UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC.,

Petitioner,

v.

LEXOS MEDIA IP, LLC,

Patent Owner.

Case IPR2023-1000

U.S. Patent No. 5,995,102

Case IPR2023-1001

U.S. Patent No. 6,118,449

DECLARATION OF DR. CRAIG ROSENBERG IN SUPPORT OF PETITIONS FOR *INTER PARTES* REVIEW OF U.S. PATENT NOS. 5,995,102 AND 6,118,449

TABLE OF CONTENTS

Page

I. INTRODUCTION AND QUALIFICATIONS 1

A. Qualifications 2

1. Education 2

2. Work Experience 3

B. Materials Reviewed 8

II. BACKGROUND OF THE ART 10

A. Level of Ordinary Skill in the Art 10

B. Priority Date 12

C. Overview of Relevant Technology When the ’102 and ’449 Patents Were Filed 13

1. User Interfaces 13

2. Custom Cursors 17

3. Tooltips and Balloon Help 23

4. Client-Server Systems 26

III. THE CHALLENGED LEXOS PATENTS 30

A. Background and General Description of the Lexos Patents 30

B. Challenged Claims 34

C. Claim Construction 34

IV. OVERVIEW OF GROUNDS 36

A. Legal Standards 36

B. Summary of Opinion 38

V. GROUNDS FOR INVALIDITY OF THE ’102 PATENT 39

TABLE OF CONTENTS
(continued)

Page

A. Ground 1: Claim 72 of the ’102 Patent Is Rendered Obvious by Malamud 39

1. Overview of Malamud 39

2. Claim 72 is Obvious in View of Malamud 42

B. Ground 2: Claim 72 Is Rendered Obvious by Malamud and Nakagawa 55

1. Overview of Nakagawa 56

2. Motivations to Combine Malamud and Nakagawa 58

3. Claim 72 60

C. Ground 3: Claim 72 Is Rendered Obvious by Nielsen and Malamud 66

1. Overview of Nielsen 66

2. Motivations to Combine Nielsen and Malamud 72

3. Claim 72 Is Obvious in View of Nielsen and Malamud 75

VI. GROUNDS FOR INVALIDITY OF THE ’449 PATENT 82

A. Ground 1: Claims 1, 27, 53 of the ’449 Patent are Rendered Obvious by Malamud 82

1. Motivation to Download Malamud’s Application Program from a Server, and the Obviousness of Doing So 82

2. Claim 1 82

3. Claim 27 88

4. Claim 53 89

B. Ground 2: Claims 1, 27, 53 of the ’449 Patent Are Rendered Obvious by Malamud and Nakagawa 99

TABLE OF CONTENTS
(continued)

Page

1. Motivation to Combine 99

2. Claim 1 100

3. Claim 27 106

4. Claim 53 107

C. Ground 3: Claims 1, 27, 38, 53 of the ’449 Patent are Rendered Obvious by Nielsen and Malamud 112

1. Motivation to Combine 112

2. Claim 1 113

3. Claim 27 120

4. Claim 38 121

5. Claim 53 121

I. INTRODUCTION AND QUALIFICATIONS

1. I, Craig Rosenberg, have been retained by Amazon, Inc. ("Petitioner") to investigate and opine on certain issues relating to United States Patent Nos. 5,995,102 ("the '102 Patent") and 6,118,449 ("the '449 Patent") (collectively, "the Lexos Patents") in their Petitions for *Inter Partes* Review. The Petitions request that the Patent Trial and Appeal Board ("PTAB" or "Board") review and cancel claim 72 of the '102 Patent and claims 1, 27, 38, and 53 of the '449 Patent.

2. The opinions set forth in this declaration are based on my personal knowledge, my professional judgment, and my analysis of the materials and information referenced in this declaration and its exhibits and appendices.

3. I am being compensated at a combined rate of $645 per hour for consulting services including time spent testifying at any hearing that may be held. I am also being reimbursed for reasonable and customary expenses associated with my work in this case. I receive no other forms of compensation related to this case. My compensation does not depend on the outcome of these *inter partes* reviews or the co-pending district court litigation, and I have no other financial interest in these *inter partes* reviews.

4. I understand that the '102 Patent and '449 Patent have ostensibly been assigned to Lexos Media IP, LLC.

5. This declaration is based on the information currently available to me. To the extent that additional information becomes available, I reserve the right to continue my investigation and study, which may include a review of documents and information that may be produced, as well as testimony from depositions that have not yet been taken.

A. Qualifications

6. My qualifications to testify about the ’102 Patent and ’449 Patent and the relevant technology are set forth in my curriculum vitae (“CV”), which I have included as EX1012. In addition, a brief summary of my qualifications is included below.

1. Education

7. I graduated from the University of Washington in 1988 with a B.S. in Industrial Engineering. After graduation, I continued my studies at the University of Washington, earning an M.S. in Human Factors in 1990. In 1994, I graduated from the University of Washington with a Ph.D. in Human Factors. When I attended the school. the Human Factors department was part of the engineering department at the University of Washington.

8. In the course of my doctoral studies, I worked as an Associate Assistant Human Factors Professor at the University of Washington Industrial Engineering

Department. My duties included teaching, writing research proposals, designing and conducting funded human factors experiments for the National Science Foundation.

9. While studying at the University of Washington, I also worked as a human factors researcher and designed and performed advanced human factors experiments relating to virtual environments and interface design, stereoscopic displays, and advanced visualization research, which was funded by the National Science Foundation. My duties included user interface design, systems design, software development, graphics programming, experimental design, as well as hardware and software interfacing. My course work at the University of Washington included programming and software development, and I learned to program in several different computer languages, including C, C++, and Java.

2. Work Experience

10. Since leaving college and for the last 30 years, I have worked in the areas of user interface design, software development, software architecture, systems engineering, human factors, and modeling and simulation across a wide variety of application areas, including aerospace, communications, entertainment, and healthcare.

11. For example, for the past 21 years, I have served as a consultant for various companies, including Global Technica, Sunny Day Software, Stanley

Associates, Techrizon, CDI Corporation, and the Barr Group. In this capacity, I have provided advanced engineering services for many companies.

12. I consulted for the Boeing Company for over 16 years as a senior human factors engineer, user interface designer, and software architect for a wide range of advanced commercial and military programs. Many of the projects that I have been involved with include advanced software development, user interface design, agent-based software, and modeling and simulations in the areas of missile defense, homeland security, battle command management, computer aided design, networking and communications, air traffic control, location-based services, and Unmanned Aerial Vehicle ("UAV") command and control. Additionally, I was the lead system architect developing advanced air traffic controller workstations and air traffic control analysis applications, toolsets, and trade study simulations for Boeing Air Traffic Management.

13. In one project for Boeing, I was the lead human factors engineer and user interface designer for Boeing's main vector and raster computer aided drafting and editing system that produces the maintenance manuals, shop floor illustrations, and service bulletins for aircraft produced by the Boeing Commercial Aircraft Company.

14. Additional responsibilities in my time as a consultant include system engineering, requirements analysis, functional specification, use case development,

user stories, application prototyping, modeling and simulation, object-oriented software architecture, graphical user interface analysis and design, as well as UML, C++, C#, and Java software development.

15. In 1995 and 1996, I was hired as the lead human factors engineer and user interface designer for the first two-way pager produced by AT&T. Prior to this technology, people could receive pages but had no way to respond utilizing their pager. This new technology allowed users to use a small handheld device to receive and send canned or custom text messages, access and update an address book, and access and update a personal calendar. This high-profile project involved designing the entire feature set, user interface/user interaction design and specification, as well as all graphical design and graphical design standards.

16. From 1999–2001, I was the lead human factors engineer and user interface designer for a company called Eyematic Interfaces that was responsible for all user interface design and development activities associated with real-time mobile handheld 3D facial tracking, animation, avatar creation and editing software for a product for Mattel. My work involved user interface design, human factors analysis, requirements gathering and analysis, and functional specifications.

17. In 2001, I was the lead user interface designer for a company called Ahaza that was building IPv6 routers. I designed the user interfaces for the configuration and control of these advanced network hardware devices. My

responsibilities included requirements analysis, functional specification, user interface design, user experience design, and human factors analysis.

18. In 2006-07, I was the lead user interface designer for a company called ObjectSpeed that developed a portable handheld telephone for use in homes and businesses that had many of the same capabilities that we take for granted in mobile cellular phones. This portable multifunction device supported voice, email, chat, video conferencing, internet radio, streaming media, Microsoft Outlook integration, photo taking and sharing, etc. The ObjectSpeed device was specifically designed and developed as a portable handheld device.

19. I am the founder, inventor, user interface designer, and software architect of WhereWuz. WhereWuz is a company that produces advanced mobile software running on GPS-enabled smartphones and handheld devices. WhereWuz allows users to record exactly where they have been and query this data in unique ways for subsequent retrieval based on time or location. WhereWuz was specifically designed and developed to run on small handheld devices.

20. I am the co-founder of a medical technology company called Healium. Healium developed advanced wearable and handheld user interface technology to allow physicians to more effectively interact with electronic medical records.

21. I am the co-founder of a medical technology company called StratoScientific. StratoScientific is developing an innovative case for a smartphone

that turns a standard handheld smartphone into a full featured digital stethoscope that incorporates visualization and machine learning that can be utilized for telemedicine and automated diagnosis.

22. In 2012-13, I designed and developed a large software project for Disney World called xVR that allowed the operational employees of Disney World to utilize a handheld device to view the current and historical status of all of the guests of Disney World within multiple attractions as well as within one of their restaurants. The application could run in a real-time/live mode where it would display data collected from sensors that showed the location and status of all guests within the attraction; the application could also be run in a fast-time/simulated mode. The application was developed on a laptop computer and was specifically designed to run on a variety of devices, including laptops, PCs, smartphones, and tablets.

23. I have received several awards for my engineering work relating to interface design, computer graphics, and the design of spatial, stereographic, and auditory displays, including a $10,000 scholarship from the I/ITSEC for advancing the field of interactive computer graphics for flight simulation and a Link Foundation award for furthering the field of flight simulation and virtual interface design. I have also created graphics for several popular book covers as well as animations for a movie produced by MIRAMAR.

24. I have published twenty-one research papers in professional journals and proceedings in the areas of user interface design, computer graphics, and the design of spatial, stereographic, and auditory displays. I also authored a book chapter on augmented reality displays in the book "Virtual Environments and Advanced Interface Design" (Oxford University Press, 1995). In addition, I created one of the first virtual spatial musical instruments called the MIDIBIRD that utilized the MIDI protocol, two six-dimensional spatial trackers, a music synthesizer, and a computer graphics workstation to create an advanced and novel musical instrument.

B. Materials Reviewed

25. In formulating my opinions in this matter, I have considered the following materials:

Exhibit No.	Description
EX1001	U.S. Patent No. 5,995,102 entitled *Server System and Method for Modifying a Cursor Image* to James Samuel Rosen, et al. ("the '102 Patent").
EX1002	U.S. Patent No. 6,118,449 entitled *Server System and Method for Modifying a Cursor Image* to James Samuel Rosen, et al. ("the '449 Patent").
EX1004	U.S. Patent No. 6,437,800 to Mark A. Malamud ("Malamud").
EX1005	U.S. Patent No. 5,835,911 to Toru Nakagawa, et al. ("Nakagawa").

Exhibit No.	Description
EX1006	U.S. Patent No. 5,937,417 to Jakob Nielsen ("Nielsen").
EX1007	*Lexos Media IP, LLC v. APMEX, Inc.*, No. 2:16-cv-00747-JRG-RSP ("*APMEX*"), Early Claim Construction Opinion and Order, Dkt. 86 (E.D. Tex., Mar. 16, 2017).
EX1008	*Lexos Media IP, LLV v. Amazon.com, Inc.*, No. 2:22-cv-00169-JRG, Parties' Joint Claim Construction and Prehearing Statement, Dkt. 89 (E.D. Tex., May 16, 2023) (including the exhibits attached thereto).
EX1009	U.S. Patent No. 5,754,176 to Chris Crawford ("Crawford").
EX1010	File History of the '102 Patent.
EX1011	File History of the '449 Patent.
EX1012	Curriculum Vitae of Dr. Craig Rosenberg.
EX1014	Appendices to the Declaration of Dr. Craig Rosenberg (compilation of additional references cited herein, but not cited in the IPR petitions).

26. I have also reviewed documents from the prior IPR proceedings regarding the '102 and '449 Patents.

27. I have also relied on my own expertise and experience in user interface design, computer systems, and Human Factors in forming my opinions.

II. BACKGROUND OF THE ART

A. Level of Ordinary Skill in the Art

28. I am not an attorney and offer no legal opinions. I have been informed about certain aspects of the law for purposes of my analyses and opinions.

29. I understand that in analyzing questions of invalidity and infringement, the perspective of a person having ordinary skill in the art ("POSITA") is often implicated, and the Court may need assistance in determining that level of skill.

30. I understand that the claims and written description of a patent must be understood from the perspective of a POSITA. I have been informed that the following factors may affect the level of skill of a POSITA: (1) the educational level of the inventor; (2) the type of problems encountered in the art; (3) the prior art solutions to those problems; (4) the rapidity with which innovations are made; (5) the sophistication of the technology; and (6) the educational level of active workers in the field. A person of ordinary skill in the art is also a person of ordinary creativity in the art.

31. Based on my experience and reading of the '102 and '449 Patents, a person of ordinary skill in the art ("POSITA") at the claimed priority date would have had experience in the fields of human factors engineering or human-computer interaction. The POSITA would have at least a bachelor's degree in computer science, computer engineering, human factors engineering, or a related field and

would have had at least two years of relevant work experience in the fields of UI design, or equivalent experience.

32. Human-Computer Interaction (HCI) and User Interface (UI) Design are two interrelated disciplines within the field of computer science and design, but they focus on different aspects of the user experience.

33. Human-Computer Interaction (HCI): HCI is a broad field that studies how humans interact with computers and technology. It involves designing and implementing interactive computing systems and understanding the ways in which such systems fit into the user's life and daily routine. HCI is concerned with theories, methods, and tools for understanding the relationship between users and digital technologies. It not only includes the design and evaluation of user interfaces but also studies how people interact with technology in different contexts (e.g., work, home, school, etc.), and it involves psychological and sociological aspects as well.

34. User Interface (UI) Design: UI Design is a subset of HCI and focuses specifically on designing the points of interaction between users and digital devices. These points of interaction, or interfaces, may include screens, keyboards, mouse devices, and even voice-controlled interfaces. UI design is concerned with aesthetics (how the product looks), usability (how easy it is to use), and the overall experience provided to the user.

35. In short, HCI is the study of how humans interact with computers and digital technologies, whereas UI design is the practice of creating interfaces that facilitate those interactions. UI Design is more specific, focusing on the aesthetics and functionality of software or a website, while HCI has a broader scope, incorporating aspects like human behavior, psychology, and the societal impacts of technology.

36. I am at least a person of ordinary skill in the art and was so on the date to which the ’102 and ’449 Patents claim priority. As shown by my qualifications and my curriculum vitae, I am aware of and had the knowledge and skill possessed by a person of ordinary skill in the art at the time of the priority date of the ’102 and’449 Patents. In performing my analysis, I have applied the standard set forth above.

B. Priority Date

37. I have been informed that a U.S. patent application may claim the benefit of the filing date of an earlier patent application if the earlier patent application disclosed each limitation of the invention claimed in the later-filed U.S. patent application. I have also been informed that priority is determined on a claim-by-claim basis so that certain claims of a patent may be entitled to the priority date of an earlier-filed patent application even if other claims of the same patent are not entitled to that priority date.

38. I have also been informed that a patented claim is invalid if the claimed invention was patented, described in a printed publication, or in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention, or the claimed invention was described in an issued patent or a published patent application that was effectively filed before the effective filing date of the claimed invention.

39. I understand that the claims of the '102 and'449 Patents claim a priority date of June 25, 1997. I have applied that date as the priority date for the two patents in my analysis.

C. Overview of Relevant Technology When the '102 and '449 Patents Were Filed

1. User Interfaces

40. Generally, a graphical user interface (GUI) is an interface that allows users to interact with a computer through menus or icons displayed on a screen. GUIs are designed to be visually appealing and easy to use, and users can interact with the displayed information using devices like a mouse, touchpad, or stylus pen. One popular type of GUI is the "desktop metaphor," where the computer screen represents a virtual desktop with icons representing files and applications that users can select. The "desktop metaphor" was widely used before 1997 (e.g., in Windows 1995) and remains widely used today.

41. The first public demonstration of a GUI is credited to Douglas Engelbart in 1968[1], which showcased the first computer mouse and cursor. XEROX Parc developed the Alto computer in 1973, featuring the "Neptune Directory Editor" GUI with graphical buttons and file lists. In the following year, PARC researchers created the object-oriented Smalltalk-71 language, enabling the creation of GUIs with the desktop metaphor, icons, scrollbars, superposed windows, and popup menus.

42. Apple introduced GUIs with the Lisa system in 1983 and the Macintosh system in 1984. These GUIs featured drop-down menus, overlapping windows, and icons for programs and documents. The Lisa and Macintosh systems used a one-button mouse, introducing concepts like double-clicking to launch applications and drag-and-drop functionality.

43. As computers and their GUIs became more complex, computer operating systems (OS) were developed to manage the hardware and software interactions. An OS is software that oversees computer hardware (e.g., processor,

---

[1] Douglas C. Engelbart & William K. English, *A Research Center for Augmenting Human Intellect*, Proc. of the Dec. 9-11, 1968 Fall Joint Comput. Conf. 395-410, AFIPS '68 (Fall, part I) (EX1013, Appx. A), *available at* http://dx.doi.org/10.1145/1476589.1476645. [References cited in this declaration but not relied upon as grounds for invalidity are included the Appendix to my declaration, submitted as EX1013.]

memory, display devices, pointing devices, etc.) and software, utilizing specialized programs called drivers to facilitate these interactions.

44. For instance, when a user moves a pointing device like a mouse, the cursor on the screen moves accordingly. This is achieved through the device driver detecting the pointing device's movement and relaying that information to the OS. The OS then communicates the movement to relevant application programs and sends commands to the display driver to depict the cursor's movement on the screen. The OS or an application program can determine the signals sent to the display driver, enabling the rendering of text, images, and graphical elements at the cursor's location. The cursor's image includes a single pixel known as the "hotspot," which indicates where user input, such as a mouse click, will have an effect.

45. Display drivers are one approach employed by OSs to control their integrated display components. Cursors can be presented on the screen through API calls to the operating system as well as custom code that other applications can implement to present cursors on the screen. For example, an application may send data to an OS to initiate or modify a cursor's display.

46. The functionality of a display driver for managing the display of cursors was well-known before the priority date for the Lexos Patents.[2] For example, in 1994, the Windows 95 display driver was described as shown below:

> Once all the initialization is over, GDI, the mini-driver, the DIB engine, VFLATD, and the display VxD are all hooked together and ready to actually put something on the screen. The display mini-driver provides a standard set of about 30 or so interfaces that allow GDI to interact with the driver. Many of these functions are the same as those defined for existing Windows 3.1 display drivers, such as those for managing the cursor. All of them are exported entry points from the driver DLL. Several functions simply accept the call from GDI and hand it directly to the DIB engine. For example, GDI will call the driver's *BitmapBits()* function whenever an application creates or copies a bitmap. The mini-driver can turn around and call the DIB engine's *DIB_BitmapBits()* entry point with no transformation of parameters or, indeed, any other processing.
>
> Management of the cursor is handled largely by the mini-driver, and, as with Windows 3.1 display drivers, the mini-driver must define the set of standard cursor resources used by GDI. This includes objects such as the standard arrow pointer, the I-beam cursor used in text fields, and the cursor we all hope we'll see a lot less of, the hourglass.

(EX1013, Appx. B at 268.)

[2] Adrian King, *Inside Windows 95*, 268 (Microsoft Press 1994). (EX1013, Appx. B.)

Ebay Exhibit 1003, Page 20 of 133
Ebay, Inc. v. Lexos Media IP, LLC
IPR2024-00337

2. Custom Cursors

47. As the mouse and cursor became widespread, their features and uses evolved, too. Early personal computers like the Apple Macintosh used many different types of cursor images to represent the status of the computer or an application program.

48. In February 1995, Microsoft released its Windows Interface Guidelines (the "Guidelines").[3] The Guidelines described mouse pointers that change in appearance, explaining that "[a]s a user moves the pointer across the screen, its appearance can change to provide feedback to the user about a particular location, operation, or state." (EX1013, Appx. C at 23.) The Guidelines showed common pointer shapes and their uses in Table 4.1.

---

[3] *The Window Interface Guidelines - A Guide for Designing Software*, (Microsoft Windows Feb. 1995) (EX1013, Appx. C).

Table 4.1 Common Pointers

Shape	Screen location	Indicates available or current action
	Over most objects	Pointing, selecting, moving, resizing.
	Over text	Selecting text.
	Over any object or location	Processing an operation.
	Over any screen location	Processing in the background (application loading), but the pointer is still interactive.
	Over most objects	Contextual Help mode.
	Inside a window	Zooming a view.
	Along column gridlines	Resizing a column.
	Along row gridlines	Resizing a row.
	Over split box in vertical scroll bar	Splitting a window (or adjusting a split) horizontally.
	Over split box in horizontal scroll bar	Splitting a window (or adjusting a split) vertically.
	Over any object	Not available.

(*Id.*)

49. The Guidelines further described the concepts of the "hot spot" and "hot zone" when using a mouse. (*Id.* at 33.) Each pointer defines an exact screen location, known as the "hot spot," of the mouse, where the hot spot determines what object is affected by the mouse's actions. (*Id.*) Screen objects can also define a "hot zone," which defines the area that the hot spot must cover in order to be considered over the object. (*Id.*) Pressing the mouse button identifies an operation and releasing the mouse button activates or carries out the operation. (*Id.*) The Guidelines

provided the following exemplary common behaviors performed with the mouse as shown below.

Action	Description
Pointing	Positioning the pointer so it "points to" a particular object on the screen without using the mouse button. Pointing is usually part of preparing for some other interaction, because the mouse pointing action is often an opportunity to provide visual cues or other feedback to a user.
Clicking	Positioning the pointer over an object and then pressing and releasing the mouse button. Generally, the mouse is not moved during the click, and the mouse button is quickly released after it is pressed. Clicking identifies (selects) or activates objects.
Double-clicking	Positioning the pointer over an object and pressing and releasing the mouse button twice in rapid succession. Double-clicking an object typically invokes its default operation.
Pressing	Positioning the pointer over an object and then holding down the mouse button. Pressing is often the beginning of a click or drag operation.
Dragging	Positioning the pointer over an object, pressing down the mouse button while holding the mouse button down, and moving the mouse. Use dragging for actions such as selection and direct manipulation of an object.

(*Id*.)

50. For example, the Guidelines explain that the action of "Pointing" refers to positioning a cursor to "point to" a particular object on the screen. The Guidelines go on to explain that the "pointing action is often an opportunity to provide visual cues or other feedback to a user." (*Id*. at 33.)

51. Cursor images, however, were not just limited to the use of standard cursors made available by an OS. Instead, application programs could use Win32 API functions made available in Windows 95 and Windows NT "to change or reset

a system mouse cursor."[4] Even though the article was published in August of 1997, the content of the article discusses the cursor capabilities of Windows NT (which was released in July of 1993), and Windows 95 (which was released in August of 1995.) Through the use of the Win32 API functions, users could program a GUI to use their own cursor in lieu of Windows' standard cursors, for example, by using the LoadCursor function.[5]

52. Similarly, the X Windows system also made it possible for application programs to use customized cursors long before June 1997. The X Windows system was developed through a joint venture between MIT Laboratory for Computer Science and Digital Equipment Corporation in the mid-1980s. (EX1013, Appx. F at

---

[4] P.V. Tatavu, *How to change and how to reset mouse cursors (pointers)*, Level Extreme (Aug. 23, 1997) (EX1013, Appx. D), available at https://www.levelextreme.com/Home/ShowHeader?Activator=23&ID=7677.

[5] D. D. Hobson, *An Introduction to Windows programming* at Section 5.3 (Feb. 1997) (EX1013, Appx. E), available at https://ecs.syr.edu/faculty/Fawcett/handouts/coretechnologies/WindowsProgramming/Win32Prog/Don_Hobson/Don_Hobson.htm ("If you want to have your own cursor when the user is in the window you can use *LoadCursor* accordingly.").

p. 79.)[6] In a 1986 publication describing X Windows, the authors and developers of the X Windows system explained that application programs had the ability to define customized cursors for use in the application window. (*Id*. at p. 88 ("The basic resources provided by the server are windows, fonts, ***mouse cursors***, and offscreen images; later sections describe each of these.") (emphasis added); *id.*, at p. 93 ("Bitmaps are also used to construct cursors, as described in Section 8.").)

53. The X Windows article further explains:

> We now turn to a discussion of input events, but first we briefly describe the support for mouse cursors. Clients can define arbitrary shapes for use as mouse cursors. A cursor is defined by a source bitmap, a pair of pixel values with which to display the bitmap, a mask bitmap that defines the precise shape of the image, and a coordinate within the source bitmap that defines the "center" or "hot spot" of the cursor. Cursors of arbitrary size can be constructed, although only a portion of the cursor may be displayed on some hardware.

(*Id*. at 98.)

54. The X Windows system served as an extremely popular platform for graphical user interfaces (GUIs) running on Unix systems for many years. I

---

[6] Robert W. Scheifler & Jim Gettys, *The X Windows System*, ACM Transactions on Graphics (vol. 5, Apr. 1986) (EX1013, Appx. F).

personally used X Windows on IBM, Sun, HP, and NeXT computers at both the University of Washington and Boeing from 1988 to 2003.

55. Additionally, U.S. Patent No. 6,437,800 ("Malamud," filed on October 26, 1994) teaches defining custom cursors using application data.

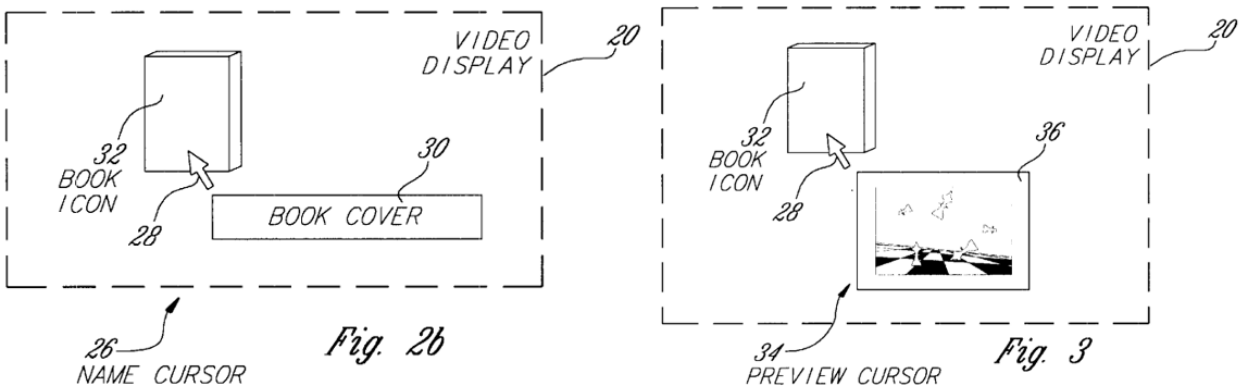
> The present invention provides "information cursors" which display graphical or textual information about an object to which the cursor points. Each information cursor has two portions: a pointing portion and an information portion. The pointing portion points to a position on a video display. The information portion displays textual or graphical information about the object to which the pointing portion points. Information cursors are made available by an operating system to applications that are run on the operating system. Information cursors are available in a number of different varieties. Four varieties of information cursors are provided by the preferred embodiment. The four varieties of information cursors are name cursors, preview cursors, combined name and preview cursors, combined name and preview cursors, and property cursors.

(EX1004 at 2:66-3:12.)

56. Malamud demonstrated changing the cursor's form based on its location on the display and the object pointed to by the cursor. (*Id*. at 5:2-9, 5:24-44). When the cursor is moved over a named object displayed on the screen, it is modified from a "conventional cursor" to an "information cursor" that can "display

graphical or textual information about [the] object to which the cursor points." (*Id.* at 2:66-3:1, 5:32-39).

57. Figures 2b and 3 depict cursor modifications where the "conventional cursor" changes to an "information cursor" to display the name of the object (Figure 2b) or preview the object (Figure 3).



58. In systems like Malamud, the information is presented as part of the cursor because that is typically where user's focus their attention. In other words, presenting the information as part of the cursor makes it more likely that the user will notice the information. As another example, U.S. Patent No. 5,754,176 describes a "tooltip" system built into Microsoft Windows that displays help information when a user held the cursor over an object displayed on the screen. (EX1009 at 2:28-37.)

3. Tooltips and Balloon Help

59. Prior to 1997, Tooltips had been developed and were in public use. Tooltips allowed a computer to display additional supplemental information when

the cursor hovered over a given object on the display. Balloon Help was similar and provided customized context-sensitive information in a balloon image for the user when the user hovered the cursor over a given interface object. Each object on the display would have and display its own customized additional information, that would be presented to the user when the user moved their mouse cursor over the object of interest.

60. There were multiple alternative names given to this type of user interface feature including Info Balloons, Help Text, Hover Tips, Tool Hints, Quick Info, Assistance Bubbles, Pop-up Messages, Screen Tips, etc. An article published in the ACM SIGDOC Journal of Computer Documentation in May of 1993 titled "The Role of Balloon Help" by David Farkas (https://dl.acm.org/doi/10.1145/154425.154426) (EX1013, Appx. G) shows how the company Apple Computer implemented Balloon Help in their Macintosh System 7 operating system. This article, published in May of 1993, shows that tooltips (e.g., Balloon Help) was available long before 1997. The following is an excerpt from page 4 of the article:

> Once in the Balloon Help mode, many interface objects on the screen are "hot." That is, they will display small "balloons" (see Figure 1) containing brief help messages when the user moves the pointer (cursor) over them. These balloons, which are named after the balloons used in comic book dialog, appear at the

> location of the hot spot. Each balloon has a "tip" that points precisely at the hot interface object. Balloons are parsimonious in design: there are no buttons to click, no scroll bars, no title area. The dimensions of the balloon are barely larger than the space required for the balloon message. [Page 4]

The following is from Figure 1. of the article:

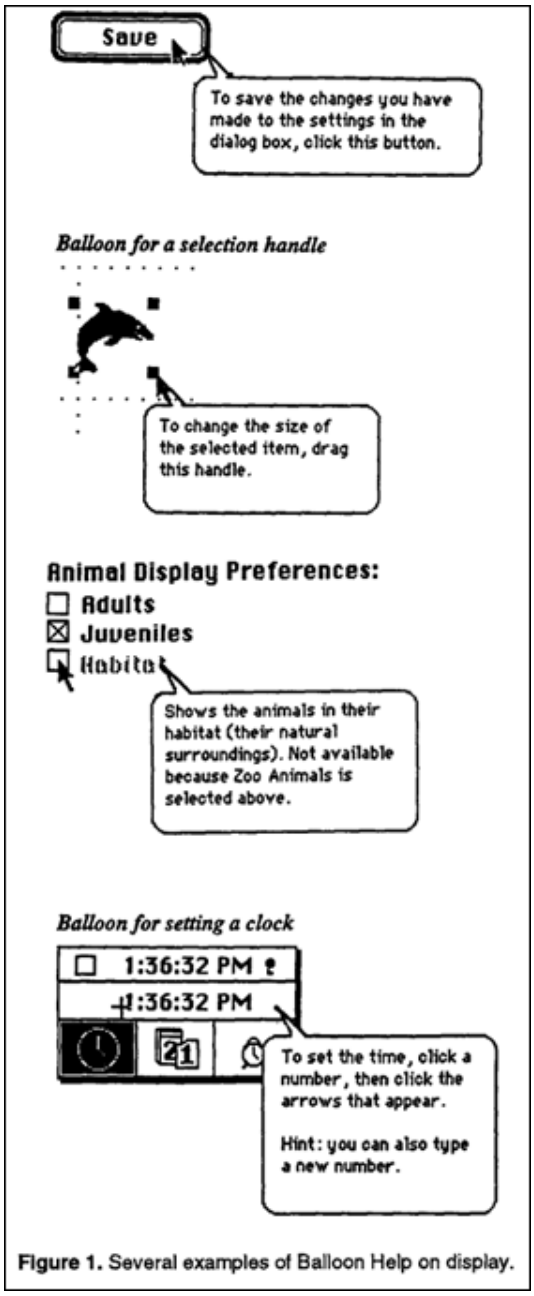


Figure 1. Several examples of Balloon Help on display.

4. Client-Server Systems

61. The term client/server was first used in the 1980s in reference to personal computers on a network. The actual client/server model started gaining acceptance in the late 1980s. A client-server model for computing devices describes a distributed application architecture that partitions tasks between service requesters

(i.e., clients) and providers of resources or services (i.e., servers.) This model is based on the distribution of functions between two types of independent and autonomous entities: servers and clients. A client is any process that requests specific services from server processes. A server is a process that provides requested services for clients.

62. The clients and servers generally communicate over a computer network on separate hardware. The one or more clients are usually associated with the terminals of users and the servers are typically at a remote location. The clients request content or service from the servers and the servers share their resources with the clients. Some examples of client-server systems include email and the World Wide Web.

63. The client-server model is a foundational computer architecture that was widely used by 1997 in many different settings. For example, companies used the client-server model by storing central information, such as files and data stored on servers, whereby those files and data were accessible throughout the company by employee computers (clients). Similarly, schools had servers from which students could access information from one or more servers. List services were in widespread use and permitted clients to access information shared through servers. By 1997, the Internet had already gained great popularity and permitted client devices to access information, data, application programs, and other services from servers

through the Internet. Examples of types of servers and the content that they provide include file servers, print servers, mail servers, directory services servers, web servers, and database servers.

64. It was also well known for clients to download application programs from server prior to June of 1997. (*See, e.g.*, EX1005 at 3:1-5.) For example, clients could request an application program that was compiled and linked on the server computer, with the executable application program downloaded to the client. (*Id*. at 1:66-2:4.) Alternatively, and client could request and receive from a server the source programs, and then the client could compile and link the programs into an executable file that could be run by the client. (*Id*. at 2:4-6.)

65. Downloading programs from the internet began to take shape with the advent of file transfer protocols and the birth of the World Wide Web in the early 1990s. Before then, software was primarily distributed through physical media like floppy disks and CD-ROMs, or through proprietary networks and bulletin board systems (BBS). Once the World Wide Web became more popular in the mid-1990s, software could be hosted on a website and downloaded directly and more easily. Early examples of downloadable programs that were all available before 1997 included:

- Netscape Navigator: Released in 1994, this was one of the first widely-used web browsers that could be downloaded from the internet.
- Winamp: A media player for Windows, released in 1997. It was widely downloaded and known for its customizable skins.
- ICQ: One of the first internet-based instant messaging programs, also released in 1996.
- Various Linux distributions: The free and open-source nature of Linux meant that it was available for download early in the history of the internet. Distros such as Debian, Slackware, and Red Hat could be downloaded and installed from the internet.
- Shareware/Freeware games: Games like Doom, distributed as shareware, where a portion of the game was free to download and play, but you could pay to get the full version.
- FTP software: Tools like WS_FTP LE, released in the mid-'90s, allowed users to transfer files between computers over the internet.

When an application is downloaded from a server, it includes information relating to what will be displayed on the screen as the application program executes. Additionally, to the extent the application program controls the cursor image that is displayed, that information would also be downloaded as part of the application

program. For example, if a user computer downloaded Netscape Navigator 2.0, which was available for download in 1995 from www.netscape.com, the user's computer (e.g., the client) would download both information to be displayed on the screen and instructions regarding how to change the cursor to various shapes as the program is run on the user's computer. This webpage shows various details of the Netscape Navigator 2.0 web browser which was released and available for download from the Internet on www.netscape.com on September 18, 1995 (https://www.webdesignmuseum.org/old-software/web-browsers/netscape-navigator-2-0). This YouTube video shows the use of Netscape Navigator 2.01 in 1995 (https://www.youtube.com/watch?v=qKPO3Xk4DYw). In this video, one can see the Netscape Navigator program changing between at least four different cursor types (arrow cursor [time 0:10], hand cursor [time 0:40], text selection cursor [time 0:15], and hourglass cursor [time 0:03].)

III. THE CHALLENGED LEXOS PATENTS

A. Background and General Description of the Lexos Patents[7]

66. The Lexos Patents primarily focus on the modification of a cursor utilizing a conventional client-server architecture.

[7] The Lexos Patents share a common specification. For consistency and ease of reference for the Board across both related IPR Petitions and my declaration, all

67. This process, as described in the Lexos Patents, encompasses three primary components, "Cursor Display Instructions," "Cursor Display Code," and "Cursor Information." "Cursor Display Code" is defined as the code that is responsible for the actual transformation of the cursor 44." (EX1001 at 8:53-57.) "Cursor Information" is described as a dataset which delineates the actual cursor images, and potentially corresponding audio content. (*Id*. at 8:57-59.) Lastly, "Cursor Display Instructions" involve data conveying information used by the Cursor Display Code 52 to control drivers, such as 36, 40, 46, and to identify attributes such, which among other consist of: the physical location of Cursor Information 54, the format of its representation, the intended manner and duration of its display, and information pertaining to how (and for how long) any cached Cursor Information 54 should be stored." (*Id*. at 8:61-67.)

68. Figure 3 (shown below) shows a flowchart outlining the cursor modification procedure in one possible embodiment.

---

citations to the specification herein will be made to the column and line numbers of the '102 Patent (EX1001).

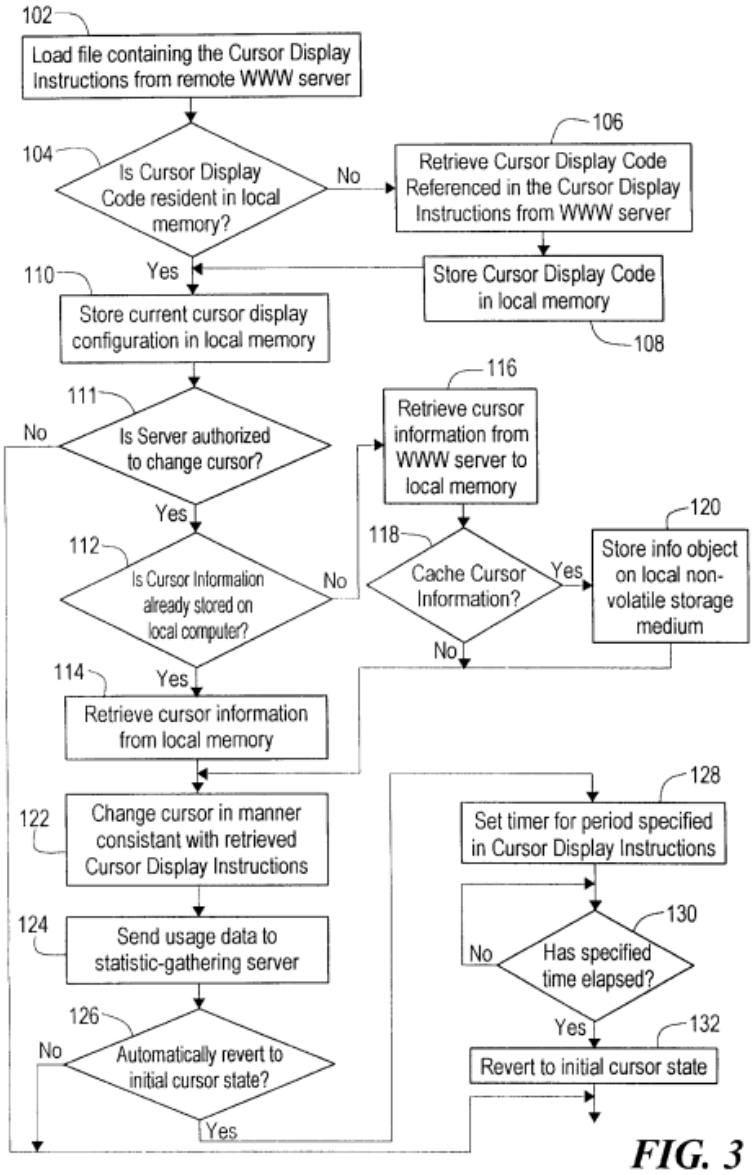


FIG. 3

(*Id*. at Fig. 3.)

69. At Step 102, the user terminal's browser retrieves an HTML file containing Cursor Display Instructions. (*Id*. at 10:23-29.) These instructions include an indication of Cursor Display Code, which, if not stored in local memory (step 104) (*id*. at 10:46-50), is retrieved from a web server (step 106) and stored in local memory along with the current cursor display configuration in local memory (steps

108, 110). (*Id*. at 10:51-58, 11:6-12.) If the server is authorized to change the cursor (step 111) (*id*. at 11:13-15), Cursor Information is retrieved from local memory (step 114) or the server (step 116). (*Id*. at 11:22-29.) Subsequently, the cursor undergoes a change in accordance with the retrieved Cursor Display Instructions (step 122). (*Id*. at 11:34-36.) These Cursor Display Instructions activate a function, resulting in the modification of the cursor's display on the user terminal's video monitor. (*Id*. at 11:46-48.)

70. Figures 7 and 8 illustrate how the standard arrow cursor (designated "44" and annotated below with a red circle) is modified into the "specific image" of a bottle (designated "44a" and also annotated) to advertise and promote a cola beverage:

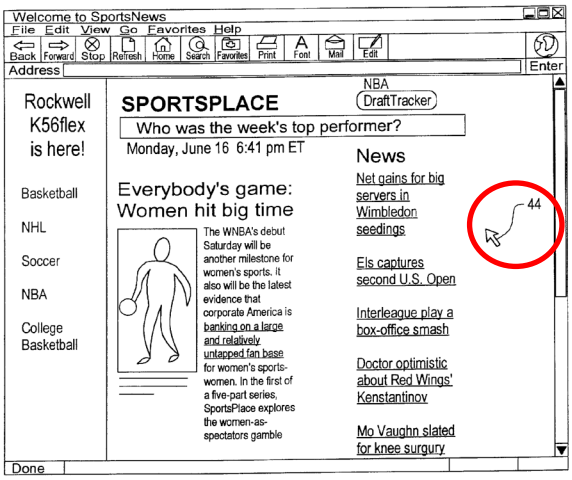


FIG. 7

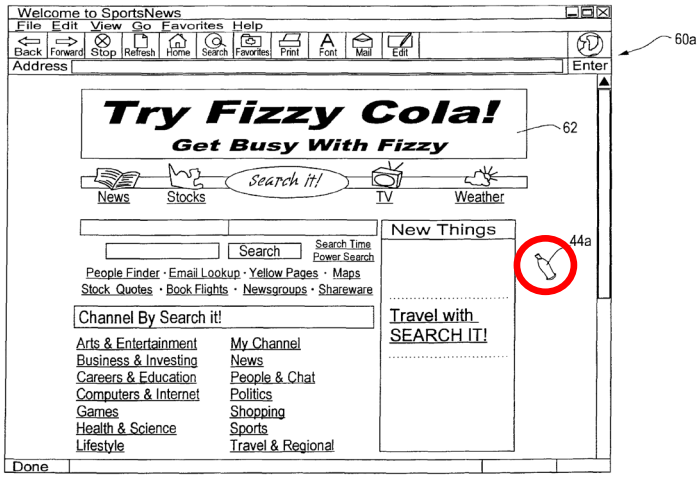


FIG. 8

(*Id*. at Figs. 7-8 (annotated).)

71. Figure 8 demonstrates that, when web page 60a is loaded, the arrow cursor transforms into a bottle-shaped cursor symbolizing "Fizzy Cola," coinciding with the Fizzy Cola banner advertisement 62. When the user subsequently loads a new web page, the bottle-shaped cursor reverts to the standard arrow cursor. (*Id*. at 13:33-45.)

B. Challenged Claims

72. Claim 72 of the '102 Patent and claims 1, 27, 38, and 53 of the '449 Patent ("Challenged Claims") are analyzed in this Declaration.

C. Claim Construction

73. I understand that claim terms generally are construed in accordance with the ordinary and customary meaning they would have to a POSITA at the time

of the invention in light of the claim language, the specification, and the prosecution history. I understand that technical dictionaries and other extrinsic evidence may be considered as well, though such evidence is typically regarded as less significant than the intrinsic record in determining the meaning of the claim language. I also understand that an applicant may act as its own lexicographer and define terms.

74. I understand that Patent Owner previously asserted the Lexos Patents in a district court litigation, and that the district court construed the term "said specific image includes/including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal" to mean "an image representative of at least a portion of the subject or topic being displayed on the screen." (EX1007 at 12-13). I have applied that construction in my analysis. That construction does not change my analysis regarding any of the grounds presented here because, as discussed below, Malamud's preview cursor contains content corresponding to the object displayed on the user's screen to which the pointer is pointing, and Nielsen's tooltips contain content corresponding to the object displayed on the user's screen to which the pointer is pointing. So, the construction of this limitation is met by both references.

75. For the remaining terms of the challenged claims of the Lexos Patents, I have interpreted them as they would have been understood by a POSITA at the time of the alleged invention.

76. I understand that Petitioner and Patent Owner are currently going through the claim construction process in the corresponding district court. For purposes of my opinions in this proceeding, I do not believe any of the claim terms require construction. I have also reviewed the proposed constructions in the pending district court litigation (EX1008), and I do not think that any proposed construction impacts my analysis because the references discussed below disclose or render obvious each claim limitation under any of the proposed constructions.

IV. OVERVIEW OF GROUNDS

A. Legal Standards

77. I understand that a person is not entitled to a patent if the invention was described in a printed publication or patent in this or a foreign country published more than one year prior to the earliest priority date of the application for the patent in the United States. I also understand that a person may not receive a patent for an invention if the invention was described in a patent application having an initial filing date before the date on which the applicant first conceived of the invention in sufficient detail to enable the applicant to implement the invention. I further understand that if each element of a patent claim is disclosed by such a prior patent or publication, either explicitly or inherently, the claim is invalid as "anticipated" by the prior patent or publication.

78. I understand that a patent is obvious if the differences between the subject matter sought to be patented and the prior art are such that the subject matter, as a whole, would have been obvious, at the time of the invention was made, to a person having ordinary skill in the art. I also understand that, in determining whether the subject matter of a patent claim is obvious, neither the particular motivation nor the avowed purpose of the patentee controls. I further understand that what matters is the objective reach of the claim because, if the claim extends to what is obvious, it is invalid. I further understand that one of the ways in which a patent's subject matter can be proved obvious is by noting that there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent's claims. I further understand that to determine whether an invention would have been obvious at the time it was made, the Board may consider four factors: (1) the scope and content of the prior art; (2) the differences between the prior art and the claimed invention; (3) the level of skill in the pertinent art; and (4) any secondary considerations of non-obviousness, such as commercial success of the claimed invention, long felt need, failure of others, and unexpected results or benefits.

79. I understand that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand that a claimed invention may be obvious when there is a

design need or market pressure to solve a problem and there are a finite number of identified, predicable solutions, as a POSITA would have good reason to pursue the known options within his or her technical grasp. I further understand that if pursuing the known options leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense.

B. Summary of Opinion

80. Based on my 35 years of experience in the field of human-computer interaction and user interface design, my review of the Lexos Patents, and the Challenged Claims, as well as other materials in the exhibits to the Petitions or the appendices to this Declaration, it is my opinion that claim 72 of the '102 Patent and claims 1, 27, 38, and 53 of the '449 Patent are obvious over the prior art as summarized below.

'102 Patent

Ground	Section	Challenged Claim	References
1	Pre-AIA 35 U.S.C. § 103(a)	72	Malamud
2	Pre-AIA 35 U.S.C. § 103(a)	72	Malamud and Nakagawa
3	Pre-AIA 35 U.S.C. § 103(a)	72	Nielsen and Malamud

'449 Patent

Ground	Grounds	Challenged Claim	References
1	Pre-AIA 35 U.S.C. § 103(a)	1, 27, 53	Malamud
2	Pre-AIA 35 U.S.C. § 103(a)	1, 27, 53	Malamud and Nakagawa
3	Pre-AIA 35 U.S.C. § 103(a)	1, 27, 38, 53	Nielsen and Malamud

V. GROUNDS FOR INVALIDITY OF THE '102 PATENT

A. Ground 1: Claim 72 of the '102 Patent Is Rendered Obvious by Malamud

81. In my opinion, the teachings of Malamud and the knowledge of a person of ordinary skill in the art render claim 72 of the '102 Patent obvious.

1. Overview of Malamud

82. Like the Lexos Patents, Malamud describes modifying a cursor.

83. Figure 6 provides a flowchart describing a method for modifying a cursor according to Malamud's invention.

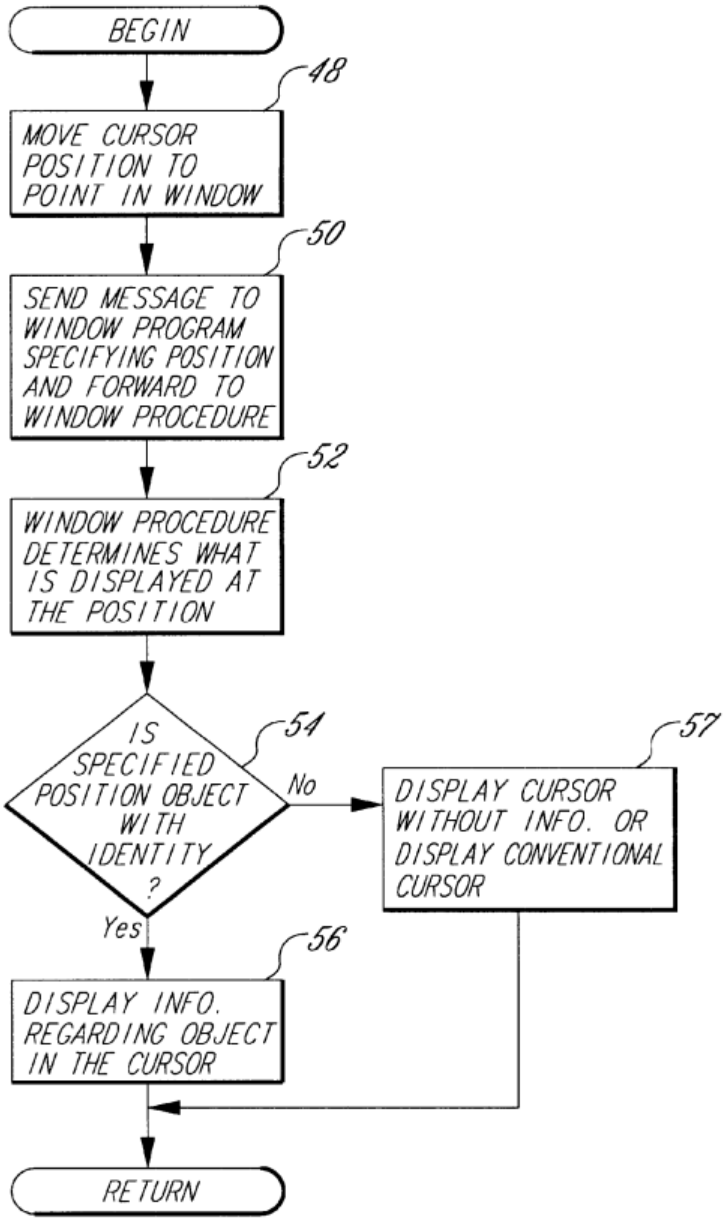


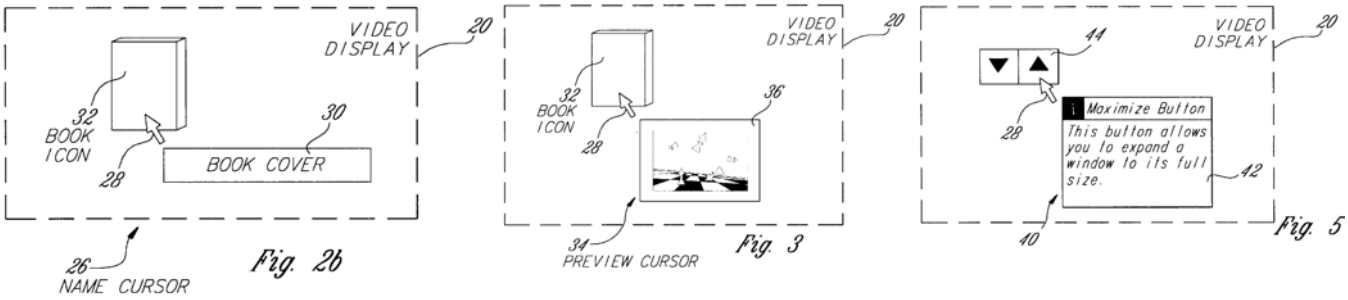
Fig. 6

(EX1004 at Fig. 6.)

84. At Step 48, a user's input device points to a cursor position within a window displayed on the user's display device. (*Id*. at 5:22-24.) The window procedure of the window retrieves a message specifying the cursor position from the executing application program (step 50). (*Id*. at 5:24-27.) The window procedure

processes the message and determines what is displayed at the cursor position within the window (step 52) (*id*. at 5:27-30), such as an object with an entity (step 54). (*Id*. at 5:32-34.) If the application program chooses to change the cursor because an object is present at the cursor position, information about the object is retrieved. (*Id*. at 5:35-39, 5:44-45.) The cursor is then changed to display on the display device with the retrieved information about the object (step 56). (*Id*.) If no object is present, a conventional or blank cursor is displayed on the display device (step 57). (*Id*. at 5:39-44.)

85. Information about the object displayed with the cursor can include the object's name, a preview of the object's contents, or property information about the object. (*Id*. at 3:4-14.) For example, when the object is a book about chess, the information portion may provide a preview of the contents of the book. (*Id*. at 3:59-4:3, Fig. 3). Similarly, the information may provide the name of the object by displaying "Book Cover." (*Id*. at 3:36-47, Fig. 2b).



(EX1004 at Figs. 2b, 3, and 5.)

2. Claim 72 is Obvious in View of Malamud

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

86. In my opinion, Element 72[Preamble] would have been obvious to a POSITA in view of Malamud.

87. Malamud teaches a method for modifying on the display of a user terminal an initial cursor image, such as a "conventional cursor," to a specific cursor image that includes a graphical image related to the object to which the cursor is pointing. Specifically, Malamud teaches that when the cursor is at a location on the display screen where no named entity is present, a "conventional cursor" image or an "information cursor" without any corresponding specific information is displayed. (*Id*. at 5:39-44.) When the cursor is moved over a named entity displayed on the screen, the cursor image changes to an "information cursor" that includes text and/or a graphical image related to the object on the display screen to which the cursor is pointing. (*Id*. at 5:32-39; *id*. at 2:66-3:1 ("The present invention provides 'information cursors' which display graphical or textual information about an object to which the cursor point."); *id*. at 3:59-4:3 (discussing a "preview cursor," which depicts a graphical image related to the object on the display screen to which the cursor points).)

88. Malamud does not address whether the user's computer is connected by a network to a server, but a POSITA would have been motivated to incorporate that feature into Malamud and would have had a reasonable expectation of success in doing so for several reasons, as discussed below.

89. Connecting user computers to a server was well known to POSITAs and commonplace well before June 1997. (EX1005 at 3:1-5; EX1013, Appx. H at 158-160.) Network interface cards (NICs) became common in personal computers (PCs) during the late 1980s and early 1990s. This was when Ethernet started becoming a standard for local area networks (LANs) in organizations. However, it was really with the spread of the internet in the mid-1990s that network cards became practically ubiquitous in PCs, as accessing the internet requires some form of network interface. For example, it was quite common before June 1997 to have use a computer connected to one or more servers in many different contexts, such as: in an enterprise, where employee computers were connected to company servers; in an educational setting, where student computers were connected to a school server; or in a home, where an individual's computer would connect to internet servers. The inventors of the '102 Patent admit that, by 1997, millions of people all over the world were using personal computers to access digital content over the internet. (EX1001 at 1:9-19.)

90. Accessing the World Wide Web was common before 1997. The World Wide Web was first developed in 1990 when Tim Berners-Lee created the first web browser, HTML editor, and web server. This utilized the well-known client-server architecture that allowed HTML and media files to be stored on a server and then fetched by clients that were running a web browser. The image below from page 159 of EX1013, Appx. H, shows the client-server topology that allowed clients (in this figure, it is the computer labelled "Browser" that is running a web browser application) to communicate with web servers that would serve up HTML code (e.g., web pages) to the various clients that request them.

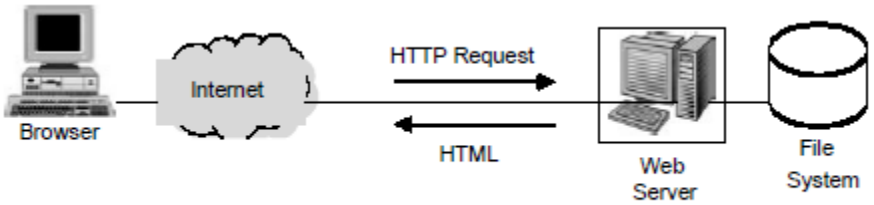


Fig. 8.1(a): Web Technology Yesterday

As described previously, browsers such as Netscape Navigator 2.0, which ran on the client computers, could be downloaded from the internet and could change cursor types depending on the kind of content that the cursor was hovering over. While the cursors available through Netscape were functional in nature, as they related to the state of the computer or program or available functionality, the cursor system and method in Malamud was not so limited.

91. Moreover, it was well-known in the art that application programs could be downloaded over a network from a server (as I discussed in the Background

section of this declaration), and it would have been common sense to do so given 1) the reduction in distribution costs, 2) the increased speed by which users could obtain an application program using the client-server model, and 3) the fact that software companies could ensure that the latest version of the software was always available to end users.

92. At the time of the invention of the '102 Patent, there were a limited number of ways for a user computer to obtain an application program. These include having the application program pre-installed on a computer when the computer was purchased; copying the application program to the user's computer from a floppy disc, CD ROM, or an external drive; or copying the application program to the user's computer from a server over a computer network. Any application program that used Malamud's cursor-modification method that was not already installed on a computer at the time of purchase would have needed to be distributed using one of the other methods. As discussed previously, distribution over a computer network would be faster and less expensive than distribution via floppy disc, CD ROM, or other external media. A POSITA would therefore have been motivated to distribute Malamud's application program from a server to user computers over a network because that would lower costs by eliminating the need to produce and distribute copies of the program on floppy discs or CD ROMS, and it would increase the speed at which users could obtain the program. Additionally, by distributing the

application program from a server over a network, users could be sure that they received the most up-to-date version of the program, while a program distributed by floppy disc or CD ROM could easily become outdated upon the release of a newer version of the program before the user even received the floppy disc or CD ROM.

93. In sum, while Malamud does not discuss distributing software or connecting a computer network to a user's computer, by 1997, it was extremely likely for a user's computer to be connected to a network and/or have networking capabilities. Additionally, the distribution of software from a server to a user's computer over a network was also well-known and a common practice. A POSITA would have been motivated to distribute Malamud's application program over a network using a server, and it would have been well within the technical skill of a POSITA to do so with a high likelihood of success. This distribution of Malamud's application program from a server would not have been innovative, but rather the result of ordinary skill and common sense.

94. Thus, it would have been obvious to a POSITA to use Malamud's method for modifying a cursor image on a computer connected to a server, and a POSITA would have been motivated to do so for the reasons discussed above.

<u>Element 72[a]:</u> receiving a request at said at least one server to provide specified content information to said user terminal;

95. In my opinion, Element 72[a] would have been obvious to a POSITA in view of Malamud.

96. Malamud teaches the use of information cursors, including a "preview cursor," by an application program. (EX1004 at Abstract ("Information cursors are provided for use in an operating system and/or *application programs*.") (emphasis added).) The application program comprises "specified content information" because it includes content to be displayed on the screen when the program is run, and it includes instructions regarding the modified cursor, such as whether to display an information cursor, the type of information cursor to display, and an indication of the content of the information cursor. (*Id*. at 5:24-45, Figs. 3 and 4 (depicting book icon 32, which relates to the application running in that window), 4:56-59 (discussing windows displayed on the screen generated by various programs), 3:59-4:18 (discussing the "preview cursor" and the "combined name and preview cursor").) Thus, Malamud discloses an application program that includes information for use in the display on the user's terminal.[8]

97. Malamud is silent as to how the data processing system (the user's computer) obtains the application program, but it was well-known to a POSITA by

[8] Petitioner and Patent Owner agree that the claim terms "specified content information" and "content information" mean "information provided to a user's terminal for use in the display, such as a web page." (EX1008 at 3; EX1001 at Abstract, 2:58-62, 3:4-9, 24:14-24.)

June of 1997 that a user could request to download an application program from a server and a POSITA would also have been motivated to download application programs, such as the one described in Malamud, from a server, as I explained above. Thus, it would have been obvious to a POSITA to combine Malamud's teaching of an application program that modifies a cursor image with the well-known procedure of downloading an application program from a server for the purpose of installation. This combination renders claim limitation 72[a] obvious, as the request to download Malamud's application program from a server is a request to a server to provide specified content information to a user terminal.

> **<u>Element 72[b]:</u> 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**

98. In my opinion, Element 72[b] would have been obvious to a POSITA in view of Malamud.

99. As discussed in conjunction with claim Element 72[a], it would have been obvious in view of Malamud for a user to request from a server, and receive from the server in response, an application program, and that application program comprises "specified content information" because the program includes instructions and information regarding the display of elements depicted by the application program, including the objects that the application programs will display

and to which the cursor can point, as well as information regarding the modified cursors (i.e., information cursors).

100. The application program in Malamud includes at least one cursor display instruction in the program, which is checked when a cursor is placed over an object on the display screen, and the instruction contains information that is passed to the operating system to cause the cursor image to change to an information cursor, such as a preview cursor. (*See* EX1004 at 5:47-62 (describing the message that the windows procedure of the application program passes, which includes the type of information cursor to display and information about the contents of the information cursor image).) For example, the application program in Malamud contains instructions whether to display an information cursor, and if so the type of information cursor (such as a preview cursor or a combined name and preview cursor) for each entity displayed in the application window, and based on those instructions in the application program, messages are sent to the OS regarding the display of the information cursors. (*Id*. at 3:59-4:18; 5:24-45; 5:46-62.) Malamud also teaches that, for a preview cursor, the application program transmits, and therefore contains, an indication of cursor image data in the form of a "pointer to a bitmap of graphical information that the operating system 22 should use in the preview portion 36." (*Id*. at 5:57-62.) The bitmap is a "specific image" that is displayed as a modified cursor image, and it relates to whatever object on the display

screen that the cursor is pointing because it provides further information about that object. (*Id*. at 5:57-62, 3:59-4:3.) For example, Malamud teaches that if the cursor is pointing to a book, the bitmap can constitute an image that provides further information about the book. (EX1004 at 3:65-4:3.)

101. It would have been obvious to a POSITA in view of Malamud to combine Malamud's system and method for modifying a cursor image with downloading from a server an application program that contains the specified content information.

> **<u>Element 72[c][i]:</u> 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**

102. In my opinion, Element 72[c][i] is disclosed in Malamud. It is also my opinion that it would be obvious that the application program would contain cursor display instructions that instruct the performance of the functionality discussed below.

103. Malamud teaches that when the cursor is not pointing to a named object in the window for the application, "either a conventional cursor is displayed or the information containing portion (e.g., name box 30, preview portion 36, or property box 40) of the information is shown in blank (step 57)." (*Id*. at 5:39-44.) The

disclosure of a conventional cursor or the information cursor without any corresponding specific information is an "initial cursor image." Malamud further teaches determining whether the cursor has been moved over an object displayed in the application's display window, where the object has an identity, (i.e., an associated information cursor). (*Id*. at 5:32-34.) Malamud teaches that "[i]f a named entity is present at the specified cursor position, the information regarding the object at the specified cursor position is displayed in the information cursor (step 56)." (*Id*. at 5:36-39.) If the information cursor is a preview cursor, the modified cursor image will include a graphical image related to the object on the display screen to which the cursor is pointing. (*Id*. at 3:59-4:3.) That graphical image is therefore a specific image and is contained in the bitmap that will be displayed in the preview cursor. (*Id*. at 5:57-62.) This functionality is carried out in response to instructions in the application program that contain information that is passed from the application program's window procedure to the operating system 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 5:47-53.) Thus, Malamud teaches transforming the initial cursor image into the shape and appearance of the specific image in response to a cursor display instruction.

104. Malamud further teaches that the application program provides the information that is to be displayed in the window for that application. (*Id*. at 3:36-

41, 3:63-4:3; *see also id*. at 5:24-30 (discussing the application program controlling the display in a particular window); *see, e.g.*, *id*. at Fig. 3 (depicting book icon 32 generated by the application program).) Thus, the "specified content information" received from the server in the form of the application program would include information to be displayed on the display of the user's terminal.

105. And as discussed above, the preview cursor (i.e., the modified cursor image) includes a graphical image related to the object in the application window to which the cursor is pointing. (*Id*. at 3:59-4:18.) For example, if the cursor is pointing to a book, the preview cursor would include an image related to the subject matter of the book. Therefore, Malamud teaches that the specific image includes content corresponding to at least a portion of the information that is to be displayed on the user's terminal.

> **<u>Element 72[c][ii]:</u> 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.**

106. In my opinion, Element 72[c][ii] is disclosed in Malamud. It is also my opinion that this limitation would have been obvious to a POSITA as discussed below.

107. Malamud teaches that the windows procedure of the application contains instructions that cause the program to pass a message to the operating system that indicates what type of cursor to display and, for a preview cursor, a pointer to a bitmap of graphical information that the operating system should display in the preview portion of the modified cursor image. (*Id*. at 5:47-62.) By indicating the type of information cursor, the cursor display instructions in the application program indicates what cursor display code[9] must formulate and convey the message and cause the user's computer to properly display the appropriate type of information cursor. (*See id*. at 5:47-62, Figs. 6-7.) For example, Malamud teaches cursor display code in the form of an algorithm that would implement portions of Figures 6 and 7. (*Id*. 1004 at Figs. 6-7.) The cursor display instruction would indicate whether there

[9] In the District Court litigation, Petitioner contends that "cursor display code" should be given its plain and ordinary meaning. (EX1008 at Ex. 1, p. 1.) Patent Owner asserts that "cursor display code" means "computer code for modifying the display of the cursor image." (*Id*.) Malamud satisfies Patent Owner's proposed construction, as the code in Malamud that interprets and implements the instructions regarding whether to display an information cursor, the type of information cursor to display, and what graphical information to display all comprises "computer code for modifying the display of the cursor image. (EX1004 at 5:46-62.)

is an information cursor associated with an object and, if so, the type of information cursor, and such an instruction would indicate what cursor display code was necessary to process the instruction and properly convey the necessary information through the message to the OS, such as through an API call. (*Id*. at 5:46-62, Fig. 6 (elements 54, 56, and 57, which would be implemented in code in the application program), Fig. 7 (element 58, which would be implemented in code in the application program, and elements 60, 62, and 64, which would be implemented in code in the OS and/or drivers).)

108. The application program would also contain instructions about the cursor image to display for a preview cursor or a combined name and preview cursor, and that information would also be conveyed as part of the message to the OS in the form of a pointer to a bitmap, which specifies the shape and appearance of the specific image (i.e., the image of the modified cursor). (*Id*.) Malamud teaches that the cursor image is modified in response to the movement of the cursor over an object displayed in the application window (EX1004 at 5:24-39), and that object is a "display of said at least a portion of said information to be displayed on said display of said user's terminal."

109. While disclosed in Malamud, it also would have been obvious to a POSITA that the cursor display instruction indicates the cursor display code needed to process the cursor display instruction because the information in the instruction

must be properly conveyed to the OS, and it was common to convey information to the OS through an API call. The cursor display instruction in Malamud instructs whether a particular entity has an associated information cursor and, if so, the type of information cursor and the text and/or preview image to display. Because different pieces of information need to be conveyed to the OS for different types of information cursors, such as text for information cursor that includes a name box or a pointer for an information cursor that includes a preview portion, it would have been obvious to a POSITA that the information contained in the instruction would indicate what code is needed to process the instruction to properly formulate and convey the message to the OS through the appropriate API call, as this was an extremely common procedure. Therefore, it would have been obvious to a POSITA that the instruction, based upon the information contained therein, would indicate the code to use to process the instruction.

110. Thus, claim 72 would have been obvious to a POSITA in view of Malamud.

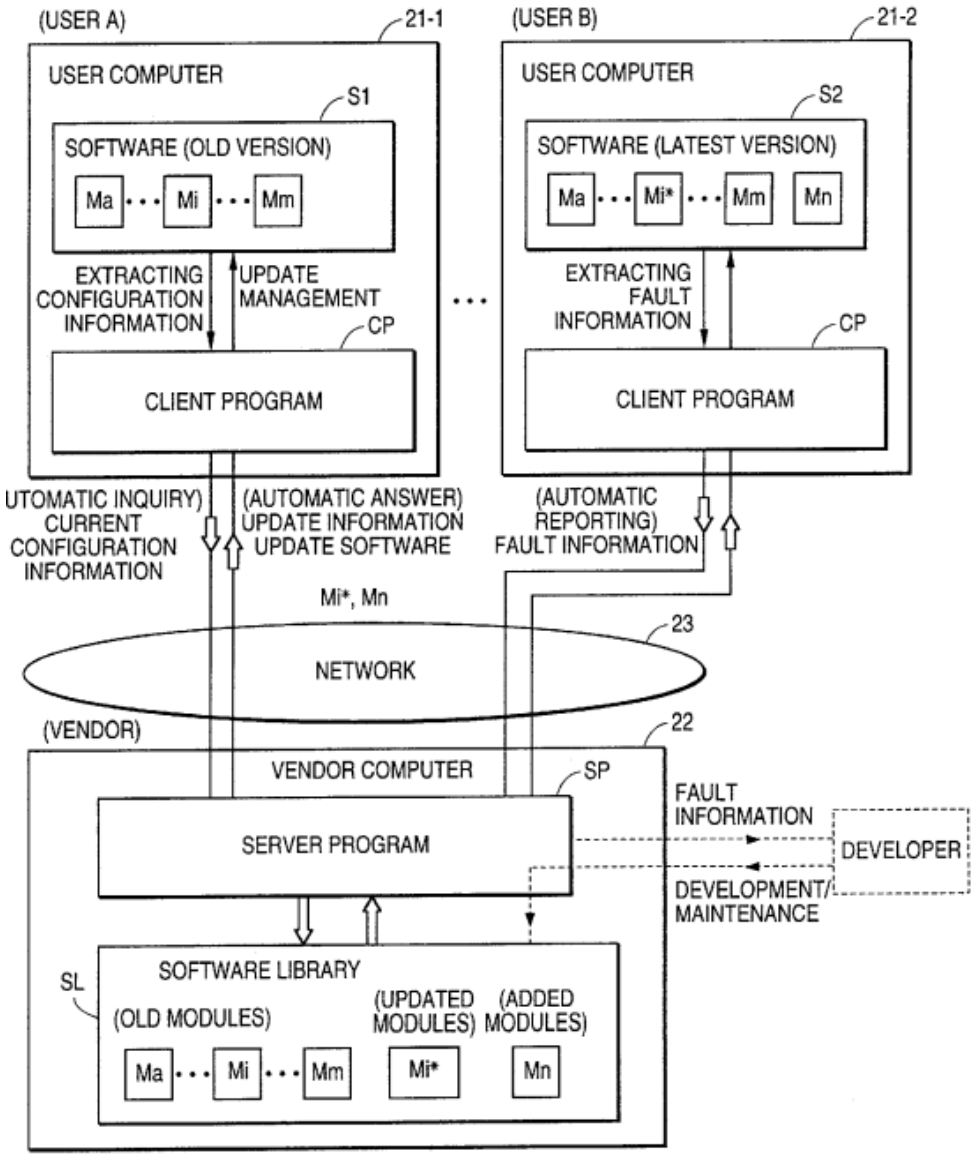
B. Ground 2: Claim 72 Is Rendered Obvious by Malamud and Nakagawa

111. In my opinion, the teachings of Malamud and Nakagawa render claim 72 of the ’102 Patent obvious.

1. Overview of Nakagawa

112. Nakagawa discloses methods for distributing software over a network such that a client can request and download software from a server. (EX1005 at Abstract, 1:13-16.) Nakagawa says that these methods are carried out in what was known as a "client/server method." (*Id*. at 3:1-5.) Nakagawa explains that these methods allow users to "quickly and properly obtain and use [] software" from a server without needing to visit stores and purchase software that is stored "in portable media such as magnetic tapes, floppy disks, etc." (*Id*. at 5:23-26, 1:23-33.) An exemplary system comprising a vendor computer 22 including a server program SP (server) for downloading software to user computers 21-1, 21-2 (clients) is provided in Figure 3 below.

FIG. 3



(*Id*. at Fig. 3.)

113. Nakagawa explains that by the time he filed his application in August 1995, there was "well-known prior art to distribute and obtain software update information" for a user's computer from a server. (*Id*. at 1:21-22, 1:34-37.) Nakagawa explains that such systems could be used by a vendor to distribute

software to customers over a network, or that a user could obtain software over a network by accessing a software library stored remotely for download. (*Id*. at 1:34-41, 1:46-50.) Nakagawa further explains that an executable program can be compiled on a vendor's computer and then downloaded by the user's computer, or the source programs can be downloaded and then compiled into an executable program on the user's computer. (*Id*. at 1:66-2:6.)

114. Nakagawa also discusses various improvements for distributing updates for such software using same client server systems and communication methods. (*Id*. at 5:15-6:15.)

2. Motivations to Combine Malamud and Nakagawa

115. A POSITA would have been motivated to combine Malamud and Nakagawa and would have had a reasonable expectation of success in doing so for several reasons.

116. As disclosed in Nakagawa, the distribution of software applications from a vendor to individuals that had purchased the software could be accomplished using portable media such as a floppy disk, or the software could be made available to the user by hosting the software in a software library on a network server. If the software is made available on a network server, users could request and download application programs over the internet. (EX1005 at 1:21-37, 1:46-49, 1:52-54, 3:1-5, 87:59-63.) There were several advantages of having users download software

from a software library on a network server. First, it was easy for the company that makes the software to ensure that the latest version of the software was always available on the software library. This would ensure that users downloading software would have the most recent versions of the software. Secondly, downloading software from the internet, while generally much slower than can be accomplished today with faster networking speeds, was likely significantly faster than having software mailed to the user via floppy disks or CD-ROMs. For the reasons discussed in the preceding Ground, a POSITA would have been motivated to distribute application programs, including Malamud's application program containing the cursor modification functionality, from a server to users over a network, as disclosed in Nakagawa.

117. A POSITA therefore would have been motivated to combine the teachings of Malamud and Nakagawa to distribute Malamud's application program to users over a network from a server, and it would have been well within the technical skill of a POSITA to do so with a high likelihood of success. This combination of Malamud combined with the teachings of Nakagawa, regarding software distribution from a server would not have been innovative, but rather the result of ordinary skill and combined with common sense.

3. Claim 72

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

118. In my opinion, Element 72[Preamble] would have been obvious to a POSITA in view of Malamud and Nakagawa.

119. As discussed above in Section V[A][2], Element 72[Preamble], Malamud discloses a method for modifying an initial cursor image on a display of a user's terminal.

120. Malamud is silent on whether the user's computer is connected by a network to a server, but for reasons discussed above, a POSITA would have found it obvious in view of Nakagawa and for the reasons set forth in paragraphs 88-93, 115-117 above.

121. In combination with the reason set forth above, a POSITA would been motivated to download Malamud's application program from a server in view of Nakagawa. (EX1005.) Nakagawa explains the client/server method for conveying information, including software, from a server to a client. (*Id*. at 3:1-5, 1:62-2:6.) Nakagawa discloses a user computer connected over a network to a server so that a user can request and download software from a server, as well as downloading updates to the software. (*See, e.g.*, *id*. at Abstract ("A number of sets of software may be systematically distributed and maintained via a network connecting many

vendors and users of client/server software."); *id*. at 1:13-16 ("The present invention relates to a software distribution and maintenance system and method with which a software vendor can provide a number of users with software over a network . . . ."); *id*. at 1:46-54 (describing a user's ability to download software from a vendor at the user's request).) As discussed above, a POSITA would have recognized the benefits of downloading Malamud's application program from a server and therefore would have been motivated to do so, and it would have been obvious to do so in view of Malamud and Nakagawa.

122. Thus, it would have been obvious to a POSITA to use Malamud's method for modifying a cursor image in conjunction with Nakagawa's disclosure of downloading software to a user's computer through a network connected to a server.

<u>Element 72[a]:</u> receiving a request at said at least one server to provide specified content information to said user terminal;

123. In my opinion, Element 72[a] would have been obvious to a POSITA in view of Malamud and Nakagawa.

124. As discussed in Section V[A][2], Element 72[a], Malamud teaches specified contention information in the form of an application program that includes information for use in the display on the user's terminal.

125. Malamud is silent as to how the data processing system obtains the application program, but it was well-known to a POSITA by June of 1997 that a user could request to download an application program from a server, as disclosed in

Nakagawa. (Ex. 1005 at 3:1-9.) In Section V[A][2] above, I describe how network interface cards (NICs) became common in personal computers (PCs) during the late 1980s and early 1990s. I also described how the spread of the internet in the mid-1990s that caused network cards became practically ubiquitous in PCs, as accessing the internet requires some form of network interface. This was when Ethernet started becoming a standard for local area networks (LANs) in organizations.

126. In addition, Nakagawa discloses a user computer connected over a network to a server so that a user can request and download software from a server, as well as downloading updates to the software. (*See, e.g.*, *id*. at Abstract ("A number of sets of software may be systematically distributed and maintained via a network connecting many vendors and users of client/server software."), 1:13-16 ("The present invention relates to a software distribution and maintenance system and method with which a software vendor can provide a number of users with software over a network . . . ."), 1:46-54 (describing a user's ability to download software from a vendor at the user's request).)

127. A POSITA would also have been motivated to download application programs, such as the one described in Malamud, from a server, as described in Nakagawa, for the reasons set forth in Section V[B][2] and Section V[B][3], Element 72[Preamble].

128. Thus, it would have been obvious to a POSITA to combine Malamud's teaching of an application program for modifying a cursor image with the Nakagawa's disclosure of the well-known procedure of downloading an application program from a server. That combination renders this claim limitation obvious, as the request to download Malamud's application program from a server would be a request to a server to provide specified content information to a user terminal.

> **<u>Element 72[b]:</u> 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**

129. In my opinion, Element 72[b] would have been obvious to a POSITA in view of Malamud and Nakagawa.

130. As discussed in conjunction with claim Element 72[a], it would have been obvious in view of Malamud and Nakagawa for a user to request from a server and receive in response an application program. The application program comprises specified content information because the application program includes information regarding the display of elements depicted by the application program, such as objects to which the cursor can point and modified cursors (i.e., information cursors), and it would include at least one cursor display instruction and an indication of cursor image data corresponding to a specific image. (*See* § V[A][2] at Element 72[b].)

131. It would have been obvious to a POSITA to download Malamud's application program (specified content information) from a server as disclosed in Nakagawa in response to a request (*see* § V[B][2]; § V[B][3], Element 72[Preamble]). Malamud's application program includes a cursor display instruction in the program, which is checked when a cursor is placed over an object on the display screen, and the instruction contains information that is passed to the operating system to cause the cursor image to change to an information cursor, such as a preview cursor. (*See* EX1004 at 5:47-62 (describing the message that the windows procedure of the application program passes, which includes the type of information cursor to display and information about the contents of the information cursor image).) For example, the application program in Malamud contains instructions whether to display an information cursor, and if so the type of information cursor (such as a preview cursor or a combined name and preview cursor) for each entity displayed in the application window, and based on those instructions in the application program, messages are sent to the OS regarding the display of the information cursors. (*Id*. at 3:59-4:18; 5:24-45; 5:46-62.) Malamud also teaches that, for a preview cursor, the application program transmits, and therefore contains, "a pointer to a bitmap of graphical information that the operating system 22 should use in the preview portion 36." (*Id*. at 5:57-62.) The bitmap is a "specific image" that is displayed as a modified cursor image, and it relates to

whatever object on the display screen that the cursor is pointing because it provides further information about that object. (*Id*. at 5:57-62, 3:59-4:3.) For example, Malamud teaches that if the cursor is pointing to a book, the bitmap can constitute an image that provides further information about the book. (*Id*. at 3:65-4:3.)

132. Thus, this limitation would have been obvious to a POSITA in view of Malamud and Nakagawa.

> **<u>Element 72[c][i]:</u> 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**

133. As addressed in Section V[A][2], Element 72[c][i], Malamud teaches this claim limitation and/or renders it obvious.

> **<u>Element 72[c][ii]:</u> 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.**

134. As addressed in Section V[A][2], Element 72[c][ii], Malamud teaches this claim limitation and/or renders it obvious.

135. Thus, claim 72 of the '102 Patent would have been obvious to a POSITA in view of Malamud and Nakagawa.

C. Ground 3: Claim 72 Is Rendered Obvious by Nielsen and Malamud

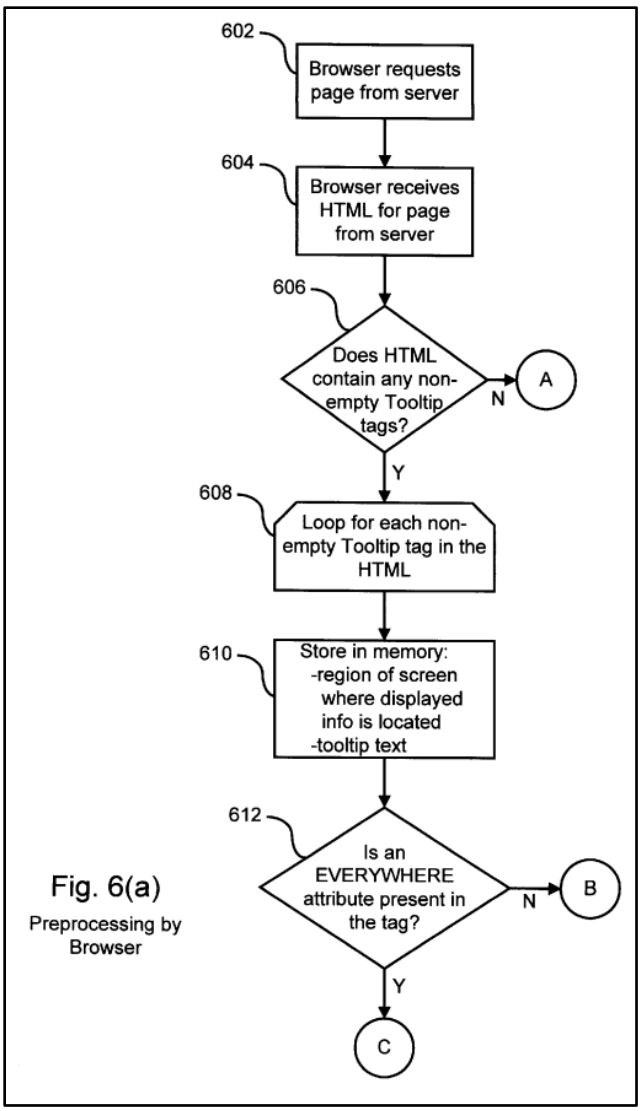
136. In my opinion, the teachings of Nielsen and Malamud render claim 72 of the ’102 Patent obvious.

1. Overview of Nielsen

137. Nielsen discloses automatically displaying a “tooltip” when a user places a cursor over predetermined text on a web page. (EX1006 at Abstract). Tooltips can take the form of “text,” “an icon,” or “other illustration” that is displayed to the user when the cursor is placed over “specified text,” “images,” “character strings,” or “animations.” (*Id*. at Abstract, 3:46-4:7.)

138. Tooltips can be implemented into a web page using the HTML format. (*Id*. at Abstract, 1:51-60.) Nielson discloses adding an extension to the HTML format that allows a web page designer to specify the text, icon, or illustration that is displayed when a cursor hovers over a specified text, image, character string, or animation. (*Id*. at Abstract, 1:51-60, 3:46-4:7.)

139. Figures 6a, 6b, and 7 provide flowcharts describing methods for modifying a cursor using a conventional client-server architecture.


602
Browser requests page from server
604
Browser receives HTML for page from server
606
Does HTML contain any non-empty Tooltip tags?
N
A
Y
608
Loop for each non-empty Tooltip tag in the HTML
610
Store in memory:
-region of screen where displayed info is located
-tooltip text
612
Is an EVERYWHERE attribute present in the tag?
N
B
Y
C
Fig. 6(a)
Preprocessing by Browser

(*Id*. at Fig. 6(a).)

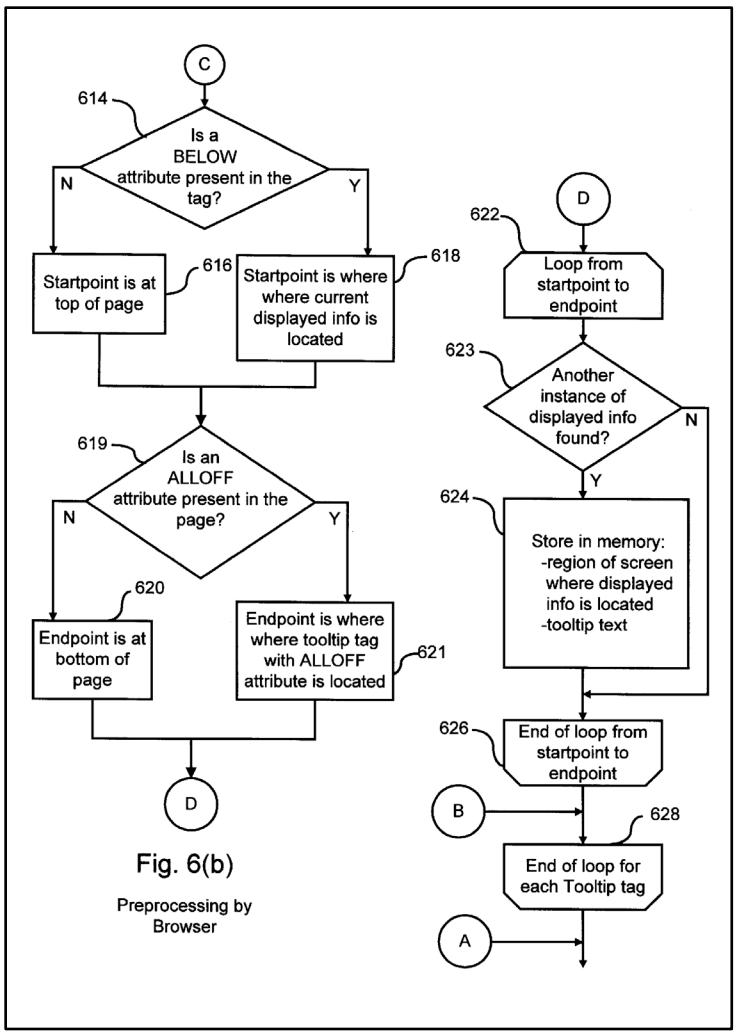


Fig. 6(b)

Preprocessing by Browser

(*Id*. at Fig. 6(b).)

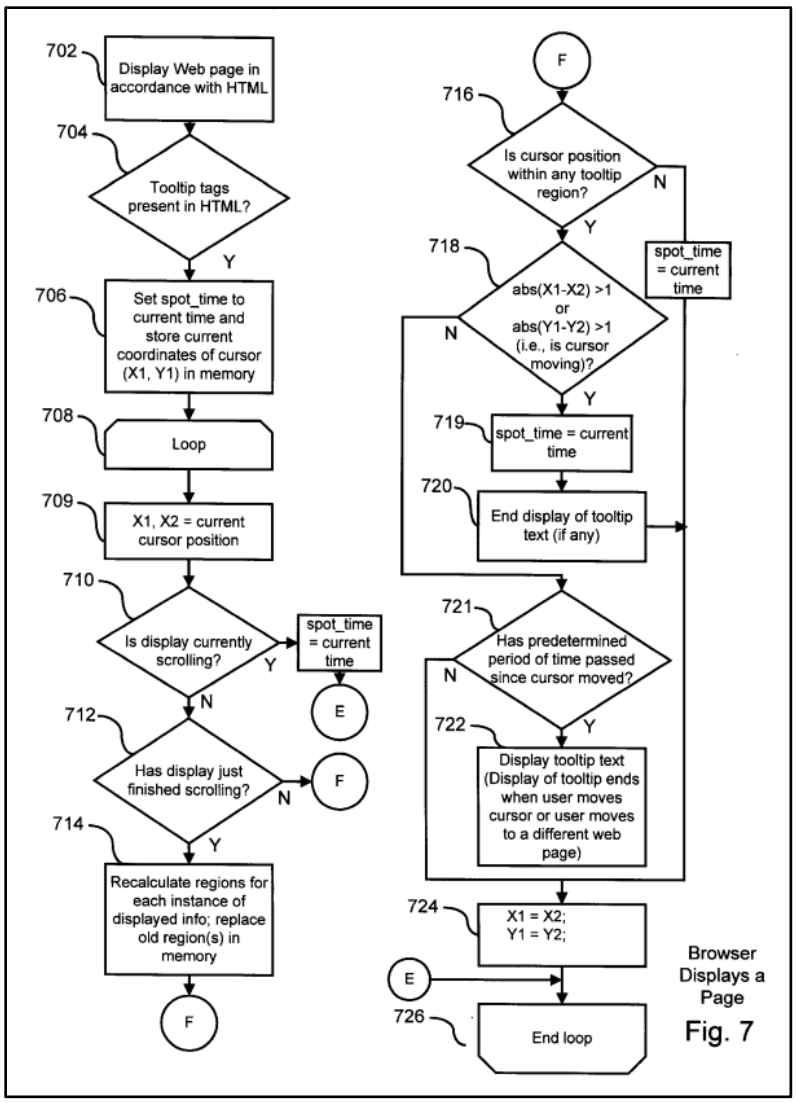


Fig. 7

(*Id*. at Fig. 7.)

140. With reference to Figure 6a, the browser on the user's display device requests (step 602) and receives HTML code for a web page from a server (step 604). After performing certain pre-processing steps, the browser performs further steps described in Figure 7. For example, the browser displays the web page in accordance with the received HTML code (step 702) and determines whether there are tooltip tags present in the HTML code (step 704). If the current cursor position

is within a tooltip region and certain other conditions are met (e.g., the cursor has not moved for a predetermined period), the tooltip text associated with the current cursor location is displayed on the browser (step 722).

141. An exemplary display of tooltip text is described with respect to Figures 3 and 4 provided below. Figure 3 shows an exemplary web page displayed on a user's display device. (*Id*. at 4:36-37.) The "From City" column includes the entry "LHR," though it may not be clear what "LHR" stands for. (*Id*. at 4:40-43.) Figure 4 shows the web page after the user places their cursor over the text "LHR." (*Id*. at 4:46-49.) In Figure 4, the tooltip text "London Heathrow" is displayed to the user in proximity to the "LHR" text, indicating that "LHR" is an abbreviation for "London Heathrow." (*Id*. at 4:44-55, 7:7-16.) The tooltip text continues to be displayed until the cursor is moved or until the user navigates to a different web page. (*Id*.)

STATUS

ACCOUNT SUMMARY

FIG. 3

Name: Dr. Jakob Nielsen
Userid: jn

Point Balance: 184243 as of 03/11/96

Points Earned Due to Flight Mileage

	Carrier	Flight Date	Flight Number	From City	To City	Flight Miles	Bonus Miles	Total Miles
302	UA	15-FEB-96	0931	LHR	SFO	5368	6710	12078
	UA	01-FEB-96	0835	SFO	HKG	6914	8643	15557
	UA	16-DEC-95	0073	EWR	SFO	2565	2565	5130
	UA	09-DEC-95	0036	SFO	BOS	2704	2704	5408
	UA	21-NOV-95	0806	HKG	SFO	6914	8643	15557
	UA	15-NOV-95	0835	SFO	HKG	6914	8643	15557
	U+	15-NOV-95	2936	LAS	SFO	414	621	1035
304	U+	12-NOV-95	2188	SFO	LAS	414	414	828
	UA	10-NOV-95	0931	LHR	SFO	5368	6710	12078

(*Id*. at Fig. 3 (showing web page before display of tooltip text, annotated with the displayed object pointed to by the cursor highlighted in green).)

STATUS

ACCOUNT SUMMARY

FIG. 4

Name: Dr. Jakob Nielsen
Userid: jn

Point Balance: 184243 as of 03/11/96

Points Earned Due to Flight Mileage

London Heathrow

	Carrier	Flight Date	Flight Number	From City	To City	Flight Miles	Bonus Miles	Total Miles
302	UA	15-FEB-96	0931	LHR	SFO	5368	6710	12078
	UA	01-FEB-96	0835	SFO	HKG	6914	8643	15557
	UA	16-DEC-95	0073	EWR	SFO	2565	2565	5130
	UA	09-DEC-95	0036	SFO	BOS	2704	2704	5408
	UA	21-NOV-95	0806	HKG	SFO	6914	8643	15557
	UA	15-NOV-95	0835	SFO	HKG	6914	8643	15557
	U+	15-NOV-95	2936	LAS	SFO	414	621	1035
304	U+	12-NOV-95	2188	SFO	LAS	414	414	828
	UA	10-NOV-95	0931	LHR	SFO	5368	6710	12078

(*Id*. at Fig. 4 (showing web page after display of tooltip text, annotated to highlight the tooltip in green).)

2. Motivations to Combine Nielsen and Malamud

142. Nielsen teaches a system and method for displaying tooltips on a web page that has been downloaded from a server. (*Id*. at Abstract, 1:51-57.) The tooltip is displayed when the cursor hovers over text or a graphical image on the user's display screen, and the tooltip provides additional information in the form of text or

graphical information about the object that the cursor is hovering over. (*Id.* at 1:51-57, 3:35-41, 3:61-63, 3:65-4:4.) In addition, the tooltip is displayed in proximity to the object that the cursor is hovering over. (*Id*. at 6:28-32, 7:8-14.)

143. Malamud is a contemporary reference to Nielsen and from the same field of endeavor, as Malamud and Nielsen both teach a system and method for displaying additional information about an object on a display screen that a cursor is hovering over. (EX1004 at Abstract, 1:35-42.) Like Nielsen, Malamud teaches that the additional information that is displayed can be in the form of text or a graphical image. (*Id*. at 2:66-3:6, 3:59-4:18.) Thus, Nielsen and Malamud address similar problems and provide similar solutions, with both patents describing solutions in the same field of human-computer interaction and user interface design.

144. While Nielsen teaches that the additional information is displayed in proximity to, such as above, the object to which the cursor is pointing, that may or may not be close to the current cursor position, and therefore may or may not be where the user's attention is usually focused (i.e., the cursor location). When a user is looking at the cursor, the user could miss information displayed elsewhere on a screen. Malamud teaches that the additional information is displayed as part of a modified cursor image, and therefore where the user's attention is likely focused. (*Id*. at Abstract, 3:59-4:18.) A POSITA, therefore, would have been motivated to

modify the teaching of Nielsen with Malamud's teaching of displaying the additional information as part of a modified cursor image.

145. For example, Nielsen teaches displaying a tooltip in proximity to the object that the cursor is hovering over, such as above the object. (EX1006 at 7:8-14.) But if that object was a large object that takes up a significant portion of a display screen and the cursor were to hover over the bottom portion of such an object, and a tooltip were displayed above the object, the user might easily miss the tooltip that was displayed as the user's attention is likely focused on the cursor location on the screen. By making Nielsen's tooltip appear as part of a modified cursor image, the additional information (i.e., text and/or graphics) would appear at the location on the screen where the user's attention is most likely focused. Therefore, a POSITA would have been motivated to combine the teachings of Nielsen and Malamud in this manner. Moreover, such a modification would have been well within the technical skill level of a POSITA, as it could have been implemented by processing Nielsen's tooltip instruction in a manner to display the tooltip information as part of a modified cursor as disclosed in Malamud. Such a modification would have had a high chance of success, as the software is not complex, with the software functioning according to its programming.

3. Claim 72 Is Obvious in View of Nielsen and Malamud

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

146. In my opinion, Element 72[Preamble] would have been obvious to a POSITA in view of Nielsen and Malamud.

147. Nielsen discloses a method for displaying a tooltip on a display of a user terminal connected to at least one server. (*Id*. at Abstract, 1:18-26, 1:51-57.) The tooltip, which can be comprised of text and/or graphics and relate to the object to which the cursor is pointing, is displayed in proximity to the object displayed to which the cursor is pointing. (*Id.* at 3:37-41, 3:52-56, 3:61-63, 7:8-12, 7:19-22.) Malamud also teaches displaying text and/or graphic relating to the object to which the cursor is pointing. (EX1004 at Abstract, 1:40-42, 3:59-4:17.) Malamud teaches that this text and/or graphic can be part of a cursor that consists of the original cursor (e.g., an arrow) and the additional text and/or graphic. (*Id*. at 3:59-4:3.) A POSITA would have been motivated to modify the teachings of Nielsen with the teachings of Malamud such that the tooltip is not just displayed in proximity to the object to which the cursor is pointing, but rather is displayed as part of a modified cursor image to ensure that user sees the supplemental information or graphic. (*See* § V[C][2].)

148. Thus, it would have been obvious for a POSITA to modify the teachings of Nielsen in order to change an initial cursor image into a modified cursor image that includes the tooltip for the reasons cited above.

<u>Element 72[a]:</u> receiving a request at said at least one server to provide specified content information to said user terminal;

149. In my opinion, Element 72[a] is disclosed in Nielsen and/or would have been obvious to a POSITA in view of Nielsen and Malamud.

150. Nielsen teaches receiving a request at server to provide specified content information. Specifically, Nielsen teaches that a user requests a web page from a server, and the server sends the HTML code for the requested web page to the user's computer. (EX1006 at 1:9-26, 1:51-57.) The HTML code for the web page includes information provided to a user's terminal for use in the display, including the web page content to be displayed as well as tooltip display instructions for displaying the tooltips. (*Id*. at 1:18-26, 1:51-57, 3:36-41, 3:46-4:7.) As discussed above, it would have been obvious to a POSITA to modify Nielsen's tooltip display instructions to make them cursor display instructions, as taught by Malamud, and a POSITA would have been motivated to make such a modification. (*See* § V[C][3] at Element 72[Preamble]; *see also* § V[C][2] (regarding motivation to combine the teachings of Nielsen and Malamud).)

<u>Element 72[b]:</u> 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

151. In my opinion, Element 72[b] would have been obvious to a POSITA in view of Nielsen and Malamud.

152. Nielsen teaches that the server provides the specified content information in the form of HTML code for a web page to the user's computer in response to the user's request for the web page. (EX1006 at 1:12-14, 1:18-26.) The HTML code for the web page is specified content regarding the web page information to be displayed, and also includes instructions regarding the display of tooltips, such as indication whether any tooltips are to be displayed, what is to be displayed, and the information on the user's screen to which the cursor must point to invoke the display of the tooltips. (*Id*. at 1:51-62, 3:35-41, 3:46-4:7.) While Nielsen's tooltips are displayed as separate images, it would have been obvious to a POSITA to display the tooltips as part of a modified cursor image, and a POSITA would have been motivated to do so. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).) Therefore, it would have been obvious to modify Nielsen's teachings to make the tooltip display instruction a cursor display instruction, which would modify the indication of a bitmap for a

graphical tooltip image into an indication of cursor image data corresponding to a specific image.

> **<u>Element 72[c][i]:</u> 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**

153. In my opinion, Element 72[c][i] would have been obvious to a POSITA in view of Nielsen and Malamud.

154. Nielsen teaches transforming the display on the user's terminal to display a tooltip when the cursor points to an object on the display screen that has an associated tooltip. (EX1006 at Abstract, 1:51-62, 3:35-41.) While Nielsen's tooltip is displayed as a separate image, it would have been obvious to a POSITA that the tooltip could be displayed as part of a modified cursor image as taught in Malamud, and a POSITA would have been motivated to modify the teachings of Nielsen in that manner. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).) The modified cursor image has the shape and appearance of the specific image specified by the bitmap and would be displayed in response to the tooltip instruction when the cursor is placed over an object on the screen that has an associated tooltip (EX1006 at 3:35-41, 3:46-4:7), and the tooltip instruction would be a cursor display instruction that would control the appearance of what is

displayed as well as the conditions of when it is displayed as part of a modified cursor image by combining the teachings of Nielsen and Malamud.

155. The specified content information received from the server in Nielsen (i.e., the HTML code for the web page, including the tooltip instructions), includes the information that is displayed on the user's computer, in the form of the web page and the tooltips. (*Id*. at 1:12-26, 1:51-57, 1:66-2:9.) Nielsen discloses that the displayed tooltip relates to and provides further information about an object on the computer display screen pointed at by the cursor. (*Id*. at 1:54-62, 3:35-41.) Similarly, Malamud teaches that the information cursor, such as a preview cursor or a combined name and preview cursor, also relate to the object on the computer display screen to which the cursor is pointing. (EX1004 at Abstract, 1:35-43, 2:9-12, 2:66-3:6.)

156. Therefore, this limitation would have been obvious to a POSITA in view of the teachings of Nielsen and Malamud, and POSITA would have been motivated to combine the teachings of those references for the reasons cited above. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).)

> **<u>Element 72[c][ii]:</u> 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.

157. In my opinion, Element 72[c][ii] would have been obvious to a POSITA in view of Nielsen and Malamud.

158. Nielsen teaches HTML code for web pages, where the code contains tooltip instructions. (EX1006 at Abstract, 1:51-62.) The tooltip instruction includes information regarding whether a tooltip is to be displayed as well as the type of tooltip to displayed, such as text or a graphical image. (*Id*. at 3:35-41, 3:46-63.) The tooltip therefore indicates the web browser code operable to process the instruction to display the appropriate type of tooltip. (*See, e.g.*, *id*. at 1:36-37 (discussing that the browser software displays the tooltip as specified by the web page designer); *id*. at 3:54-56 (identifying the HTML code that indicates the presence of a tooltip that must be processed by the appropriate browser code); *id*. at 3:61-63 (identifying that a tooltip can also indicate that the tooltip should be displayed as an icon or other illustration); 5:20-58 (describing the processing that is performed by the browser code when a tooltip instruction is detected).) Nielsen further teaches an algorithm that is implemented by code to display the tooltip appropriately. (*Id*. at 5:18-6:20; *see also* Fig. 6(a) (step 606 indicating that the presence of a tooltip instruction indicates what portion of the algorithm, implemented by code, should be used to process the HTML code).)

159. The processing of Nielsen's tooltip instruction in the HTML code causes a tooltip to be displayed in response to the user placing the cursor over an object on the displayed web page, where the object has an associated tooltip. (*Id*. at 3:35-41.) The tooltip takes the shape of the specific image in the form of an icon or other illustration as identified by the tooltip instruction and related to an image displayed on the screen and to which the cursor is pointing. (*Id*. at 3:35-41, 3:61-63.)

160. While Nielsen's tooltip is displayed as a separate image, as described previously, it would have been obvious to a POSITA that the tooltip could be displayed as part of a modified cursor image as taught in Malamud, and a POSITA would have been motivated to modify the teachings of Nielsen in that manner. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).) Modifying the teachings of Nielsen in such a manner would result in Nielsen's tooltip instruction being a cursor display instruction, Nielsen's tooltip display code being cursor display code, and Nielsen's tooltip being a modified cursor image in the shape and appearance of the specific image.

161. Thus, the combination of Nielsen and Malamud renders claim 72 obvious.

VI. GROUNDS FOR INVALIDITY OF THE '449 PATENT

A. Ground 1: Claims 1, 27, 53 of the '449 Patent are Rendered Obvious by Malamud

1. Motivation to Download Malamud's Application Program from a Server, and the Obviousness of Doing So

162. I discussed the reasons a POSITA would have been motivated to download Malamud's application program from a server in connection with the '102 Patent. (*See* Section V[A][2], Element 72[Preamble].) That same motivation applies here for the '449 Patent.

2. Claim 1

<u>Element 1[Preamble]:</u> 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:

163. In my opinion, Element 1[Preamble] would have been obvious to a POSITA in view of Malamud.

164. Malamud teaches a system for modifying on the display of a user terminal an initial cursor image, such as a "conventional cursor," to a specific cursor image that includes a graphical image relating to the object to which the cursor is currently pointing. Specifically, Malamud teaches that when the cursor is over a location on the display screen where no named entity is present, a "conventional cursor" image or an "information cursor" without any corresponding specific information is displayed. (EX1004 at 5:39-44.) When the cursor is moved over a

named entity displayed on the screen, the cursor image is modified to an "information cursor" that includes text and/or a graphical image related to the object on the display screen to which the cursor is pointing. (*Id*. at 5:32-39; *id*. at 2:66-3:1 ("The present invention provides 'information cursors' which display graphical or textual information about an object to which the cursor points."); *id* at 3:59-4:3 (discussing a "preview cursor," which depicts a graphical image object the object on the display screen to which the cursor points).) The modified cursor image is a specific image that has a desired shape and appearance as specified by the bitmap and relating to the information pointed to by the cursor. (*Id*. at 3:59-4:3, 5:57-62.)

165. Malamud does not address whether the user's computer is connected by a network to a server, but connecting user computers to a server was well known to POSITAs and commonplace well before June 1997. (EX1005 at 3:1-5; *see* § V[A][2], Element 72[Preamble].) For example, it was quite common by that time to have computers connected to one or more servers in many different contexts, such as: in an enterprise, where employee computers were connected to company servers; in an educational setting, where student computers were connected to school servers; or in a home, where an individual's computer would connect to internet servers. The inventors of the '449 Patent admit that, by 1997, millions of people all over the world were using personal computers to access digital content over the internet. (EX1001 at 1:9-19.) Thus, it would have been obvious to a POSITA to use Malamud's system

for modifying a cursor image on a computer connected to a server, and a POSITA would have been motivated to do so for the reasons discussed above (*see* § V[A][2], Element 72[Preamble]).

<u>Element 1[a]:</u> cursor image data corresponding to said specific image;

166. In my opinion, Element 1[a] is disclosed in Malamud.

167. Malamud teaches cursor image data corresponding to the specific image in the form of a "bitmap of graphical information that the operating system 22 should use in the preview portion 36." (EX1004 at 5:59-62, Fig. 3.) The bitmap is data for a modified cursor image in the form of Malamud's preview cursor or combined name and preview cursor (*id*. at 3:59-4:18), and the bitmap image is a specific image that corresponds to a portion of the subject or topic being displayed on the screen, such as providing an image related to a book being pointed to by the cursor (*id*. at 3:61-4:3, Fig. 3).

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

168. In my opinion, Element 1[b] is disclosed in Malamud.

169. Malamud discloses that the window procedure of the application program passes a message to the operating system 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 5:47-53; *see also id*. at 3:6-8

("Information cursors are made available by an operating system to applications that are run on the operating system.").) The code that interprets those instructions and causes the cursor image to change from the original cursor shape to the shape of the information cursor is "cursor display code," as it is "operable to modify said cursor image," as recited in the limitation. (*See* EX1008 at Ex. 1, p. 1 (Patent Owner asserting that "cursor display code" means "computer code for modifying the display of the cursor image").)

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

170. In my opinion, Element 1[c][i] would have been obvious to a POSITA in view of Malamud.

171. Malamud does not address how the user's computer obtains Malamud's application program, but transmitting application programs from a server to a user's computer was well known to POSITAs and commonplace well before June 1997. (*See* § V[A][2], Element 72[Preamble].) Thus, it would have been obvious to a POSITA to download Malamud's application program from a server and a POSITA would have been motivated to do so for the reasons discussed above (*see* § V[A][2], Element 72[Preamble]).

172. It is my understanding that the parties agree that "specified content information" means "information provided to a user's terminal for use in the display, such as a web page." (EX1008 at 3.) Malamud's application program comprises

"specified content information" because it includes information that is displayed on a user's computer because it generates the content displayed in the application window. Examples would be the book icon 32 pointed to by the cursor and the preview cursor image. (EX1004 at 3:36-39, 3:59-4:18, Fig. 4.)

> **<u>Element 1[c][ii]:</u> 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,**

173. In my opinion, Element 1[c][ii] is disclosed in Malamud.

174. Malamud teaches a "cursor display instruction" in the form of instructions in the application that indicate whether to display an information cursor, and if so, the type of information cursor to display. (EX1004 at 5:46-6:16.) Malamud's cursor display instruction for a preview cursor or a combined name and preview cursor includes a pointer to a bitmap for the preview image. (*Id*. at 5:46-6:12.) Malamud's cursor display instruction and cursor display code together cause the user's computer to display a modified cursor image in the form of a preview cursor or a combined name and preview cursor, which is in the shape and appearance of the specific image. (*Id*. at 3:50-4:18, 5:5:47-62.)

> **<u>Element 1[c][iii]:</u> 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,

175. In my opinion, Element 1[c][iii] would have been obvious to a POSITA in view of Malamud.

176. As discussed in regard to Elements 1[Preamble] and 1[c][i], it would have been obvious to a POSITA for a user to request and receive Malamud's application program from a server. And as further discussed in regard to Element 1[c][i], Malamud's application program comprises specified content information that includes information that is to be displayed on the user's computer, such as the image 32 of the book icon being pointed to by the cursor (EX1004 at Fig. 4).

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

177. In my opinion, Element 1[c][iv] is disclosed in Malamud.

178. Malamud's preview cursor and combined name and preview cursor include content corresponding at least a portion of the information displayed on the user's screen, such as an image that relates to the book icon. (*Id*. at 3:59-4:18.) As discussed in regard to Element 1[c][ii], the cursor display code is used to process the

cursor display instruction to modify the cursor image into the shape of the specific image when the cursor is placed over a portion of the information displayed on the user's computer screen, and the specific image relates to at least a portion of the information displayed on the user's computer. (*See also* EX1004 at Abstract, 1:38-42, 2:10-12, 2:66-3:3:6, 3:50-4:18.)

179. Thus, claim 1 would have been obvious in view of Malamud.

3. Claim 27

180. Claim 27 is word-for-word identical to claim 1 except that claim 1 recites that the modified cursor image is displayed in response to movement of the 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" (EX1002 at 19:1-3), whereas claim 27 recites that the modified cursor image is displayed in response to movement of the cursor image over "a specified location on said display of said user's terminal" (*id*. at 20:63-65). Malamud teaches that when the cursor is moved, a message is transmitted from the operating system and identifying the location of the mouse on the screen, and that information is used by the application program to determine whether to display the modified cursor. (EX1004 at 5:3-7 ("The mouse messages specify the status of mouse buttons and the position of the cursor within the window. The position of the cursor within the window is specified in (X, Y) coordinates relative to the upper left-hand cover of the window."), 5:22-45).) In light of that

teaching in Malamud and for the reasons set forth above regarding claim 1, claim 27 would have been obvious to a POSITA in view of Malamud.

4. Claim 53

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

181. In my opinion, Element 53[Preamble] would have been obvious to a POSITA in view of Malamud.

182. Malamud teaches a method for modifying on the display of a user terminal an initial cursor image, such as a "conventional cursor," to a specific cursor image that includes a graphical image of the object to which the cursor is pointing. Specifically, Malamud teaches that when the cursor is at a location on the display screen where no named entity is present, either a "conventional cursor" image or an "information cursor" without any corresponding specific information is displayed. (EX1004 at 5:39-44.) When the cursor is moved over a named entity that is displayed on the screen, the cursor image changes to an "information cursor" that includes text and/or a graphical image related to the object on the display screen to which the cursor is pointing. (*Id*. at 5:32-39; *id*. at 2:66-3:1 ("The present invention provides 'information cursors' which display graphical or textual information about an object to which the cursor points."); *id* at 3:59-4:3 (discussing a "preview cursor,"

which depicts a graphical image object the object on the display screen to which the cursor points).)

183. Malamud is silent regarding whether the user's computer is connected by a network to a server, but connecting user computers to a server was very common and well known to POSITAs well before June 1997. (EX1005 at 3:1-5; *see* § V[A][2], Element 72[Preamble].) For example, it was quite common by that time to have computers connected to one or more servers in many different contexts, such as: in enterprise settings, where employee computers were connected to company servers; in educational settings, where student computers were connected to a school server; or in home settings, where an individual's computer would connect to company servers and/or internet servers. The inventors of the '449 Patent admit that, by 1997, millions of people all over the world were using personal computers to access digital content over the internet. (EX1001 at 1:9-19.) Thus, it would have been obvious to a POSITA to use Malamud's method for modifying a cursor image on a computer connected to a server, and a POSITA would have been motivated to do so for the reasons discussed above (*see* § V[A][2], Element 72[Preamble]).

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

184. In my opinion, Element 53[a] would have been obvious to a POSITA in view of Malamud.

185. Malamud teaches the use of information cursors, including a "preview cursor," by an application program. (EX1004 at Abstract ("Information cursors are provided for use in an operating system and/or *application programs*.") (emphasis added).) The application program includes content to be displayed on the screen when the program is run, and it includes instructions regarding the modified cursor, such as whether to display an information cursor, the type of information cursor to display, and the content of the information cursor. (*Id*. at 5:24-45, Figs. 3 and 4 (depicting book icon 32, which relates to the application running in that window), 4:56-59 (discussing windows displayed on the screen generated by various programs), 3:59-4:18 (discussing the "preview cursor" and the "combined name and preview cursor").) Thus, Malamud teaches specified contention information in the form of an application program that includes information for use in the display on the user's terminal.[10]

186. Malamud is silent as to how the data-processing system obtains the application program, but it was well-known to a POSITA by June 1997 that a user

[10] Petitioner and Patent Owner agree that the claim terms "specified content information" and "content information" mean "information provided to a user's terminal for use in the display, such as a web page." (EX1008 at 3; EX1001 at Abstract, 2:58-62, 3:4-9, 24:14-24.)

could request and download an application program from a server. (*See, e.g.*, EX1005 at 3:1-9.) A POSITA would also have been motivated to download application programs, such as the one described in Malamud, from a server for the reasons set forth in Section V[A][2], Element 72[Preamble]. Thus, it would have been obvious to a POSITA to combine Malamud's teaching of an application program for modifying a cursor image with the commonplace and well-known procedures associates with downloading an application program from a server. This combination renders this claim limitation obvious, as the request to download Malamud's application program from a server would be a request to a server to provide specified content information to a user terminal because, as discussed above, the application program includes information for use in the display on the user's computer.

> **<u>Element 53[b]:</u> 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**

187. In my opinion, Element 53[b] would have been obvious to a POSITA in view of Malamud. It is also my opinion that it would be obvious that the application program would contain cursor display instructions that instruct the performance of the functionality discussed below.

188. As discussed in conjunction with claim element 53[a], in view of Malamud, a POSITA would have found it obvious for a user to request and receive an application program from a server that contained specified content information because the disclosed application program includes information regarding the display of elements depicted by the application program, such as objects to which the cursor can point, as well as the modified cursors (i.e., information cursors).

189. The application program in Malamud includes at least one cursor display instruction, which is checked when a cursor is placed over an object on the display screen, and the instruction contains information that is passed to the operating system through a message to cause the cursor image to change to an information cursor, such as a preview cursor. (*See* EX1004 at 5:47-62 (describing the message that is passed describing the type of cursor to display and information about the contents of the modified cursor image).) For example, the application program in Malamud contains instructions whether to display an information cursor, and if so, the type of information cursor (such as a preview cursor or a combined name and preview cursor) for each entity displayed in the application window. (*Id*. at 3:59-4:18; 5:24-45; 5:46-62.) Malamud also teaches that, for a preview cursor, the application program transmits, and therefore contains, an indication of cursor image data in the form of "a pointer to a bitmap of graphical information that the operating system 22 should use in the preview portion 36." (*Id*. at 5:57-62.) The

bitmap is a "specific image" that is displayed as a modified cursor image, and it relates to an object on the display screen to which the cursor is pointing because it provides further information about the object. (*Id*. at 5:57-62, 3:59-4:3.) For example, Malamud teaches that if the cursor is pointing to a book, the bitmap can constitute an image that provides further information about the book. (*Id*. at 3:65-4:3.)

190. Given the disclosures discussed above by Malamud, this limitation would have been obvious to a POSITA when reviewing Malamud.

> **<u>Element 53[c][i]:</u> 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**

191. In my opinion, Element 53[c][i] is disclosed in Malamud.

192. Malamud teaches that when the cursor is not pointing to a named object in the window for the application, "either a conventional cursor is displayed or the information containing portion (e.g., name box 30, preview portion 36, or property box 40) of the information cursor is shown in blank (step 57)." (EX1004 at 5:39-44.) The conventional cursor or the information cursor when lacking any corresponding specific information is an "initial cursor image."

193. Malamud further teaches determining whether the cursor has been moved over an object displayed in the application's display window, where the object has an identity (i.e., an associated information cursor). (*Id*. at 5:32-34.) Malamud instructs that "[i]f a named entity is present at the specified cursor position, the information regarding the object at the specified cursor position is displayed in the information cursor (step 56)." (*Id*. at 5:36-39.) If the information cursor is a preview cursor, this modified cursor image will include a graphical image related to the object on the display screen to which the cursor is pointing. (*Id*. at 3:59-4:3.) That graphical image is the specific image contained in the bitmap. (*Id*. at 5:57-62.) This functionality is carried out according to an instruction and through code that causes a message to be passed from the application program's window procedure to the operating system 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 5:47-53.) Thus, Malamud teaches transforming the initial cursor image into the shape and appearance of the specific image in response to a cursor display instruction.

194. Malamud further teaches that the application program provides the information to be displayed in the window for that application. (*Id*. at 3:36-41, 3:63-4:3; *see also id*. at 5:24-30 (discussing the application program controlling the display in a particular window); *see, e.g.*, *id*. at Fig. 3 (depicting book icon 32

generated by the application program).) Thus, the "specified content information" received from the server in the form of the application program would include "information that is to be displayed on said display of said user's terminal."

195. As discussed above, the preview cursor (i.e., the modified cursor image) includes a graphical image related to the object in the application window to which the cursor is pointing. (*Id*. at 3:59-4:18.) For example, if the cursor is pointing to a book, the preview cursor would include an image related to the subject matter of the book. (*Id*.) Therefore, Malamud teaches that the "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."

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

196. In my opinion, Element 53[c][ii] is disclosed in Malamud and/or would have been obvious to a POSITA in view of Malamud.

197. The application program in Malamud includes at least one cursor display instruction, which is checked when a cursor is placed over an object on the display screen, and the instruction contains information that is passed to the operating system to cause the cursor image to change to an information cursor, such

as a preview cursor. (*See* EX1004 at 5:47-62.) By indicating the type of information cursor to be displayed, that instruction indicates what cursor display code[11] is needed to process the instruction and properly prepare and transmit the message to the operating system. (*See id*. at 5:46-62, Figs. 6-7.) For example, Malamud teaches cursor display code in the form of an algorithm that would implement portions of Figures 6 and 7. (*Id*. at Figs. 6-7.) The cursor display instruction would indicate whether there is an information cursor for a particular entity and, if so, the type of information cursor, and that would indicate what cursor display code should be used to process the instruction and properly convey the message to the operating system and display the cursor. (*Id*. at 5:46-62, Figs. 6-7 (which would be implemented in code).)

---

[11] In the District Court litigation, Petitioner contends that "cursor display code" should be given its plain and ordinary meaning. (EX1008 at Ex. 1, p. 1.) Patent Owner asserts that "cursor display code" means "computer code for modifying the display of the cursor image." (*Id*.) Malamud satisfies Patent Owner's proposed construction, as the code in Malamud that interprets and implements the instructions regarding whether to display an information cursor, the type of information cursor to display, and what graphical information to display (EX1004 at 5:46-62) is "computer code for modifying the display of the cursor image."

198. The application program also includes a pointer to a bitmap that is conveyed as part of the message and specifies the shape and appearance of the specific image (i.e., the image of the modified cursor). (*Id*.) Malamud teaches that the cursor image is modified in response to the movement of the cursor over an object displayed in the application window, and that object is "display of said at least a portion of said information to be displayed on said display of said user's terminal." (*Id*. at 5:24-39.)

199. While disclosed in Malamud, it also would have been obvious to a POSITA that the cursor display instruction indicates the cursor display code needed to process the cursor display instruction because the information in the instruction must be properly conveyed to the OS, and it was common to convey information to the OS through an API call. The cursor display instruction in Malamud instructs whether a particular entity has an associated information cursor and, if so, the type of information cursor and the text and/or preview image to display. Because different pieces of information need to be conveyed to the OS for different types of information cursors, such as text for information cursor that includes a name box or a pointer for an information cursor that includes a preview portion, it would have been obvious to a POSITA that the information contained in the instruction would indicate what code is needed to process the instruction to properly formulate and convey the message to the OS through the appropriate API call, as this was an

extremely common procedure. Therefore, it would have been obvious to a POSITA that the instruction, based upon the information contained therein, would indicate the code to use to process the instruction.

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

200. In my opinion, Element 53[c][iii] is disclosed in Malamud.

201. Malamud teaches that the preview cursor, which is a specific image as discussed above, has a shape and appearance relating to information to be displayed on the screen. (EX1004 at 1:39-43, 2:10-12, 3:59-4:18.)

202. A POSITA, when reviewing Malamud, also would have found it obvious that the specific image identified in the bitmap could be in a shape and appearance relating to information displayed on the screen given Malamud's teaching that the bitmap provides information related to the image displayed on the screen. (*Id.*)

203. Thus, claim 53 would have been obvious to a POSITA in view of Malamud.

B. Ground 2: Claims 1, 27, 53 of the '449 Patent Are Rendered Obvious by Malamud and Nakagawa

1. Motivation to Combine

204. I discussed the reasons a POSITA would have been motivated to combine Malamud and Nakagawa in connection with the '102 Patent. (*See*

§ V[B][2]; *see also* § V[B][3], Element 72[Preamble].) That same motivation also applies here.

2. Claim 1

<u>Element 1[Preamble]:</u> 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:

205. In my opinion, Element 1[Preamble] would have been obvious to a POSITA in view of Malamud and Nakagawa.

206. Malamud teaches a system for modifying on the display of a user terminal an initial cursor image, such as a "conventional cursor," to a specific cursor image that includes a graphical image of the object to which the cursor is pointing. Specifically, Malamud teaches that when the cursor is at a location on the display screen where no named entity is present, a "conventional cursor" image or an "information cursor" without any corresponding specific information is displayed. (EX1004 at 5:39-44.) When the cursor is moved over a named entity displayed on the screen, the cursor image is modified to an "information cursor" that includes text and/or a graphical image related to the object on the display screen to which the cursor is pointing. (*Id*. at 5:32-39; *id*. at 2:66-3:1 ("The present invention provides 'information cursors' which display graphical or textual information about an object to which the cursor points."); *id* at 3:59-4:3 (discussing a "preview cursor," which depicts a graphical image object the object on the display screen to which the cursor

points).) The modified cursor image is a specific image that has a desired shape and appearance as specified by the bitmap and relating to the information pointed to by the cursor. (*Id*. at 3:59-4:3, 5:57-62.)

207. Malamud is silent regarding whether the user's computer is connected by a network to a server, but connecting user computers to a server was very common and well known to POSITAs well before June 1997. (EX1005 at 3:1-5; *see* § V[B][2]; *see also* § V[B][3], Element 72[Preamble].) For example, Nakagawa discloses a user computer connected over a network to a server so that a user can request and download software from a server, as well as downloading updates to the software from a server. (*See, e.g.*, EX1005 at Abstract ("A number of sets of software may be systematically distributed and maintained via a network connecting many vendors and users of client/server software."); *id*. at 1:13-16 ("The present invention relates to a software distribution and maintenance system and method with which a software vendor can provide a number of users with software over a network . . . ."); *id*. at 1:46-54 (describing a user's ability to download software from a vendor at the user's request).)

208. Connecting user computers to a server was well known to POSITAs and commonplace well before June 1997. (*Id*. at 3:1-5; EX1010 at 158-160.) Network interface cards (NICs) became common in personal computers (PCs) during the late 1980s and early 1990s. This was when Ethernet started becoming a

standard for local area networks (LANs) in organizations. However, it was really with the spread of the internet in the mid-1990s that network cards became practically ubiquitous in PCs, as accessing the internet requires some form of network interface. For example, it was quite common by that time to have use a computer connected to one or more servers in many different contexts, such as: in an enterprise, where employee computers were connected to company servers; in an educational setting, where student computers were connected to a school server; or in a home, where an individual's computer would connect to internet servers. The inventors of the '449 Patent even admit that, by 1997, millions of people all over the world were using personal computers to access digital content over the internet. (EX1001 at 1:9-19.) Additionally, a user would have been motivated to combine the teachings of Malamud and Nakagawa. (*See* § V[B][2].) Thus, it would have been obvious to a POSITA to use Malamud's method for modifying a cursor image in conjunction with Nakagawa's disclosure of a server system to create a server system for modifying a cursor image in accordance with the recitation in the preamble. (*See* § V[B][2]; *see also* § V[B][3], Element 72[Preamble].)

<u>Element 1[a]:</u> cursor image data corresponding to said specific image;

209. In my opinion, Element 1[a] is disclosed in Malamud.

210. This limitation is taught by Malamud for the same reasons set forth in Section VI[A][2] at Element 1[a].

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

211. In my opinion, Element 1[b] is disclosed in Malamud.

212. This limitation is taught by Malamud for the same reasons set forth in Section VI[A][2] at Element 1[b].

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

213. In my opinion, Element 1[c][i] would have been obvious to a POSITA in view of Malamud and Nakagawa.

214. Malamud does not address how the user's computer obtains Malamud's application program. Receiving application programs from a server, however, was well known to POSITAs and commonplace well before June 1997. (EX1005 at 3:1-5; § V[B][2]; *see also* § V[B][3], Element 72[Preamble].) For example, Nakagawa discloses a user computer connected over a network to a server so that a user can request and download software from a server, as well as downloading updates to the software. (*See, e.g.*, EX1005 at Abstract ("A number of sets of software may be systematically distributed and maintained via a network connecting many vendors and users of client/server software."); *id*. at 1:13-16 ("The present invention relates to a software distribution and maintenance system and method with which a software vendor can provide a number of users with software over a network . . . ."); *id*. at 1:46-54 (describing a user's ability to download software from a vendor at the user's

request).) The inventors of the '449 Patent even admit that, by 1997, millions of people all over the world were using personal computers to access digital content over the internet. (EX1001 at 1:9-19.) Additionally, a user would have been motivated to combine the teachings of Malamud and Nakagawa. (*See* § V[B][2].) Thus, it would have been obvious to a POSITA to download Malamud's application program from a server; and a POSITA would have been motivated to do so for the reasons discussed above (*see* § V[B][2]; *see also* § V[B][3], Element 72[Preamble]).

215. The parties agree that "specified content information" means "information provided to a user's terminal for use in the display, such as a web page." (EX1008 at 3.) Malamud's application program comprises "specified content information" because it includes information for use in the display on a user's computer because it generates the content displayed in the application window, for example the book cover and the preview cursor image. (EX1004 at 3:36-39, 3:59-4:18.)

> **<u>Element 1[c][ii]:</u> 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,**

216. In my opinion, Element 1[c][ii] is disclosed in Malamud.

217. This limitation is taught by Malamud for the same reasons set forth in Section VI[A][2] at Element 1[c][ii].

<u>Element 1[c][iii]:</u> 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,

218. In my opinion, Element 1[c][iii] would have been obvious to a POSITA in view of Malamud and Nakagawa.

219. As discussed in regard to Element 1[c][i], it would have been obvious to a POSITA for a user to request and receive Malamud's application program from a server using a client/server system as disclosed in Nakagawa. Also, as further discussed in regard to Element 1[c][i], Malamud's application program contains specified content information that comprises information to be displayed on the user's computer.

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

220. In my opinion, Element 1[c][iv] is disclosed in Malamud.

221. This limitation is taught by Malamud for the same reasons set forth in Section VI[A][2] at Element 1[c][iv].

222. Thus, claim 1 would have been obvious in view of Malamud and Nakagawa.

3. Claim 27

223. Claim 27 is word-for-word identical to claim 1 except that claim 1 recites that the modified cursor image is displayed in response to movement of the 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" (EX1002 at 19:1-3), whereas claim 27 recites that the modified cursor image is displayed in response to movement of the cursor image over "a specified location on said display of said user's terminal" (*id*. at 20:63-65). Malamud teaches that when the cursor is moved, a message is transmitted identifying the location of the mouse on the screen, and that information is used to determine whether to display the modified cursor. (EX1004 at 5:3-7 ("The mouse messages specify the status of mouse buttons and the position of the cursor within the window. The position of the cursor within the window is specified in (X, Y) coordinates relative to the upper left-hand cover of the window."), 5:22-45).) In light of that teaching in Malamud and for the reasons set forth above regarding claim 1, claim 27 would have been obvious to a POSITA in view of Malamud and Nakagawa.

4. Claim 53

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

224. In my opinion, Element 53[Preamble] would have been obvious to a POSITA in view of Malamud and Nakagawa.

225. As discussed above in Section VI[A][4], Element 53[Preamble], Malamud discloses a method for modifying an initial cursor image on a display of a user's terminal.

226. Malamud does not address whether the user's computer is connected by a network to a server, but connecting user computers to a server was commonplace and well known to POSITAs well before June 1997. (EX1005 at 3:1-5; *see* § V[B][2]; *see also* § V[B][3], Element 72[Preamble].) For example, Nakagawa discloses a user computer connected over a network to a server so that a user can request and download software from a server, as well as downloading updates to the software. (*See, e.g.*, EX1005 at Abstract ("A number of sets of software may be systematically distributed and maintained via a network connecting many vendors and users of client/server software."); *id*. at 1:13-16 ("The present invention relates to a software distribution and maintenance system and method with which a software vendor can provide a number of users with software over a network . . . ."); *id*. at 1:46-54 (describing a user's ability to download software from

a vendor at the user's request).) The inventors of the '449 Patent even admit that, by 1997, millions of people all over the world were using personal computers to access digital content over the internet. (EX1001 at 1:9-19.) Additionally, a user would have been motivated to combine the teachings of Malamud and Nakagawa. (*See* § V[B][2].) Thus, it would have been obvious to a POSITA to use Malamud's method for modifying a cursor image in conjunction with Nakagawa's disclosure of downloading software to a user's computer through a network connected to a server.

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

227. In my opinion, Element 53[a] would have been obvious to a POSITA in view of Malamud and Nakagawa.

228. As discussed in Section VI[A][4], Element 53[a], Malamud teaches specified contention information in the form of an application program that includes information for use in the display on the user's terminal.

229. Malamud is silent as to how the data processing system obtains the application program, but it was well-known to a POSITA by June 1997 that a user could request to download an application program from a server, as disclosed in Nakagawa. (Ex. 1005 at 3:1-9.) For example, Nakagawa discloses a user computer connected over a network to a server so that a user can request and download software from a server, as well as downloading updates to the software. (*See, e.g.*, *id*. at Abstract ("A number of sets of software may be systematically distributed and

maintained via a network connecting many vendors and users of client/server software."), 1:13-16 ("The present invention relates to a software distribution and maintenance system and method with which a software vendor can provide a number of users with software over a network . . . ."), 1:46-54 (describing a user's ability to download software from a vendor at the user's request).)

230. A POSITA would also have been motivated to download application programs, such as the one described in Malamud, from a server, as described in Nakagawa, for the reasons set forth in Section V[B][2] and Section § V[B][3], Element 72[Preamble].

231. Thus, it would have been obvious to a POSITA to combine Malamud's teaching of an application program for modifying a cursor image with the Nakagawa's disclosure of the common and well-known procedure of downloading an application program from a server. That combination renders this claim limitation obvious, as the request to download Malamud's application program from a server would be a request to a server to provide specified content information to a user terminal.

> **<u>Element 53[b]:</u> 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**

232. In my opinion, Element 53[b] would have been obvious to a POSITA in view of Malamud and Nakagawa.

233. As discussed in conjunction with claim element 53[a], it would have been obvious in view of Malamud and Nakagawa for a user to request from a server, and receive in response, an application program. Malamud's application program comprises specified content information because it includes information regarding the display of elements depicted by the application program, such as objects to which the cursor can point and modified cursors (i.e., information cursors), and it includes at least one cursor display instruction and an indication of cursor image data corresponding to a specific image. (*See* § VI[A][4] at Element 53[b].)

234. It would have been obvious to a POSITA to download Malamud's application program (comprising specified content information) from a server as disclosed in Nakagawa in response to a request (*see* § V[B][2]; *see also* § V[B][3], Element 72[Preamble]). Malamud's application program includes a cursor display instruction and an indication of cursor image data relating to a specific image for particular entities, and that information is conveyed to the OS through a message sent by the application program. (*See* EX1004 at 5:47-62 (describing the message that is passed describing the type of cursor to display and information about the contents of the modified cursor image).) For example, the application program in Malamud contains instructions whether to display an information cursor, and if so

the type of information cursor (such as a preview cursor or a combined name and preview cursor) for each entity displayed in the application window. (*Id*. at 3:59-4:18; 5:24-45; 5:46-62.) Malamud also teaches that, for a preview cursor, the application program transmits (and therefore contains) an indication of cursor image data in the form of "a pointer to a bitmap of graphical information that the operating system 22 should use in the preview portion 36." (*Id*. at 5:57-62.) The bitmap is a "specific image" that is displayed as a modified cursor image, and it relates to an object on the display screen to which the cursor is pointing because it provides further information about the object. (*Id*. at 5:57-62, 3:59-4:3.) For example, Malamud teaches that if the cursor is pointing to a book, the bitmap can constitute an image that provides further information about the book. (*Id*. at 3:65-4:3.)

235. For the reasons cited above, this limitation would have been obvious to a POSITA in view of Malamud and Nakagawa.

> **<u>Element 53[c][i]:</u> 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**

236. In my opinion, Element 53[c][i] is disclosed in Malamud.

237. As addressed in Section VI[A][4], Element 53[c][i], Malamud teaches this claim limitation.

<u>Element 53[c][ii]:</u> 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, and

238. In my opinion, Element 53[c][ii] is disclosed in Malamud and/or would have been obvious to a POSITA in view of Malamud and Nakagawa.

239. As addressed in Section VI[A][4], Element 53[c][ii], Malamud teaches this claim limitation and/or renders it obvious.

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

240. In my opinion, Element 53[c][iii] is disclosed in Malamud.

241. As addressed in Section VI[A][4], Element 53[c][iii], Malamud teaches this claim limitation and/or renders it obvious.

242. Thus, claim 53 would have been obvious to a POSITA in view of Malamud and Nakagawa.

C. Ground 3: Claims 1, 27, 38, 53 of the '449 Patent are Rendered Obvious by Nielsen and Malamud

1. Motivation to Combine

243. I discussed the reasons a POSITA would have been motivated to combine Malamud and Nielson in connection with the '102 Patent. (*See* § V[C][2].) That same motivation applies here.

2. Claim 1

<u>Element 1[Preamble]:</u> 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:

244. In my opinion, Element 1[Preamble] would have been obvious to a POSITA in view of Nielsen and Malamud.

245. Nielsen teaches a method for displaying a tooltip on a display of a user terminal connected to at least one server. (EX1006 at Abstract, 1:18-26, 1:51-57.) The tooltip, which can be a text and/or a graphic (a specific image having a desired shape and appearance) relating to the object to which the cursor is pointing (*id.* at 3:37-41, 3:52-56, 3:61-63), is displayed in proximity to the object displayed to which the cursor is pointing (*id.* at 7:8-12, 7:19-22). Malamud also teaches displaying text and/or a graphic that relates to the object to which the cursor is pointing. (EX1004 at Abstract, 1:40-42, 3:59-4:17.) Malamud teaches that this text and/or graphic can be part of a cursor that includes the additional text and/or graphic. (*Id*. at 3:59-4:3.). A POSITA would have been motivated to modify the teachings of Nielsen in view of Malamud such that the tooltip is not displayed above the object to which the cursor is pointing, but rather is displayed as part of a modified cursor image. (*See* § V[C][2].) Because of this, it would have been obvious to a POSITA to modify the teachings of Nielsen to modify an initial cursor image into a modified cursor image

that includes the tooltip having a desired shape and appearance and displayed on a user's terminal.

<u>Element 1[a]:</u> cursor image data corresponding to said specific image;

246. In my opinion, Element 1[Preamble] would have been obvious to a POSITA in view of Nielsen and Malamud.

247. Nielsen teaches that the server transmits HTML code for a web page to the user's computer. (EX1006 at 1:12-14, 1:18-26.) The HTML code specifies, in part, an "icon or other illustration" (image data) to be depicted as the tooltip (*id*. at 3:61-63), and the tooltip would be a specific image related to the information displayed on the screen (*id*. at 3:35-41). While Nielsen's tooltips are displayed as separate images, it would have been obvious to a POSITA to display the tooltips as part of a modified cursor image, and a POSITA would have been motivated to do so in view of Nielsen and Malamud. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).) Thus, it would have been obvious to a POSITA to modify Nielsen's teachings to make the tooltip display instruction a cursor display instruction, which would make the indication of a bitmap for a graphical tooltip image into an indication of cursor image data corresponding to a specific image. This would make it much more likely that the user would notice and see the additional information that is presented to them.

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

248. In my opinion, Element 1[Preamble] would have been obvious to a POSITA in view of Nielsen and Malamud.

249. Nielsen teaches display code for processing tooltip instructions, where the code is operable to process the instruction and display the appropriate tooltip on the users display screen. (EX1006 at Abstract, 1:36-37 (discussing that the browser software displays the tooltip as specified by the web page designer),1:51-62, 3:35-41, 3:46-63, 5:20-58 (describing the processing that is performed by the browser code when a tooltip instruction is detected).) Nielsen further teaches an algorithm implemented by code to display the tooltip appropriately. (*Id*. at 5:18-6:20; *see also* Fig. 6(a) (step 606 indicating that the presence of a tooltip instruction indicates what portion of the algorithm, implemented by code, should be used to process the HTML code).)

250. While Nielsen's tooltip is displayed as a separate image, it would have been obvious to a POSITA that the tooltip could be displayed as part of a modified cursor image as taught in Malamud, and a POSITA would have been motivated to modify the teachings of Nielsen in that manner. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).) Modifying the teachings of Nielsen in such a manner would result in Nielsen's tooltip display code

being utilized as cursor display code, and Nielsen's tooltip being presented as a modified cursor image.

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

251. In my opinion, Element 1[c][i] is disclosed in Nielsen.

252. Nielsen teaches that the server provides the specified content information in the form of HTML code for a web page to the user's computer in response to the user's request for the web page. (EX1006 at 1:12-14, 1:18-26.) The HTML code for the web page is specified content information regarding the web page to be displayed, and it includes instructions regarding the display of tooltips, such as indication whether any tooltips are to be displayed, what is to be displayed, and the information on the user's screen to which the cursor must point to invoke the display of the tooltips. (*Id*. at 1:51-62, 3:35-41, 3:46-4:7.) Likewise, Malamud's application program includes information to be displayed on the user's screen, such as book icon 32, and instructions regarding the display of information cursors. (EX1004 at 3:59-4:18). While Nielsen's tooltips are displayed as separate images, it would have been obvious to a POSITA in view of Malamud to display the tooltips as part of a modified cursor image, and a POSITA would have been motivated to do so. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).)

<u>Element 1[c][ii]:</u> 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,

253. In my opinion, Element 1[c][ii] would have been obvious to a POSITA in view of Nielsen and Malamud.

254. Nielsen teaches that the HTML code for the web page (specified content information) includes a tooltip display instruction that specifies when to display the tooltip, the type of tooltip to display, and the content of the tooltip. (EX1006 at 1:18-26, 1:51-57, 3:36-41, 3:46-4:7.) As discussed above, it would have been obvious to a POSITA to modify Nielsen's tooltip display instructions to make them cursor display instructions, as taught by Malamud, and a POSITA would have been motivated to make such a modification. (*See* § VI[[C][2] at Element 1[Preamble]; *see also* § V[C][2] (regarding motivation to combine the teachings of Nielsen and Malamud).)

255. Nielsen's tooltip instruction indicates the location of image data by specifying the name of the icon or other illustration to display or by specifying a file name and location for the tooltip. (EX1006 at 3:56-63.) It also would have been obvious to a POSITA in view of Nielsen and Malamud that the cursor display instruction could include a pointer to a bitmap, as taught in Malamud. (Ex. 1004 at 5:57-62.)

256. Nielsen's tooltip display instruction and display code together cause the tooltip to be displayed. (EX1006 at 5:18-6:20, 7:8-19.) If the teachings of Nielsen were modified for the reasons discussed above to display the tooltip as a modified cursor image, the modified Nielsen code would be cursor display code and the modified instructions would be the cursor display instructions that cause a modified cursor image to be displayed in the shape and appearance of the specific image identified by the icon, filename, or bitmap. (*Id*. at 3:56-63; EX1004 at 5:57-62.)

> **<u>Element 1[c][iii]:</u> 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,**

257. In my opinion, Element 1[c][iii] is disclosed in Nielsen.

258. As discussed in regard to Element 1[c][i], Nielsen teaches that the server provides the specified content information in the form of HTML code for a web page to the user's computer in response to the user's request for the web page. (EX1006 at 1:12-14, 1:18-26.) That HTML code would include information to be displayed on the user's screen in the form of the web page displayed by the browser. (*Id*. at Abstract, 6:36-41, Fig. 7 (element 702 ("Display web page in accordance with HTML")).)

> **<u>Element 1[c][iv]:</u> said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said**

cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image in response to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal, and wherein said specific image relates to at least a portion of said information to be displayed on said display of said remote user's terminal.

259. In my opinion, Element 1[c][iv] would have been obvious to a POSITA in view of Nielsen and Malamud.

260. Nielsen teaches that the tooltip image includes content corresponding to at least a portion of the information on the web page displayed on the user's terminal. (EX1006 at 3:35-41 (". . . tooltip text is displayed, providing further explanation of the information at the predetermined location."), 3:61-63 (explaining that the tooltip display can be an image in the form of an icon or other illustration).)

261. As discussed above in regard to Element 1[c][ii], it would have been obvious to a POSITA to display Nielsen's tooltip as part of a modified cursor image, thus making Nielsen's tooltip display instructions and display code into cursor display instructions and cursor display code. Such cursor display code would be used to process the cursor display instruction, as Nielsen teaches display code that is used to process the tooltip instruction. (*See, e.g.*, EX1006 at 5:30-6:35.) The modified image is then displayed in the shape and appearance of the specific image when the cursor is moved over at least a portion of the information displayed on the user's terminal. (*Id*. at Abstract, 1:51-57, 3:35-41.) In addition, as discussed above

at the outset of this element, Nielsen teaches that the tooltip image includes content corresponding to at least a portion of the information on the web page displayed on the user's terminal, so the tooltip image "relates to at least a portion of said information to be displayed on said display of said remote user's terminal."

262. Thus, claim 1 would have been obvious to a POSITA in view of Nielsen and Malamud.

3. Claim 27

263. Claim 27 is word-for-word identical to claim 1 except that claim 1 recites that the modified cursor image is displayed in response to movement of the 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" (EX1002 at 19:1-3), whereas claim 27 recites that the modified cursor image is displayed in response to movement of the cursor image over "a specified location on said display of said user's terminal" (*id*. at 20:63-65). Nielsen teaches both claims by teaching that display of the tooltip depends upon the location of the cursor and what is depicted at that location. (EX1006 at 1:51-56 (discussing the display of tooltips based upon the displayed information that the cursor is placed over), 3:35-41 (discussing displaying the tooltip when the cursor is at a "predetermined location on [the] display device"), 6:63-7:19 (explaining how the cursor location and information displayed at that location is used to determine whether to display a tooltip).)

264. As discussed above, it would have been obvious to a POSITA to combine the teachings of Nielsen and Malamud and display the tooltip as a modified cursor image, thus rendering claim 27 obvious.

4. Claim 38

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

265. Nielsen teaches that the specified content information (as discussed above in Element 1[c][i], which is identical to Element 27[c][i]) is transmitted from a server to a user in the form of HTML code that defines a web page. (EX1006 at Abstract, 1:12-28, 1:51-62, 1:66-2:9, 3:46-4:7, Fig. 6(a) at element 604.)

266. Thus, claim 38 would have been obvious to a POSITA in view of Nielsen and Malamud.

5. Claim 53

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

267. In my opinion, Element 53[Preamble] would have been obvious to a POSITA in view of Nielsen and Malamud.

268. Nielsen teaches a method for displaying a tooltip on a display of a user terminal connected to at least one server. (EX1006 at Abstract, 1:18-26, 1:51-57.) The tooltip, which can be a text and/or a graphic relating to the object to which the cursor is pointing (*id.* at 3:37-41, 3:52-56, 3:61-63), is displayed in proximity to the

displayed object to which the cursor is pointing (*id.* at 7:8-12, 7:19-22). Malamud also teaches displaying text and/or a graphic relating to the object to which the cursor is pointing. (EX1004 at Abstract, 1:40-42, 3:59-4:17.) Malamud teaches that this text and/or graphic can be part of a modified cursor image. (*Id*. at 3:59-4:3.). A POSITA would have been motivated to modify the teachings of Nielsen in view of Malamud such that the tooltip is not displayed above the object to which the cursor is pointing, but rather is displayed as part of a modified cursor image to ensure that the user sees the supplemental information or graphic. (*See* § V[C][2].) Therefore, it would have been obvious to a POSITA to modify the teachings of Nielsen to modify an initial cursor image into a modified cursor image that includes the tooltip.

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

269. In my opinion, Element 53[a] is disclosed in Nielsen.

270. Nielsen teaches receiving a request at the server to provide specified content information. Specifically, Nielsen teaches that a user requests a web page from a server, and the server sends the HTML code for the requested web page to the user's computer. (EX1006 at 1:9-26, 1:51-57, Fig. 1, Fig. 6(a).) The HTML code for the web page includes information provided to a user's terminal for use in the display, including the web page content to be displayed, and tooltip display instructions for displaying tooltips. (*Id*. at 1:18-26, 1:51-57, 3:36-41, 3:46-4:7.) As discussed above, it would have been obvious to a POSITA to modify Nielsen's

tooltip display instructions to make them cursor display instructions, as taught by Malamud, and a POSITA would have been motivated to make such a modification. (*See* § VI[C][5] at Element 53[Preamble]; *see also* § V[C][2] (regarding motivation to combine the teachings of Nielsen and Malamud).)

> **<u>Element 53[b]:</u> 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**

271. In my opinion, Element 53[b] would have been obvious to a POSITA in view of Nielsen and Malamud.

272. Nielsen teaches that the server provides the specified content information in the form of HTML code for a web page to the user's computer in response to the user's request for the web page. (EX1006 at 1:12-14, 1:18-26.) The HTML code for the web page is the specified content regarding the web page information to be displayed, and it includes instructions regarding the display of tooltips, such as indication whether any tooltips are to be displayed, what is to be displayed, and the information on the user's screen to which the cursor must point to invoke the display of the tooltips. (*Id*. at 1:51-62, 3:35-41, 3:46-4:7.) While Nielsen's tooltips are displayed as separate images, it would have been obvious to a POSITA to display the tooltips as part of a modified cursor image, and a POSITA would have been motivated to do so. (*See* § V[C][2] (discussing the motivation to

combine the teachings of Nielsen and Malamud).) Thus, it would have been obvious to modify Nielsen's teachings to make the tooltip display instruction a *cursor* display instruction, which would make the indication of a bitmap for a graphical tooltip image into an indication of *cursor* image data that corresponds to a specific image.

> **<u>Element 53[c][i]:</u> 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**

273. In my opinion, Element 53[c][i] would have been obvious to a POSITA in view of Nielsen and Malamud.

274. Nielsen teaches transforming the display on the user's terminal to display a tooltip when the cursor points to an object on the display screen that has an associated tooltip. (EX1006 at Abstract, 1:51-62, 3:35-41.) While Nielsen's tooltip is displayed as a separate image, it would have been obvious to a POSITA that the tooltip could be displayed as part of a modified cursor image as taught in Malamud, and a POSITA would have been motivated to modify the teachings of Nielsen in that manner. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).) The modified cursor image is in the shape and appearance of the specific image specified by the bitmap and would be displayed in response to the tooltip instruction when the cursor is placed over an object on the

screen that has an associated tooltip (EX1006 at 3:35-41, 3:46-4:7), and the tooltip instruction would be a cursor display instruction when the tooltip is displayed as part of a modified cursor image by combining the teachings of Nielsen and Malamud.

275. The specified content information received from the server in Nielsen (i.e., the HTML code for the web page, including the tooltip instructions), includes information displayed on the user's computer, in the form of the web page and the tooltips. (EX1006 at 1:12-26, 1:51-57, 1:66-2:9.) Nielsen discloses that the displayed tooltip relates to and provides further information about an object on the computer display screen pointed at by the cursor. (*Id*. at 1:54-62, 3:35-41.) Similarly, Malamud teaches that the information cursor, such as a preview cursor or a combined name and preview cursor, also relates to the object on the computer display screen to which the cursor is pointing. (EX1004 at Abstract, 1:35-43, 2:9-12, 2:66-3:6.)

276. Therefore, this limitation would have been obvious to a POSITA in view of the teachings of Nielsen and Malamud, and POSITA would have been motivated to combine the teachings of those references. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).)

> **<u>Element 53[c][ii]:</u> 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, and

277. In my opinion, Element 53[c][ii] would have been obvious to a POSITA in view of Nielsen and Malamud.

278. Nielsen teaches HTML code for web pages, where the code contains tooltip instructions. (EX1006 at Abstract, 1:51-62.) The tooltip instruction includes information regarding whether a tooltip is to be displayed and the type of tooltip to be displayed, such as text or a graphical image. (*Id*. at 3:35-41, 3:46-63.) The tooltip therefore indicates the web browser code operable to process the instruction to display the appropriate type of tooltip. (*See, e.g.*, *id*. at 1:36-37 (discussing that the browser software displays the tooltip as specified by the web page designer); *id*. at 3:54-56 (identifying the HTML text that indicates the presence of a tooltip that must be processed using the appropriate browser code); *id*. at 3:61-63 (identifying that a tooltip can also indicate that the tooltip should be displayed as an icon or other illustration); 5:20-58 (describing the processing that is performed using the browser code when a tooltip instruction is detected).) Nielsen further teaches an algorithm implemented by code to display the tooltip appropriately. (*Id*. at 5:18-6:20; *see also* Fig. 6(a) and Fig. 6(b) (see step 606 indicating that the presence of a tooltip instruction indicates what portion of the algorithm, implemented by code, should be used to process the HTML code).)

279. The processing of Nielsen's tooltip instruction in the HTML code causes a tooltip to be displayed in response to the user placing the cursor over an object on the displayed web page, where the object has an associated tooltip. (*Id*. at 3:35-41.) The tooltip takes the shape of the specific image in the form of an icon or other illustration as identified by the tooltip instruction and related to an image displayed on the screen and to which the cursor is pointing. (*Id*. at 3:35-41, 3:61-63.)

280. While Nielsen's tooltip is displayed as a separate image, it would have been obvious to a POSITA that the tooltip could be displayed as part of a modified cursor image as taught in Malamud, and a POSITA would have been motivated to modify the teachings of Nielsen in that manner. (*See* § V[C][2] (discussing the motivation to combine the teachings of Nielsen and Malamud).) Modifying the teachings of Nielsen in such a manner would result in Nielsen's tooltip instruction being a cursor display instruction, Nielsen's tooltip display code being cursor display code, and Nielsen's tooltip being a modified cursor image in the shape and appearance of the specific image.

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

281. In my opinion, Element 53[c][iii] is disclosed in Nielsen.

282. Nielsen teaches that the graphical tooltip image, which is a specific image as discussed above, has a shape and appearance relating to information to be

displayed on the screen. (EX1006 at 1:51-55, 3:55-41, 3:61-64.) Likewise, Malamud teaches that the preview cursor (and the combined name and preview cursor), which is a specific image as discussed above, has a shape and appearance relating to information to be displayed on the screen. (EX1004 at 1:39-43, 2:10-12, 3:59-4:18.)

283. It also would have been obvious to a POSITA in view of Nielsen and Malamud that the specific image of Nielsen's tooltip could be in a shape and appearance relating to information displayed on the screen.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001; and further that such willful false statements may jeopardize the validity of the application or any patent issued thereon. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on June 5th, 2023 in Shoreline, Washington.

Dr. Craig Rosenberg