and system control controls execution of input interpretation and the basic execution unit. (*Id.* at 14:30-38, 10:62-11:1.)

When a meaningful input signal is interpreted, a basic execution unit (including an executable command and a set of animations) is accessed. (*Id.* at 10:53-63.) While the pictorial 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 at ¶ 109.) The animation frames that the pictorial UI of Baker displays must therefore be identified or transmitted to those functions and application for display. (*Id.* at ¶ 109.)

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 at the time the invention was made to a POSITA. 35 U.S.C. § 103. Obviousness is a question of law based on underlying facts including: (1) "the scope and content of the prior art," (2) the "differences between the prior art and the claims at issue," (3) "the level of ordinary skill in the pertinent art," and (4) any "secondary considerations," or objective indicia, of nonobviousness. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007); *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). 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*, 550 U.S. at 418-21. 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. InfoUSA, Inc.*, 587 F.3d 1324, 1329 (Fed. Cir. 2009).

A POSITA, who is "a person of ordinary creativity, not an automaton," would have known to combine the teachings in Malamud, Anthias, and Baker in the manner claimed by the Challenged Claims. *ClassCo v. Apple, Inc.*, 838 F.3d 1214, 1219 (Fed. Cir. 2016) (citing *KSR*, 550 U.S. at 421); *Belden Inc. v. Berk-Tek LLC*, 805 F.3d 1064, 1074-75 (Fed. Cir. 2015).

Malamud, Anthias, 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 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.) 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, which discloses 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.) 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 defines a device screen to request 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 requirements and processing overhead in the server computer. (Ex. 1005 at 2:30-34; Ex. 1003 ¶ 112.) A POSITA would have recognized that the client-server architecture of Anthias could be combined with Malamud and Baker

to customize cursors for websites. (Ex. 1003 ¶ 112.)

Malamud, Anthias, 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 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, 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 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) 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 documents viewable via the Internet (i.e., webpages and websites). (Ex. 1003 ¶ 112.)

C. Ground 1: Claims 70-73 Are Rendered Obvious by *Malamud* and *Anthias.*

1. Claim 70 Is Rendered Obvious by *Malamud and Anthias*

Element 70[Preamble]: A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

Malamud teaches an "information cursor" that includes a pointing portion and an information portion. (Ex. 1004 at Abstract; Ex. 1003 ¶ 116.) The information cursor in Malamud is "generally manipulable by an input device, such as a keyboard or a mouse" (*id*. at 1:16-18), and its pointing portion is a standard/generic cursor (e.g., arrow pointer) (*id*. at Figs. 2a-5, 5:66-6:17 ("conventional cursor")). When its pointing portion points to an object displayed on the screen, the information cursor is modified to display its information portion ("*modifying a cursor image to a specific image having a desired shape and appearance*"[4]), as drawn by the computer OS, which a POSITA would understand is commonly handled by a function or application employed by the OS ("*displayed on a display of a remote user's terminal*"). (Ex. 1004 at 3:54-56; Ex. 1003 ¶¶ 116, 414-416.) In the cases of the preview cursor and the combined name and preview cursor types of information cursor, the cursor's preview portion displays a

[4] Hereinafter, language from the Challenged Claims will be set off by italicization.

graphical preview of the contents of an object. (Ex. 1004 at 3:59-4:18, Figs. 3 and 4.)[5]

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, 414-416.) 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, 414-416.)

Combining Malamud and Anthias, a POSITA would have recognized that the computer OS in Malamud corresponds to the client "*remote user's terminal*" of

[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 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 '102 Patent applies similarly to Malamud's disclosure of a combined name and preview cursor.

Claim 1 and the application program's window procedure in Malamud corresponds to the "*server*" of Claim 1. (Ex. 1003 ¶¶ 118, 414-416.) 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.

Malamud teaches that an OS divides the computer's UI into program windows, each having an associated window procedure. (Ex.1004 at 3:53-55; Ex. 1003 ¶¶ 118, 414-416.) As shown in Fig. 6 below, when the cursor is moved into the window of an application program, the OS ("*remote user terminal*") sends a message (specifying the position of the cursor) to the application program, which forwards the message to the window procedure ("*server*") associated with the window. (*Id*. at 5:24-28; Ex. 1003 ¶¶ 118, 414-416.) The window procedure then determines whether the position corresponds to the position of an object and, 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)." (*Id*. at 5:28-53.)

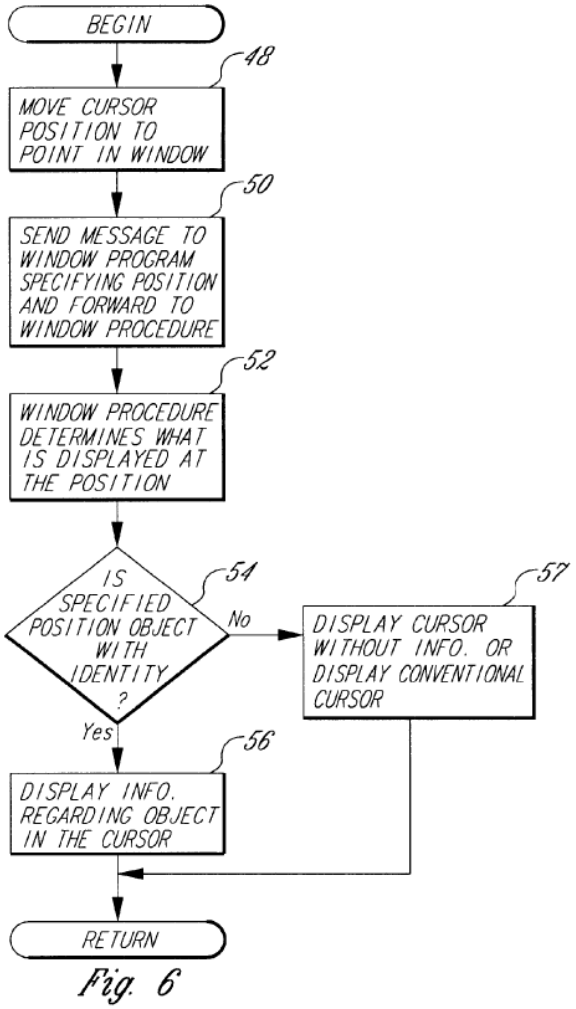

Fig. 6

Malamud teaches four types of information cursors: name cursors, preview cursors, combined name and preview cursors, and property cursors. (*Id*. at 3:10-12.) All four types of information cursors have a pointing portion, and each type of information cursor has an information portion that becomes visible to the user when the cursor's pointing portion points at an object onscreen. The name cursor has a name box that displays the name of the object. (*Id*. at 3:31-34, 3:54-56.) The preview cursor has a preview portion that displays a graphical preview of the contents of the object. (*Id*. at 3:61-63, Fig. 3.) The property cursor has a property

box that displays property information regarding the object. (*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.) A preview cursor is shown in Figure 3 below, and a combined name and preview cursor is shown in Figure 4.

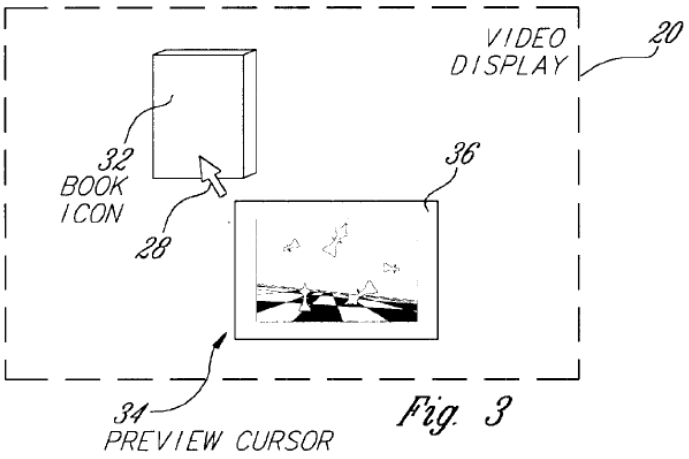

Fig. 3

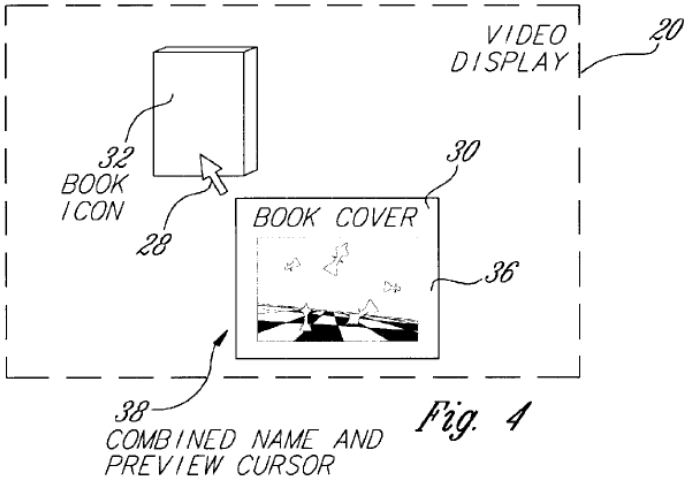

Fig. 4

Element 70[a]: cursor image data corresponding to said specific image;

Malamud teaches that 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, 414-416.) The OS stores bitmaps of images ("*specific*

image") that are to be displayed in the preview portion and other information corresponding to the preview cursor ("*cursor image data corresponding to said specific image*"). (Ex. 1004 at 5:16-18, 5:59-62; Ex. 1003 ¶¶122, 414-416.) 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, 414-416.) 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 a function or application employed 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, 414-416.)

Element 70[b]: cursor display code, said cursor display code operably to modify said cursor image; and

Malamud teaches the use of information cursors in conjunction with a conventional OS (such as Microsoft Windows), which a POSITA would understand uses functions or 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 ("*said 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, 414-416.) The OS divides the UI into a number of program windows, each having

an associated window procedure. (Ex. 1004 at 3:53-55; Ex. 1003 ¶¶ 126, 414-416.) When the cursor is moved into the window of an application program, the OS sends a message (specifying the position of the cursor) to the application program, which forwards the message to the window procedure ("*first server computer*") associated with the window. (Ex. 1004 at 5:24-28; Ex. 1003 ¶¶ 126, 414-416.) 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, 414-416.) The OS then employs a function or application ("*cursor display code*") to modify the information cursor so as to display its preview portion. (Ex. 1003 ¶¶ 126, 414-416.)

Element 70[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. (Ex. 1003 ¶¶ 114-119, 129, 414-416.) Malamud discloses that an application program's window procedure ("*first server computer*") transmits a message ("*specified content information*") to the OS ("*remote user*

terminal") and the functions and applications it employs ("*cursor display code*") to display its information portion. (Ex. 1004 at 4:53-54, 5:53-57; Ex. 1003 ¶¶ 130, 414-416.) (See section VI.C.1, Element 70[Preamble], above regarding Anthias.)

Element 70[c][ii]: said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image,

Malamud, 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. (Ex. 1003 ¶¶ 114-119, 133, 414-416.) 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, to instruct the OS 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, 414-416.)

The OS maintains a message queue for each application program, and when an event occurs, the event is translated into a message that is put in the application program's message queue. (Ex. 1004 at 4:57-61; Ex. 1003 ¶¶ 134, 414-416.) The

application program forwards the message to the proper window procedure by executing a block of code known as the "message loop." (Ex. 1004 at 4:61-63; Ex. 1003 ¶¶ 134, 414-416.) When a user positions the cursor over an object within a program's window, a message specifying the cursor's position within the window is transmitted from the OS to the application program's message queue, then to the application program, and on to the program window's window procedure. (*Id*. at 4:66-5:14; Ex. 1003 ¶¶ 134, 414-416.) The window procedure uses the information in the message to respond to the cursor's position and other mouse activities (Ex. 1004 at 5:7-10), including determining whether an object is present where the cursor is pointing (Ex. 1004 at 5:27-39). Once the window procedure identifies the object at the specified cursor position, the window procedure sends a message back to tell the OS what type of cursor to display and identify the content and type of information to be displayed in the cursor. (*Id*. at 5:46-52; Ex. 1003 ¶¶ 134, 414-416.)

In the case of preview cursors, 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 indicating location of said cursor image data*"), the OS and the functions and applications it employs ("*cursor display code*") interfaces with the operating system to display the bitmap image in

the preview cursor's preview portion ("*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*"). (Ex. 1004 at 5:57-62; ¶¶ 136, 414-416.) For example, when the preview cursor points to an icon for a book about chess that is displayed on the screen, the preview portion of the cursor displays a graphical scene of chess piece shapes flying through the air ("*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*"). (Ex. 1004 at 3:32-35, 3:59-4:3, Fig. 3-4; ¶¶ 136, 414-416.)

Element 70[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*"), which a POSITA would understand sends to the functions and applications the OS employs ("cursor display code") for displaying the information cursor's information portion on the screen. (Ex. 1004 at 4:53-54, 5:53-57; Ex. 1003 ¶¶ 140, 414-416.) Malamud also discloses

that modifying an information cursor to display its information portion initiates when "the cursor position is moved by the mouse 18 or other input device to point within the window," and the OS generates and sends a message to the application program which forwards the message (with new cursor position) to the application program's window procedure. (Ex. 1004 at 5:22-28; Ex. 1003 ¶¶ 140, 414-416.) 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, 414-416.) This process of sending new cursor positions to the application program's window procedure allows the OS ("*remote user terminal*") to request that the application 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; Ex. 1003 ¶¶ 140, 414-416.) Malamud teaches that 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 that also includes the name of the object ("*said specified content information further comprises information to be displayed on said display of said user's terminal*"). (Ex. 1004 at 4:4-17; Ex. 1003 ¶¶ 140, 414-416.)

(See section VI.C.1, Element 70[Preamble], above regarding Anthias.

Element 70[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 responsive to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal.

Malamud, in view of Anthias, discloses an application program's window procedure ("*first server computer*") and an OS ("*remote user's terminal*") on a client-server network. Malamud teaches the use of information cursors in conjunction with a conventional OS (such as Microsoft Windows), which a POSITA would understand uses functions or applications to display and modify graphics on the UI ("*cursor display code*") for rendering graphics, including cursors in their initial, standard forms as well as any subsequent modifications ("*said 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*"). (Ex. 1003 ¶¶ 144-145, 414-416.) (See section VI.C.1, Element 70[b], above regarding Malamud.

In the case of preview cursors, the message ("*specified content information*") containing a pointer to the bitmap ("*cursor display instruction*") is transmitted by the window procedure ("*server*") to the operating system ("*remote terminal*") and

the functions and applications it employs to display the information portion of an information cursor. The message ("*specified content information*") is processed by those functions and applications ("*cursor display code*") which then display the preview portion of the preview cursor ("*modify said cursor image to said cursor image in the shape and appearance of said specific image*"). (Ex. 1004 at 5:59-62; Ex. 1003 ¶¶ 145, 414-416.) Receipt of the message containing a pointer to the bitmap causes the OS to invoke a function or application for display, which then retrieves the bitmap for display on the user screen: "[I]if 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 operating system]. 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.) Because a POSITA would understand that the OS employs functions and applications to display images, including cursors, the message sent with the pointer to the bitmap image is used by those functions and applications to display the preview portion of the preview cursor. (Ex. 1003 ¶¶ 146, 414-416.)

Malamud discloses that when a preview cursor points to an object displayed on the screen, the cursor is modified so that the preview cursor's preview portion appears displaying graphical data depicting the contents of the object. (Ex. 1004 at 3:32-35, 3:61-63, Figs. 3-4.) (See, for example, section VI.C.1, Element 70[c][ii],

above regarding Malamud.)

Thus, claim 70 is rendered obvious by Malamud and Anthias.

2. Claim 71 Is Rendered Obvious by *Malamud* and *Anthias*

See claim 70. Claim 71 is substantially identical to claim 70 except that claim 71 recites modification of the cursor image "responsive to movement of said cursor image over a specified location on said display of said user's terminal" rather than "responsive to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal." Malamud and Anthias satisfy claim 71 for the same reasons discussed above regarding claim 70. (Ex. 1003 ¶¶ 417-420.)

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

Element 72[Preamble]: A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

Malamud, in view of Anthias, discloses an application program's window procedure ("*server*") and an OS ("*user terminal*") on a client-server network. (See, for example, section VI.C.1, Element 70[Preamble], above regarding Malamud.) 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; Ex. 1003 ¶¶ 167, 421-423.) The information cursor in Malamud is a "generally manipulable by an

input device, such as a keyboard or a mouse" (*id*. at 1:16-18), and its pointing portion is a standard/generic cursor (e.g., arrow pointer) (*id*. at Figs. 2a-5, 5:66-6:17 ("conventional cursor")) (Ex. 1003 ¶¶ 167, 421-423.). Each time that the information cursor's pointing portion points at an object, the information cursor is modified to display its information portion on the computer screen ("*modifying a cursor image to a specific image having a desired shape and appearance*"), as drawn by the computer's OS, which a POSITA would understand is commonly handled by a function or application employed by the OS, only when the information cursor is positioned to point to an object displayed on the screen. (Ex. 1004 at 3:54-56; Ex. 1003 ¶¶ 167, 421-423.) The contents of the information portion may include the name of the object, a graphical preview of the contents of the object, or property information about the object. (Ex. 1004 at Abstract.) In the case of the preview cursor and the combined name and preview cursor types of information cursor, the cursor includes a pointing portion and a preview portion (the information portion that shows a graphical preview of the contents of an object). (Ex. 1004 at 3:59-4:18, Figs. 3 and 4.)

Element 72[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 72.

Malamud discloses that an application program's window procedure ("*first server computer*") transmits a message ("*specified content information*") to the OS ("*remote user terminal*") and the functions and applications it 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 ¶¶ 172, 421-423.)

(See section VI.C.1, Element 70[c][iii], above regarding Anthias.)

Element 72[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 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*") for displaying the cursor's information portion on the screen. (See section VI.C.1, Elements 70[c][ii] and [iii], above regarding Malamud.)

Element 72[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

Malamud teaches that information cursors are invoked when an application's window procedure sends a message ("*specified content information*") to the OS and the functions and applications it employs ("*cursor display code*") that "tells the operating system what type of cursor to display…." when pointed to an object displayed in a window on the screen. (Ex. 1004 at 5:47-53; Ex. 1003 ¶¶ 172, 421-423.) (See section VI.C.1, Element 70[c][ii], above regarding Malamud.)

Element 72[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 responsive 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.

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 72. (Ex. 1003 ¶¶ 180, 421-423.)

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 the position of the cursor) to the application program, which forwards the message to the window procedure associated with the window. Malamud teaches that information cursors are displayed 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, to instruct the 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. 60; Ex. 1003 ¶¶ 181, 421-423.)

For preview cursors, 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 uses functions and applications ("*cursor display code*") to display the bitmap image in the preview cursor's preview portion. (Ex. 1004 at 5:57-62; Ex. 1003 ¶¶ 182, 421-423.) Malamud teaches, in the case of a combined name and preview cursor, the message includes the name of the object to which the cursor points ("*said specified content information further comprises information to be displayed on said display of said user's terminal*"). (Ex. 1004 at 4:4-17, 5:54-62.)

(See section VI.C.1, Elements 70[b] and 70[c][ii], above regarding Malamud.)

Thus, claim 72 is rendered obvious by Malamud and Anthias.

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

See claim 72. Claim 73 is substantially identical to claim 72 except that claim 73 recites modification of the cursor image "responsive to movement of said cursor image over a specified location on said display of said user's terminal" rather than "responsive to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal." Malamud and Anthias satisfy claim 73 for the same reasons discussed above regarding claim 72. (Ex. 1003 ¶¶ 424-427.)

D. Ground 2: Claims 70-73 Are Rendered Obvious by *Baker* and *Anthias.*

1. Claim 70 Is Rendered Obvious by *Baker and Anthias*

Element 70[Preamble]: A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

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, 429.) The user can control the position of the cursor onscreen by moving the user computer's pointing device (mouse) (*id*. at 5:18-23), and the cursor is, initially, a

"standard cursor" (*id*. at 13:22-30). (Ex. 1003 ¶¶ 290, 429.) 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 location on the screen. (Ex. 1007 at 13:47-52, 14:59-67.) For example, the cursor's hotspot may appear as the foot or hand of the character. (Ex. 1007 at 13:25-30.) Moving the cursor animates the character cursor image 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 image is animated to "wave goodbye, or, alternatively, wipe 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. In Anthias, 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 ¶¶ 291, 429.) A particular cursor type is associated with "device areas" onscreen. A "device area" is a "subarea window displayed within an application window and which is designated by the application program as an action field through which a user may write to the application." (*Id.* at 3:2-7; Ex. 1003 ¶¶ 291, 429.) 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. (*Id.* at 4:16-25; Ex. 1003 ¶¶ 291, 429.)

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 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, 429.) Objects in the pictorial GUI ("server") are "icons linked to file objects." (Ex. 1007 at 10:23-25; Ex. 1003 ¶¶ 292, 429.) The user provides input to the interface using a gamepad controller, a mouse, or a keyboard. (Ex. 1007 at 10:28-30.) User input "causes movement of the animated character relative to the pictorial image and animates objects in the pictorial image." (Ex. 1007 at 10:25-28.) The animated character cursor adopts a different pose ("specific image") and the cursor's hotspot occupies a different part of the character depending on the character's location on the screen. (*Id*. at 13:47-52, 14:59-67.) For example, the cursor's hotspot may appear as the foot or hand of the character. (*Id*. at 13:25-30.)

In order 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. In Baker, the 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. (Ex. 1003 ¶¶ 296, 429.)

Element 70[a]: cursor image data corresponding to said specific image;

Baker teaches storing information about the size of the cursor and its specific position ("*cursor image data*") on the animated character (shown as the "*specific image*") in a character_model list or array ("*cursor display instruction*"). (Ex. 1007 at 23:1-29; Ex. 1003 ¶¶ 299, 429.) The character_model data structure also contains the location of where animation frames are stored ("*cursor image data*") because the character_model data structure is used to "register images of the character across animations." (Ex. 1007 at 23:24-26; Ex. 1003 ¶¶ 299, 429.) Additionally, Baker teaches storing pointers to bitmap animation frames ("*cursor image data*") (Ex. 1007 at 24:33-59; Ex. 1003 ¶¶ 299, 429) 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, 429.)

Element 70[b]: cursor display code, said cursor display code operable to modify said cursor image; and

Baker discusses various OSs that employ a graphical UI (GUI) and thus discloses those OS's functions and procedures. (Ex. 1007 at 2:32-56; Ex. 1003 ¶¶

303, 429.) A POSITA would understand each OS employs functions or applications to display and modify graphics on the UI ("*cursor display code*"). (Ex. 1003 ¶¶ 303, 429.) Baker discloses rendering a cursor on the screen, tracking the cursor, and modifying its appearance. (Ex. 1007 at 13:47-52, Appendix E, pages 21, 29; Ex. 1003 ¶¶ 305, 429.)

Element 70[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 computer OS in Baker corresponds to the client "remote user's terminal" of Claim 70 and the pictorial UI system in Baker corresponds to the "server" of Claim 70. Ex. 1003 ¶¶ 308, 429.)

Baker teaches a pictorial UI system ("*first server computer*") that transmits to an OS ("*remote user terminal's*") information that the functions and applications employed by the OS ("*cursor display code*") may use to modify the animated character cursor ("*cursor image*"), including the background animation and the character_model list or array (collectively, "*specified content information*"). (Ex. 1003 ¶¶ 310, 429.)

(See section VI.C.1, Element 70[Preamble], above regarding Anthias.)

Element 70[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,

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, Pages 27-29; Ex. 1003 ¶¶ 315, 429.) Baker also discloses maintaining information about the location and position of the character cursor ("*cursor image data*") in the character_model data structure. (Ex. 1007 at 23:20-22; Ex. 1003 ¶¶ 314-315, 429.) The character_model data structure is used to "register images of the character across animations," which means the 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 ¶¶ 314-315, 429.) The play_animation procedure, which is responsible for displaying the animated character overlaid on the background image, reads and updates the character_model when modifying the location, size, and appearance of the animated character cursor. (Ex. 1007 at 23:20-22, Appendix E, pages 27-29; Ex.

1003 ¶¶ 315, 429.) Additionally, Baker teaches storing pointers to bitmap animation frames ("*cursor image data*") (Ex. 1007 at 24:33-59; Ex. 1003 ¶¶ 315, 429) and information about the modifications for each animation in the header of the animation. (Ex. 1007 at 24:61-67; Ex. 1003 ¶¶ 315, 429.) Baker also teaches storing information about the cursor's last location ("*cursor image data*"). (Ex. 1007 at 24:33-59; Ex. 1003 ¶¶ 315, 429.)

The pictorial UI in Baker communicates with the OS by providing the information in the character_model list or array ("*cursor display instruction*") to the OS and the functions and applications it employs ("*cursor display code*") in order to execute the prologue and epilogue animations ("*cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image*"). (Ex. 1007 at Abstract, 23:1-29; Ex. 1003 ¶¶ 316, 429.)

Element 70[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,

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 70 and the pictorial UI system in Baker corresponds to the "server" of Claim 70.

Baker's basic processes start with input interpretation to detect meaningful input signals. (*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 system.

This process of sending inputs to the pictorial UI system allows the OS ("*remote user terminal*") to request that the pictorial UI ("*first server computer*") send a 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*").

(See section VI.C.1, Element 70[c][iii], above regarding Anthias.)

Element 70[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 responsive 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.

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 70 and the pictorial UI system in Baker corresponds to the "server" of Claim 70.

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, Pages 27-29; Ex. 1003 ¶¶ 325, 429.) 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 Ex. 1003 ¶¶ 326, 429.) "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 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 operating system and the functions and applications it employs ("*cursor display code*") to display the correct animation. (Ex. 1003 ¶¶ 327, 429; Ex. 1007 at 20:12-28.) The functions or applications then processes the information in or linked from the character_model list or array and executes the prologue and epilogue animations ("*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*"). (Ex. 1003 ¶¶ 327, 429.)

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 the cursor at that time has a shape and appearance corresponding to the icon displayed on the screen ("*said 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 ¶¶ 327, 429.) 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, 429.)

Thus, claim 70 is rendered obvious by Baker and Anthias.

2. Claim 71 Is Rendered Obvious by *Baker* and *Anthias*

See claim 70. Claim 71 is substantially identical to claim 70 except that claim 71 recites modification of the cursor image "responsive to movement of said cursor image over a specified location on said display of said user's terminal" rather than "responsive 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." Baker and Anthias satisfy claim 71 for the same reasons discussed above regarding claim 70. (Ex. 1003 ¶ 430.)

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

Element 72[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.D.1, Element 70[Preamble], regarding Baker.)

To the extent Baker does not explicitly disclose use of a server to modify a cursor image on a remote user display, Anthias teaches a distributed data processing system in which an application is operating on a client system and graphics data for the application is being drawn on a server system. (See section VI.C.1, Element 70[Preamble], above regarding Anthias.)

Element 72[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 72 and the pictorial UI system in Baker corresponds to the "server" of Claim 72. (Ex. 1003 ¶¶ 354, 429.)

When the user moves a cursor so that it covers a portion of an icon, Baker teaches that the play_animation procedure, which is responsible for displaying the animated character overlaid on the background image, requests content information, including the background animation and the character_model data structure ("*specified content information*"). (Ex. 1007 at Appendix E, Pages 27-29; Ex. 1003 ¶¶ 355, 429.) (See section VI.C.1, Element 70[c][iii], above regarding Anthias.)

Element 72[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

(See section VI.D.1, Elements 70[c][ii] and [iii], above regarding Baker.)

Element 72[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

(See section VI.D.1, Element 70[c][iv], above regarding Baker.)

Element 72[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 responsive 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.

(See section VI.D.1, Element 70[c][iv], above regarding Baker.)

Thus, claim 72 is rendered obvious by Baker and Anthias.

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

See claim 72. Claim 73 is substantially identical to claim 72 except that claim 73 recites modification of the cursor image "responsive to movement of said cursor image over a specified location on said display of said user's terminal" rather than "responsive 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." Baker and Anthias satisfy claim 73 for the same reasons discussed above regarding claim 72. (Ex. 1003 ¶ 432.)

VII. CONCLUSION

Petitioner has demonstrated a reasonable likelihood that it will prevail in its

challenge of patentability for Claims 70-73 of the '102 Patent. Petitioner respectfully requests, therefore, that a trial for *inter partes* review on the '102 Patent be instituted.

Dated: September 18, 2018

Respectfully submitted,

/James Valentine/

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

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

Attorneys for Petitioner
Ralph Lauren Corporation

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

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 13,091, 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

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

Attorneys for Petitioner
Ralph Lauren Corporation

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

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of the foregoing PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 6,118,102 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 '102 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

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

Attorneys for Petitioner
Ralph Lauren Corporation

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