UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,

Petitioner,

v.

NEONODE SMARTPHONE LLC,

Patent Owner.

Case No. IPR2021-01041
U.S. Patent No. 8,095,879

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

LIST OF EXHIBITS

Exhibit	Description
Ex-1001	U.S. Patent No. 8,095,879 ("the '879 patent")
Ex-1002	File History of U.S. Patent No. 8,095,879
Ex-1003	Declaration of Jacob O. Wobbrock, Ph.D.
Ex-1004	CV of Jacob O. Wobbrock, Ph.D.
Ex-1005	George G. Robertson, Buttons as First Class Objects on an X Desktop, *UIST: Proceedings of the ACM Symposium on User Interface Software and Technology: Hilton Head, South Carolina, USA*, pp. 35-44 (Nov. 11-13, 1991) ("Robertson")
Ex-1006	U.S. Patent No. 7,768,501 to Maddalozzo Jr. et al. ("Maddalozzo")
Ex-1007	U.S. Patent No. 5,745,717 to Vayda et al. ("Vayda")
Ex-1008	U.S. Patent No. 5,870,092 to Bedford-Roberts ("Bedford-Roberts")
Ex-1009	U.S. Patent No. 6,181,344 to Tarpenning et al. ("Tarpenning")
Ex-1010	U.S. Patent Application Publication No. 2001/0035880 to Musatov et al. ("Musatov")
Ex-1011	U.S. Patent Application Publication No. 2002/0149569 to Dutta et al. ("Dutta")
Ex-1012	U.S. Patent Application Publication No. 2003/0013483 to Ausems et al. ("Ausems")
Ex-1013	U.S. Patent No. 6,249,277 to Varveris ("Varveris")
Ex-1014	U.S. Patent No. 6,344,848 to Rowe et al. ("Rowe")
Ex-1015	U.S. Patent No. 6,388,870 to Canova Jr. et al. ("Canova")
Ex-1016	U.S. Patent No. 7,081,882 to Sowden et al. ("Sowden")

Exhibit	Description
Ex-1017	U.S. Patent No. 5,347,295 to Agulnick et al. ("Agulnick")
Ex-1018	Declaration of Rachel J. Watters
Ex-1019	Declaration of Kelley M. Hayes Greenhill
Ex-1020	Wisconsin MARC Record for Robertson, *available at* https://wrlc-gu.primo.exlibrisgroup.com/discovery/sourceRecord?vid=01WRLC_GUNIV:01WRLC_GUNIV&docId=alma9911002247004101&recordOwner=01WRLC_NETWORK (last visited May 26, 2021)
Ex-1021	Library of Congress MARC Record for Robertson, *available at* https://catalog.loc.gov/vwebv/staffView?searchId=16367&recPointer=0&recCount=25&bibId=11489112 (last visited May 27, 2021)
Ex-1022	U.S. Copyright Office Record, *available at* https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?v1=8&ti=1,8&Search%5FArg=Proceedings%20of%20the%20ACM%20Symposium%20on%20User%20Interface%20Software%20and%20Technology&Search%5FCode=TALL&CNT=25&PID=S5YOB6rlECB6M_LEJD300IQpy&SEQ=20210527133326&SID=3 (last visited May 27, 2021)
Ex-1023	WorldCat Record for OCLC Control Number 28864712, *available at* https://www.worldcat.org/title/proceedings-of-the-acm-symposium-on-user-interface-software-and-technology/oclc/28864712 (last visited June 2, 2021)
Ex-1024	WorldCat Record for OCLC Control Number 270712133, *available at* https://www.worldcat.org/title/proceedings-of-the-acm-symposium-on-user-interface-software-and-technology-hilton-head-south-carolina-usa-november-11-13-1991/oclc/270712133 (last visited June 2, 2021)
Ex-1025	*Neonode Smartphone LLC v. Apple Inc.*, No. 6:20-cv-00505-ADA, Docket No. 40, Order Granting Stay (W.D. Tex. Dec. 8, 2020)

Exhibit	Description
Ex-1026	*Neonode Smartphone LLC v. Samsung Electronics Co. Ltd.*, No. 6:20-cv-00507-ADA, Text Order Granting Stay (W.D. Tex. Dec. 11, 2020)
Ex-1027	Excerpts from Robert W. Scheifler & James Gettys, X Window System, Version 11, Release 5 (Digital Press 3d ed. 1992)
Ex-1028	U.S. Patent No. 5,745,116 to Pisutha-Arnond ("Pisutha-Arnond")
Ex-1029	U.S. Patent No. 5,644,628 to Schwarzer et al. ("Schwarzer")

I. Relief Requested

Petitioner Google LLC requests review under 35 U.S.C. § 311 of claims 1-7, 9, 12-13, and 15-17 of U.S. Patent 8,095,879 and cancellation under pre-AIA 35 U.S.C. § 103 based on:

Ex-1005	Robertson, Buttons as First Class Objects on an X Desktop, (November 11-13, 1991); prior art under § 102(b)
Ex-1006	U.S. Patent 7,768,501 ("Maddalozzo"); filed May 1, 1998; issued August 3, 2010; prior art under § 102(e)
Ex-1007	U.S. Patent 5,745,717 ("Vayda"); issued April 28, 1998; prior art under § 102(b)
Ex-1008	U.S. Patent 5,870,092 ("Bedford-Roberts"); issued February 9, 1999; prior art under § 102(b)
Ex-1009	U.S. Patent 6,181,344 ("Tarpenning"); issued January 30, 2001; prior art under § 102(b)

Ground	Claims	§ 103 Basis
1	1-5, 13, 15-17	Rendered obvious by Robertson and Maddalozzo

Ground	Claims	§ 103 Basis
2	6-7, 9	Rendered obvious by Robertson, Maddalozzo, and Vayda
3	12	Rendered obvious by Robertson, Maddalozzo, and Bedford-Roberts
4	1, 4-6, 13, 15-17	Rendered obvious by Tarpenning
5	2-3, 7, 9	Rendered obvious by Tarpenning and Vayda
6	12	Rendered obvious by Tarpenning and Bedford-Roberts

II. The '879 Patent

The '879 patent describes a touch-based user interface for a mobile handheld computer unit (e.g., "mobile phone," "PDA," "laptop computer"). Ex-1001, 1:6-9, 1:24-33. The user interface includes a "touch sensitive area" 1 (**red**) divided into menu area 2 and display area 3. Menu area 2 includes "representation[s]" (**blue**) of predefined functions 21, 22, 23 activated when "touch sensitive area 1 detects a movement [(**orange**)] of an object 4 [finger, pen] with its starting point A within the representation [(**blue**)] of a function on the menu area 2 and with a direction B

from the menu area 2 to the display area 3 [(yellow)],” as shown. Ex-1001, 3:50-54, 4:1-11, 2:10-14, 6:11-15.

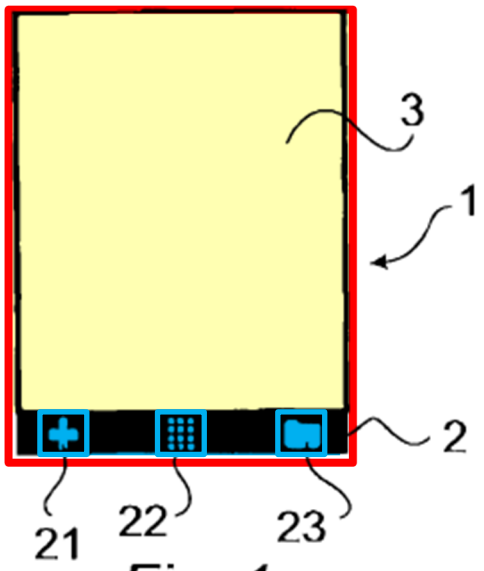


Fig. 1.

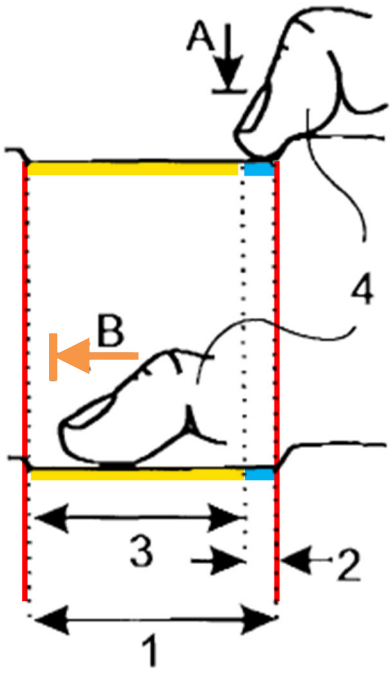


Fig. 2.

Ex-1001, FIGS. 1, 2 (annotated).

When the functions for representations 21 and 23 are activated by a flick gesture, display area 3 displays icons representing services or functions (**green**, Figure 3) or a menu/library of applications and files (**bright green**, Figure 6), as shown. Ex-1001, 4:7-23.

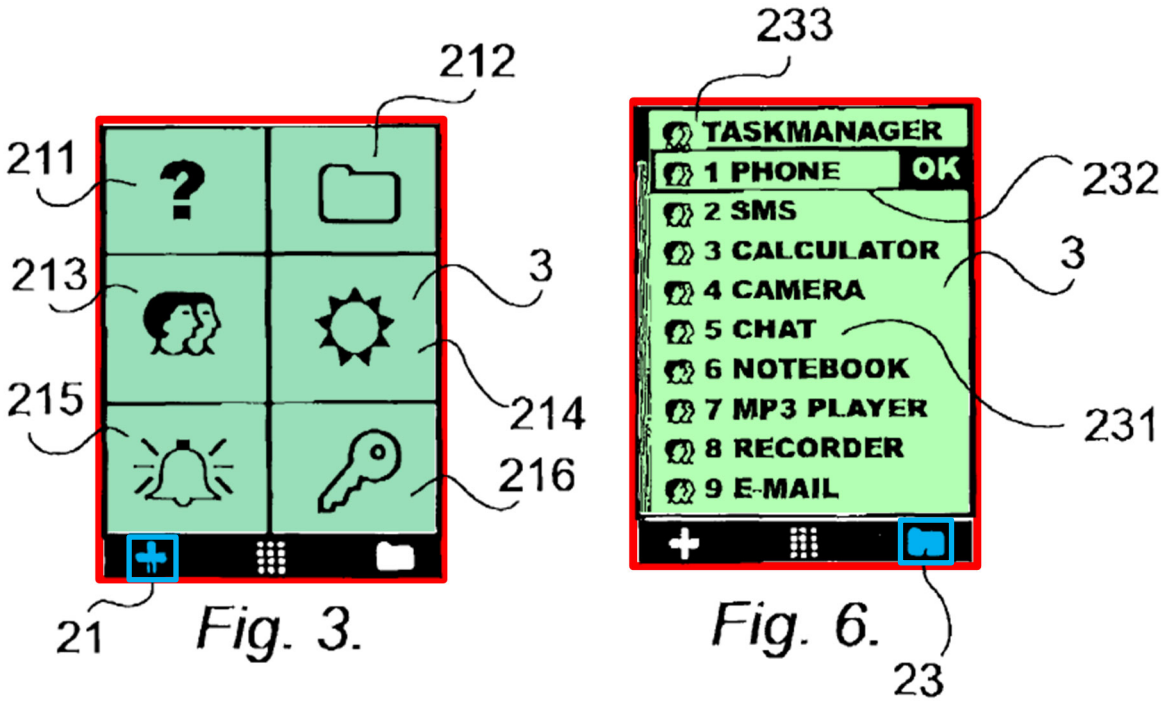


Fig. 3. Fig. 6.

Ex-1001, FIGS. 3, 6 (annotated).

III. Claim Construction

No terms need to be construed to resolve unpatentability. *Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019).

IV. Level of Ordinary Skill

A person of ordinary skill in the art ("POSITA") of the '879 patent would have at least a bachelor's degree in Computer Science, Human-Computer Interaction, Symbolic Systems, or related engineering disciplines, and at least two years of experience designing and programming graphical user interfaces. Relevant work experience can substitute for formal education and advanced degree studies could substitute for work experience. Ex-1003, ¶49.

V. The Challenged Claims Are Unpatentable and Should Be Cancelled

A. [Ground 1] Claims 1-5, 13, and 15-17 are rendered obvious by Robertson and Maddalozzo

1. Robertson

Robertson discloses a gesture-based "high-level user interface toolkit" having configurable buttons ("XButtons") that can be used with "touch screens for [user] input." Ex-1005, Abstract, §§ 1, 1.2, 2. Robertson activates a button's corresponding functions using gestures, e.g., "pen-based gestural input[s]," such as the user touching the screen and then sliding or gliding the pen across the screen. Ex-1005, §§ 3-3.1. When "a user gestures at an XButton, a gesture parser interprets … pen movement and classifies it as one of a small set of easily differentiated gestures (flick left, flick right, flick up, flick down, click, rubout, check, or insert)," and then "executes the appropriate action (i.e., there is an action for each of the gestures)" corresponding to the gesture. Ex-1005, § 3.1. The screen also shows a "track" of the gesture being drawn. *Id.*

Annotated Figure 1 shows a touch screen (**red**) having a "Phone" button (**blue**) with a corresponding function to dial a phone number using a multi-step "flick right" gesture, activated by the user touching the pen on the "Phone" button and drawing a line to the right on the screen (**orange**). Ex-1005, §§ 3-3.1, 4.2.

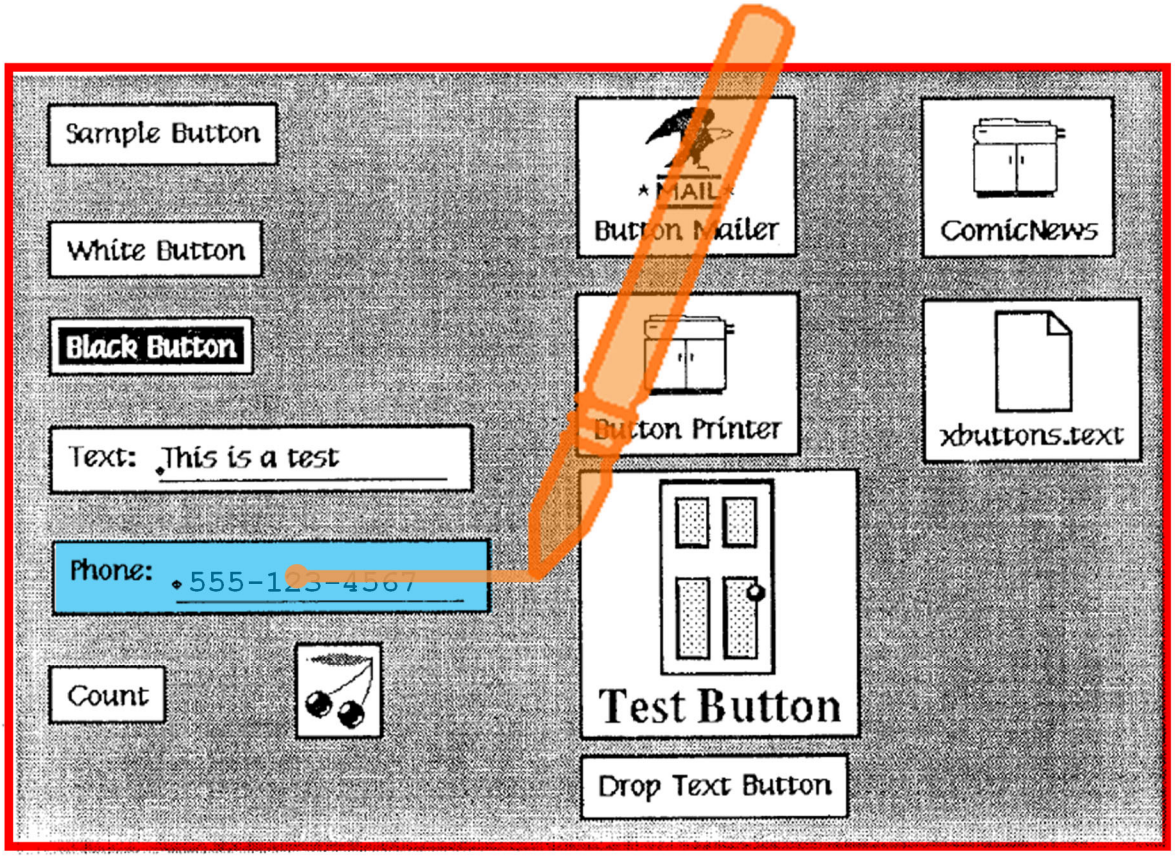


Figure 1: Sample Set of XButtons.

Ex-1005, FIG. 1 (annotated).

The user can also activate the "button editor" menu function of Figure 3 using an "Insert" gesture—drawing a "caret" starting on the button to be edited—as shown. Ex-1005, §§ 3.2-3.3.

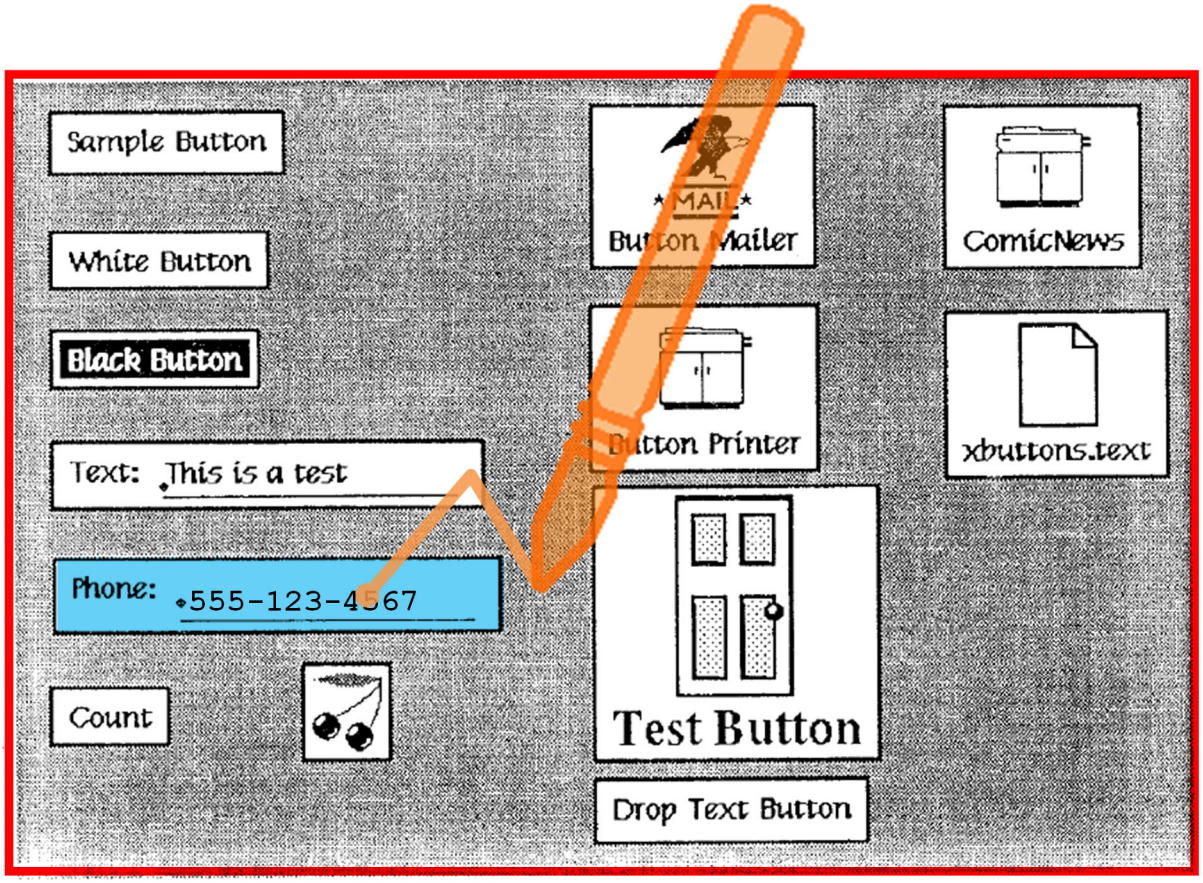


Figure 1: Sample Set of XButtons.

Ex-1005, FIG. 1 (annotated).

Robertson uses toolkits to create buttons and assign functions to gestures using "the operating system command language." Ex-1005, Abstract, §§ 3.2-3.3. Robertson discloses command languages, like Unix and Lisp, but does not limit itself to these languages. Ex-1005, §§ 1.2, 3.2; Ex-1003, ¶67.

Annotated Figure 3 shows a button editor to configure button attributes (e.g., title, bitmap, foreground, background, text) and assign functions (**green**) activated by gestures (**orange**). Ex-1005, §§ 3.2-3.3.

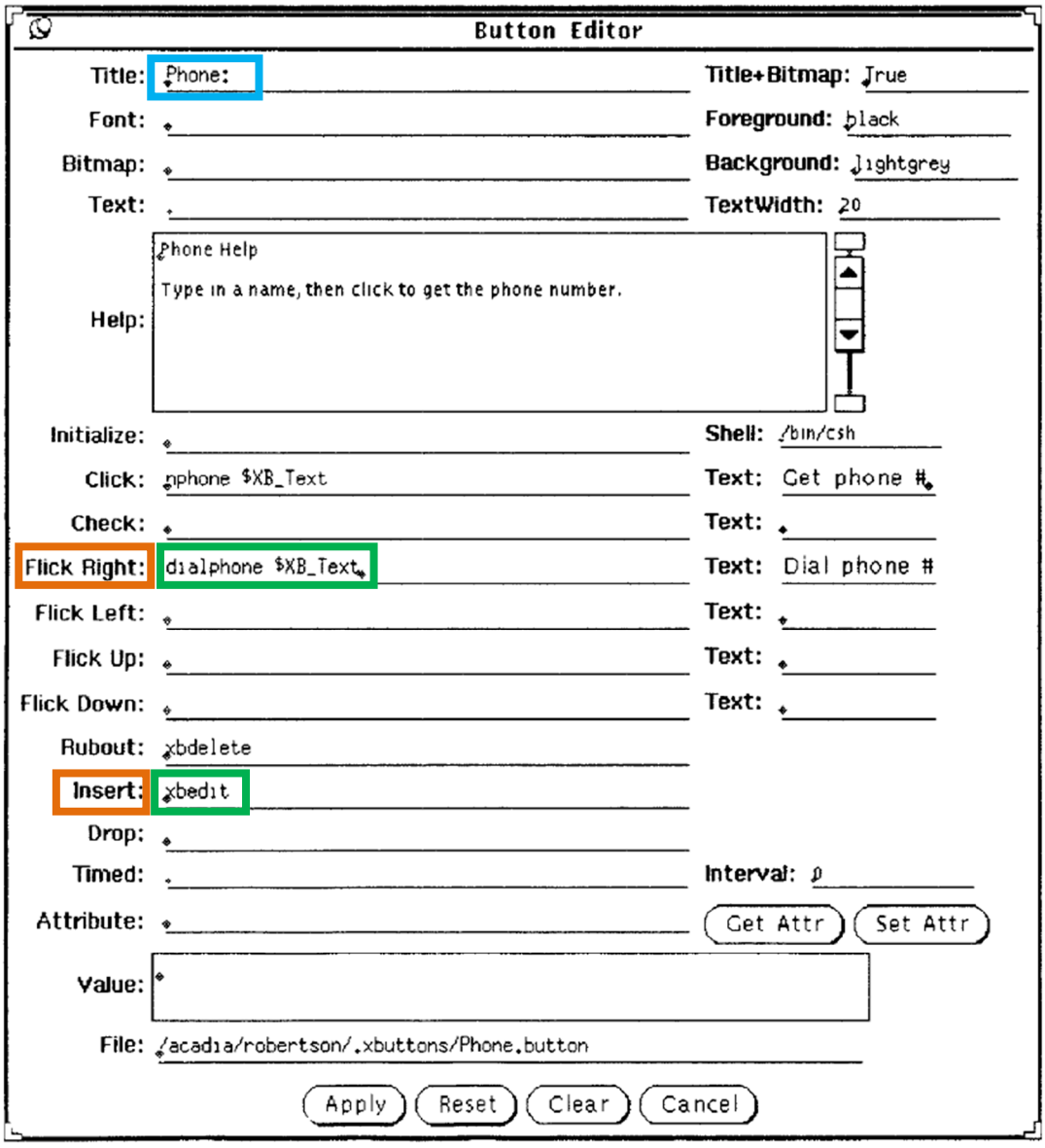


Figure 3: Structured Button Editor.

Ex-1005, FIG. 3 (annotated).

Robertson was publicly available to POSITAs more than one year before the '879 patent filing as part of published materials from the ACM's 4th Annual Symposium on User Interface Software and Technology (UIST), held November 11-13, 1991. Ex-1005; Ex-1003, ¶70. These papers were distributed to conference attendees, including POSITAs, at the conference. Ex-1003, ¶71. This is confirmed by the U.S. Copyright Office's publication date of October 28, 1991. Ex-1019, ¶21; Ex-1022. Robertson bears the ACM's 1991 copyright date and numerical publication code and was available for purchase for $1.50. Ex-1005, 11 fn.; Ex-1003, ¶¶73-74. ACM publications were widely distributed and reliably published on or about the ACM copyright date, qualifying Robertson as publicly available on or about 1991. *Microsoft Corp. v. IPA Techs. Inc.*, IPR2019-00811, Paper 44, at 142 (Oct. 15, 2020); *Ericsson Inc. v. Intellectual Ventures I LLC*, IPR2014-00527, Paper 41, at 10-12 (May 18, 2015); Ex-1003, ¶75.

Robertson was catalogued and shelved at the Library of Congress and would have been publicly available to POSITAs through searching as of about September or October 1993. Ex-1019, ¶¶16-18. Robertson was also catalogued and shelved at the University of Wisconsin library and made publicly available to POSITAs through searching about August 20, 1998. Ex-1018, 1-2.

2. Maddalozzo

Maddalozzo discloses mobile handheld computers, e.g., "laptop," "notebook," and "calculator-size mini-portable computers," having touch-screen user interfaces. Ex-1006, 1:12-35, 2:1-22. Like Robertson, these devices may run Unix or similar operating systems. Ex-1006, 4:36-40. These handheld computers have touch-sensitive on-screen keyboards (**purple**) allowing users to input text into a text field (**orange**). Ex-1006, Abstract, FIG. 6.

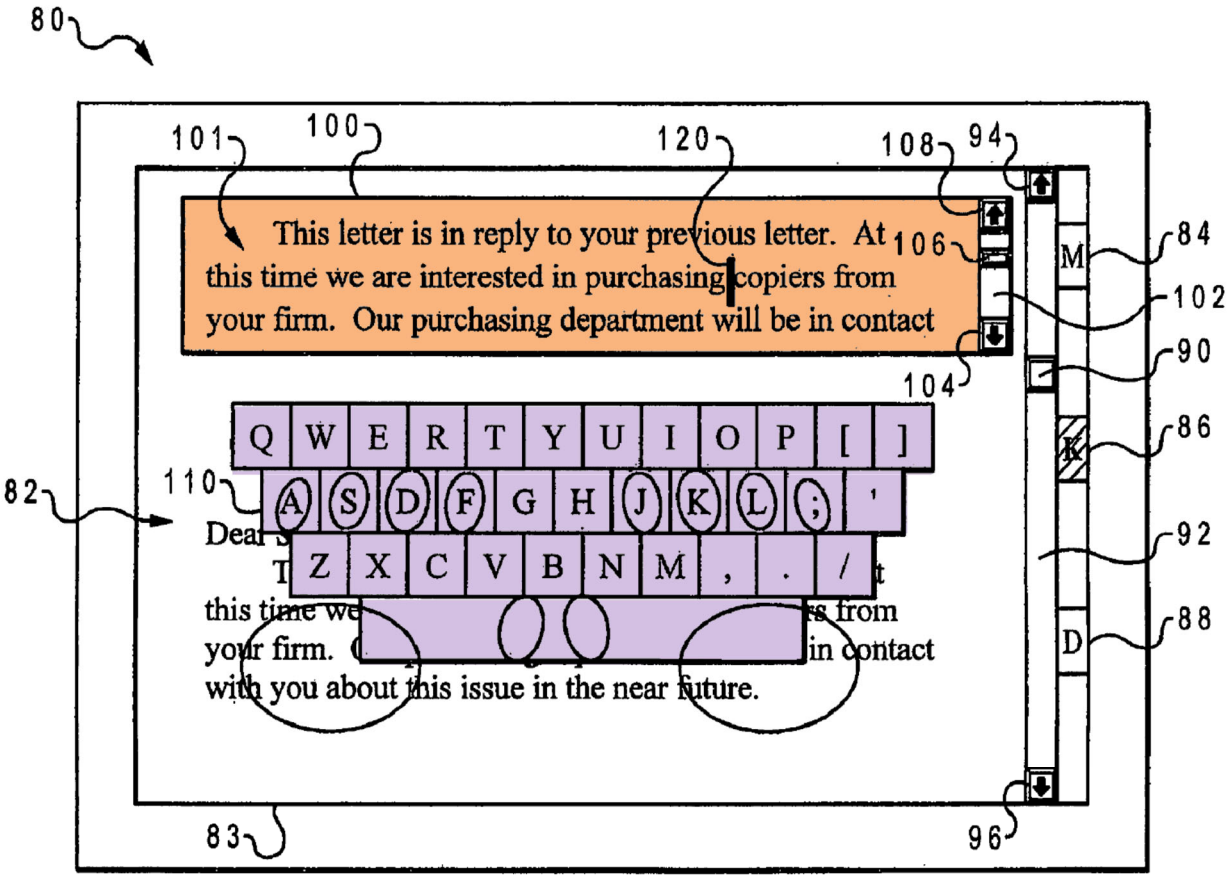


Fig. 6

Ex-1006, FIG. 6 (annotated).

3. Claim 1

Robertson and Maddalozzo render obvious claim 1.

Robertson teaches or suggests each element of claim 1. Maddalozzo discloses, confirms, and renders obvious Robertson's computing device being a mobile handheld computer unit having a "non-transitory computer readable medium" in [1Preamble]. Ex-1003, ¶79.

a. [1Preamble] "A non-transitory computer readable medium storing a computer program with computer program code, which, when read by a mobile handheld computer unit, allows the computer to present a user interface for the mobile handheld computer unit, the user interface comprising:"[1]

Robertson discloses or suggests [1Preamble]. Ex-1003, ¶¶80-89. Maddalozzo discloses and renders obvious the computer unit being a "mobile handheld computer unit" and the computer program being stored in a "non-transitory computer readable medium." Ex-1003, ¶80.

Robertson discloses a computer program with code for implementing gesture-activated buttons (e.g., XButtons), which, when read by a computing device, presents a user interface for receiving gestural inputs (e.g., touch-screen inputs) of the device. Ex-1005, Abstract, §§ 3.1-3.2, 4.1; Ex-1003, ¶81.

Robertson discloses a "user interface toolkit, called *XButtons*" (computer program code) that presents user-activatable "on-screen buttons" to activate functions in a computing system, like an X Windows system. Ex-1005, Abstract. Robertson's XButtons are implemented by computer programs having program

[1] Petitioner does not concede the preamble is limiting. If Patent Owner argues it is, Robertson and Maddalozzo render obvious [1Preamble].

code, like the "operating system command language." Ex-1005, §§ 1.2, 3.2, 4.1; Ex-1003, ¶82.

Robertson suggests that the computing device can be a mobile handheld computer unit. Robertson describes "X"-based commands, Unix-based commands, and Lisp-based commands. Ex-1005, Abstract, §§ 1.2, 2.1, 3-3.2. A POSITA would understand that Robertson's teachings can apply to many types of gesture-based devices, including handheld computing devices, like laptop computers and PDAs. Ex-1003, ¶83. Although one goal of Robertson is to make buttons user-editable, a POSITA would understand that Robertson's gesture activations applied to any device capable of receiving and processing gesture-based inputs, e.g., laptop computers, PDAs, and other small form-factor devices. *Id.*

Robertson teaches activating a button's functions through "pen-based gestural input[s]" in addition to "pressing" (e.g., tapping on a button) to execute the button's function(s). Ex-1005, § 3.1. A POSITA would have known that pen-based input devices were a method of inputting commands to mobile handheld computer units, including touch-screen laptops and PDAs (e.g., Palm OS devices/Palm Pilot, IBM Simon, Apple Newton and MessagePad, Windows Pocket PC devices, and Nokia Symbian-based devices). Ex-1003, ¶84.

A POSITA would understand that Robertson's computer unit stores computer program code on a non-transitory computer readable medium for

implementing the user interface because without computer program code or a storage medium, the computer would not function. Ex-1003, ¶85. Maddalozzo confirms and renders obvious the non-transitory computer readable medium. *Id.* Maddalozzo discloses a computer system including "a graphical user interface that resides within a machine-readable medi[um] to direct the operation of [a] computer system." Ex-1006, 3:32-34. Maddalozzo's mobile computer includes various types of non-transitory machine-readable media, e.g., "RAM 34, ROM 36, … or optical disk," for storing code. Ex-1006, 4:28-34. Dutta also confirms POSITAs knew to implement graphical user interfaces using "software residing in computer readable media" within a computer unit, e.g., PDA. Ex-1011, [0021].

A POSITA would have found it obvious to store Robertson's user interface computer program code in a non-transitory computer readable medium of the laptop, PDA, or other handheld computing device to provide a unitary system. Ex-1003, ¶86.

Robertson discloses a computer unit having a user interface for "pen-based gestural input[s]," but does not specify the type of computer unit. *See generally* Ex-1005; Ex-1003, ¶87. Robertson's user interface using "pen-based gestural input[s]" suggests that the device could be a mobile handheld computing unit (Ex-1005, § 3.1), and Maddalozzo renders obvious implementing Robertson's teachings in a mobile handheld computer unit, e.g., a laptop or PDA. Ex-1003, ¶87.

Maddalozzo discloses different mobile handheld computer devices and explains that touch-screen technology was commonly used in mobile handheld computer units, like laptops, notebook computers, and calculator-size mini-portable computers. Ex-1006, 1:12-35, 2:1-22; Ex-1003, ¶88.

A POSITA would have found it obvious to implement Robertson's XButtons (or other gesture-activated buttons) in mobile handheld computer units to enhance the user experience through more capable buttons supporting gesture activations. Ex-1003, ¶89. POSITAs knew handheld computers could run X protocol and/or Unix using touch screens. Ex-1010 (Musatov), [0034], [0038], [0155]; Ex-1006, 4:32-40; Ex-1003, ¶89. A POSITA would also have been motivated to use Robertson's gesture-based user interface on mobile computing devices with small touch screens and without external keyboards to enhance the functionality of the devices and improve user experience. Ex-1003, ¶89.

i. Motivation

Robertson and Maddalozzo are directed to touch-based user interfaces. Ex-1005, Abstract, §§ 3-3.1; Ex-1006, 1:13-16, 6:13-19; Ex-1003, ¶90. A POSITA would understand that Robertson's teachings apply to many types of gesture-based devices. Ex-1003, ¶90. Maddalozzo's mobile handheld devices, e.g., laptop or PDA, work with "[a]ny suitable operating system and associated graphical user interface," including Robertson's "Unix" operating system or Lisp-based

commands.[2] Ex-1006, 4:32-38; Ex-1005, Abstract, §§ 1.2, 2.1, 3-3.2; Ex-1003, ¶90.

A POSITA would have been motivated to implement Robertson's gesture-based operations in mobile handheld computer units (e.g., laptop, PDA-type device) having non-transitory computer readable media to provide a fully integrated handheld device that stores and executes user interface program code to eliminate any need for external computing components and to provide a simple, configurable multi-function user interface. Ex-1003, ¶92.

"Touch screen technology [was] increasingly being implemented … with portable computers," and the "increasingly portable" nature of such devices would have motivated a POSITA to implement Robertson's gesture-based function activation to keep the form factor small, while still allowing for convenient user interfaces. Ex-1006, 1:28-33; Ex-1003, ¶¶91-93.

[2] Robertson's teachings apply to other command languages. Ex-1005, Abstract, §§ 1.2, 2.1, 3-3.2; Ex-1003, ¶90.

A POSITA would have been motivated to implement Robertson's gesture-based operations on mobile computing devices, e.g., mobile laptop computers[3] or PDA-style devices, to implement gesture-based action buttons using "pen-based gestural input[s]" common for such handheld devices and more convenient to operate on smaller screens. Ex-1003, ¶93; Ex-1005, § 3.1. A POSITA would have been motivated to implement gesture-based interfaces on mobile computing devices because consumers desired portability and gestures were more convenient for such devices, often lacking external mice and keyboards. Ex-1006, 1:28-60; Ex-1003, ¶91.

Robertson's teachings would have been implemented on X-based handheld devices (e.g., laptop computers, PDAs) to take advantage of operations the "Unix Shell" command language provides to design gesture-based buttons that are more useful and convenient for users. Ex-1005, Abstract, §§ 1.2, 2.1, 3.2, 5; Ex-1003, ¶93. A POSITA would also have been motivated to implement Robertson's teachings on other mobile devices (non-X-based, non-Unix) that use well-known

[3] The '879 patent explains that "[m]obile handheld computers" have "various embodiments," including the "personal digital assistant (PDA)" and "laptop computer, which is getting smaller and … competing in size with the PDA's." Ex-1001, 1:24-33.

gesture-based inputs to provide convenient execution of commands and functions using the limited screen space available. Ex-1003, ¶93; *see* Ex-1009 (Tarpenning); Ex-1028 (Pisutha-Arnond); Ex-1029 (Schwarzer); Ex-1012 (Ausems); Ex-1008 (Bedford-Roberts); Ex-1007 (Vayda (stylus)).

ii. Expected success

Implementing pen-based gesture inputs on touch-screen mobile devices was well-known to POSITAs. Ex-1003, ¶95; s*ee* Ex-1009 (finger/stylus inputs to perform touch-screen gestures); Ex-1010 (mobile handheld computer with touch screen implementing X-protocol); Ex-1011 (PDA with touch-screen user interface); Ex-1012; Ex-1013; Ex-1014; Ex-1015. The ’879 patent acknowledges laptop computers, PDAs, and mobile phones having touch-screen gestural inputs were previously known. Ex-1001, 1:24-33.

A POSITA would have known how to implement Robertson’s touch-based gestures on mobile handheld devices because Robertson teaches principles for designing touch-based gestures applicable to various systems using touch-screen inputs and gestures. Ex-1003, ¶96. Robertson describes using multiple command languages (e.g., “X”-based, Unix-based, and Lisp-based commands), but does not limit itself to these languages. Ex-1005, Abstract, §§ 1.2, 2.1, 3-3.2; Ex-1003, ¶96. A POSITA would have expected success implementing Robertson’s principles in other handheld device operating systems, including Palm OS, Windows Pocket PC,

Symbian, etc., that allow for touch and gesture inputs. Ex-1003, ¶96. Implementing Robertson's user interface in a mobile handheld computer unit would result in the ordinary and expected function of providing touch screen user interface in a mobile device using known functions as expected. *Id.*

b. [1a] "a touch sensitive area in which a representation of a function is provided,"

Robertson discloses [1a]. Ex-1003, ¶¶97-101.

Robertson describes a touch sensitive area that accepts "pen-based gestural input[s]" to activate on-screen buttons and associated functions, which a POSITA would understand would include touch screens. Ex-1005, §§ 2 ("touch screens for input" known), 3-3.2; Ex-1003, ¶98. When a user uses a pen to perform a gesture on the touch-sensitive user interface, "a gesture parser interprets … [the] pen movement and classifies it as one of a small set of easily differentiated gestures." Ex-1005, §§ 3-3.1; Ex-1003, ¶99. "Once a [user's] gesture has been identified, the XButton executes the appropriate action (i.e., there is an action for each of the gestures)" corresponding to the gesture. Ex-1005, § 3.1. While the user is performing the gesture, "feedback" is provided as "a mouse track displayed on the screen," which is erased when the gesture is completed and the corresponding function is executed. Ex-1005, § 3.1; Ex-1003, ¶99.

Annotated Figure 1 shows a button representation (e.g., "Phone" button (**blue**)) on the touch-sensitive user interface (**red**). Ex-1005, §§ 3-3.2. The "Phone" button represents a function activated when the user performs a multi-step gesture on the button. Ex-1005, §§ 3-3.2; Ex-1003, ¶100.

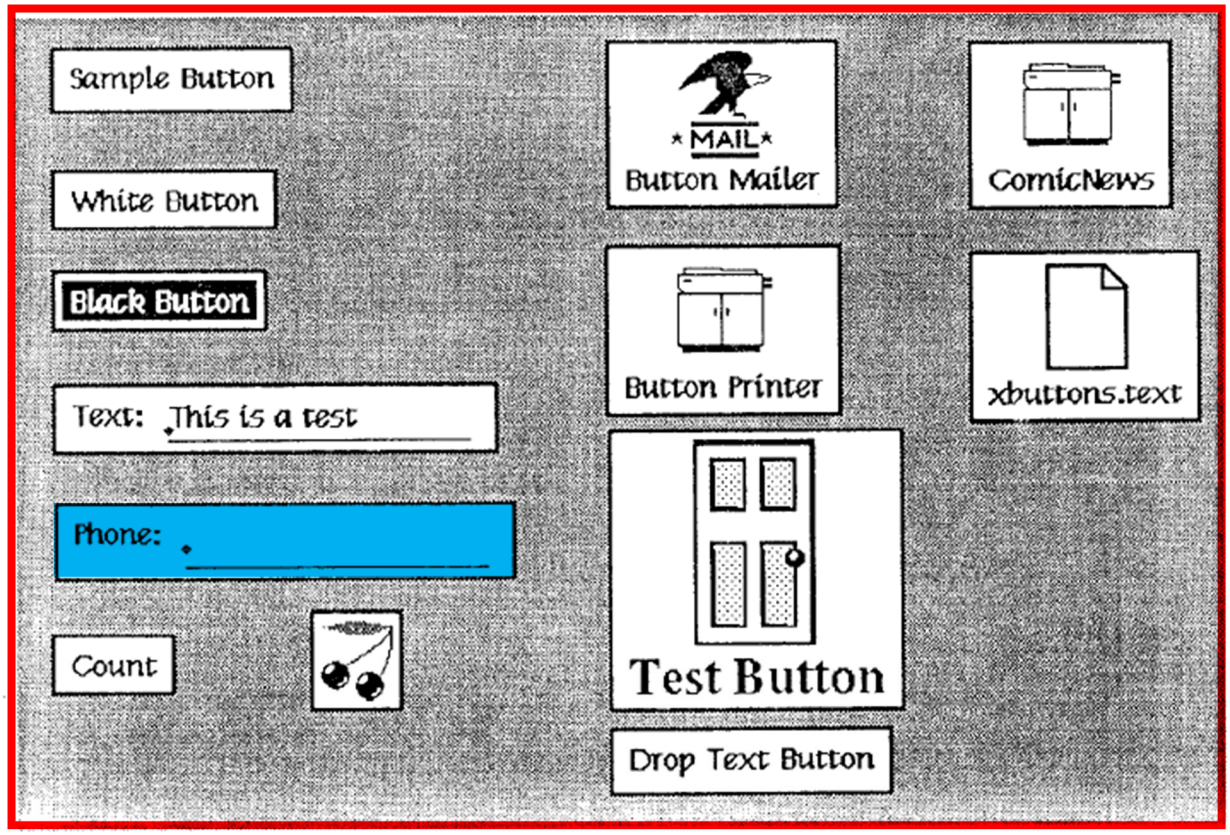


Figure 1: Sample Set of XButtons.

Ex-1005, FIG. 1 (annotated).

Annotated Figures 2 and 3 show functions (**green**) corresponding to multi-step gestures (**orange**) associated with the "Phone" button (**blue**). One function is the "dialphone" function to "dial the [phone] number" activated by a "flick right

gesture." Ex-1005, §§ 3-3.1; Ex-1003, ¶101. Another function, "xbedit," opens the "button editor" menu shown in Figure 3, activated by an "'Insert' gesture (made like an editor's caret)" drawn on the touch screen. Ex-1005, §§ 3.1-3.2; Ex-1003, ¶101.

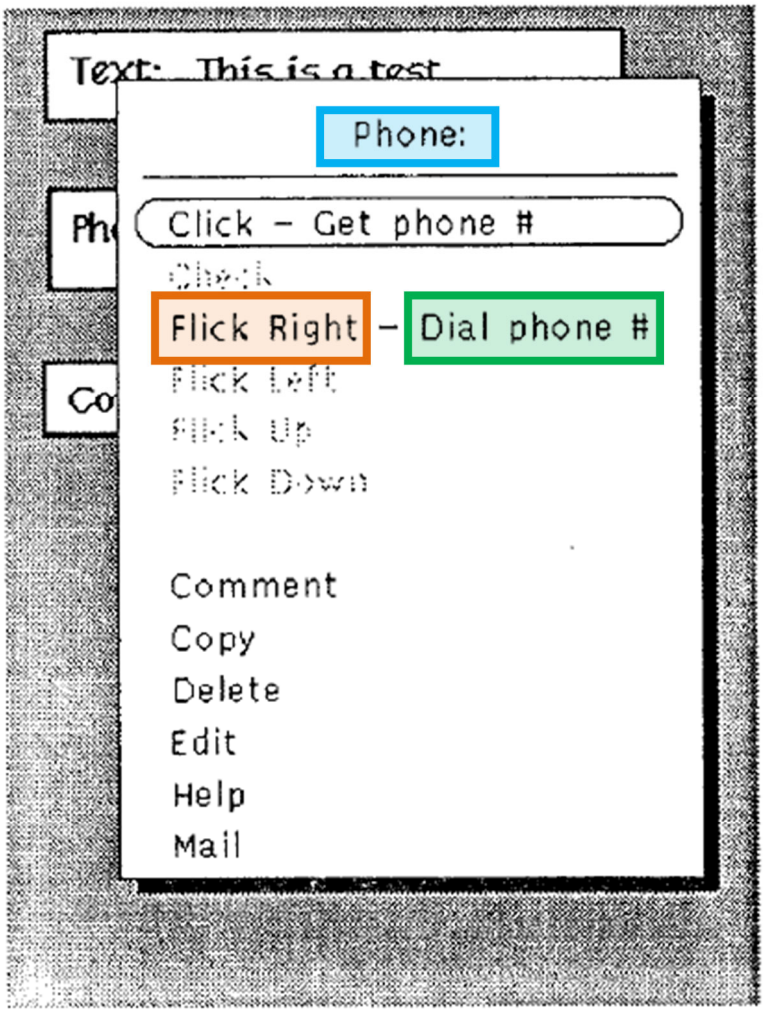


Figure 2: Sample Button Menu.

Ex-1005, FIG. 2 (annotated).

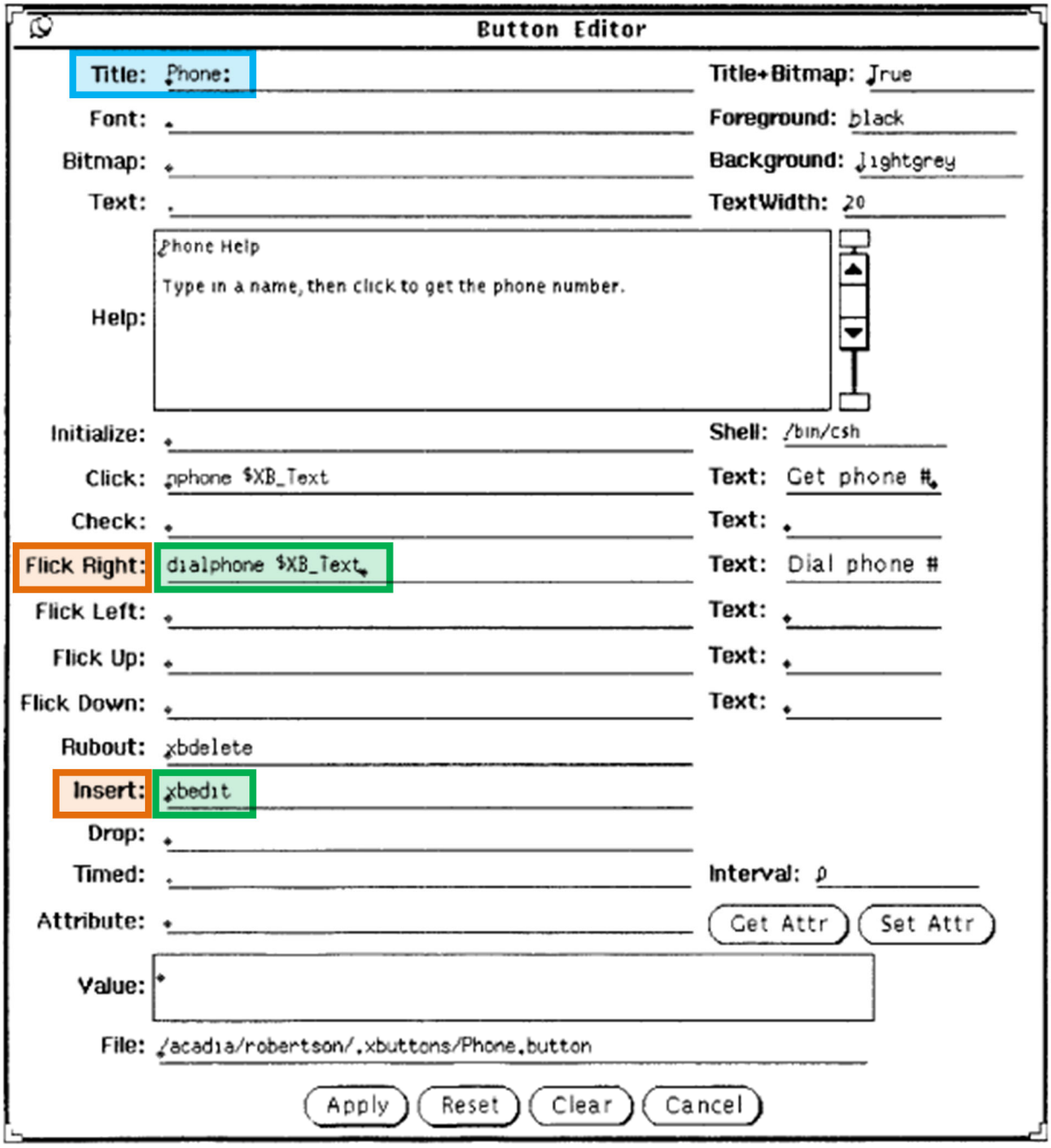


Figure 3: Structured Button Editor.

Ex-1005, FIG. 3 (annotated).

c. [1b] "wherein the representation consists of only one option for activating the function and"

Robertson explains that each function is defined by a specific gesture performed on a button, even if a button may have several functions. Ex-1005, §§ 3-3.2 ("there is an action for each of the gestures"); Ex-1003, ¶¶102-104.

Robertson's "Phone" button (**blue**) activates the "dialphone" function (**green**) by only a "flick right" gesture (**orange**): touching the phone button then sliding the pen to the right. Ex-1005, § 3.1; Ex-1003, ¶104. The phone button activates the "xbedit" function (**bright green**) by only an "Insert gesture [(**light orange**)]" drawn on the screen. Ex-1005, §§ 3.1-3.2; Ex-1003, ¶104.

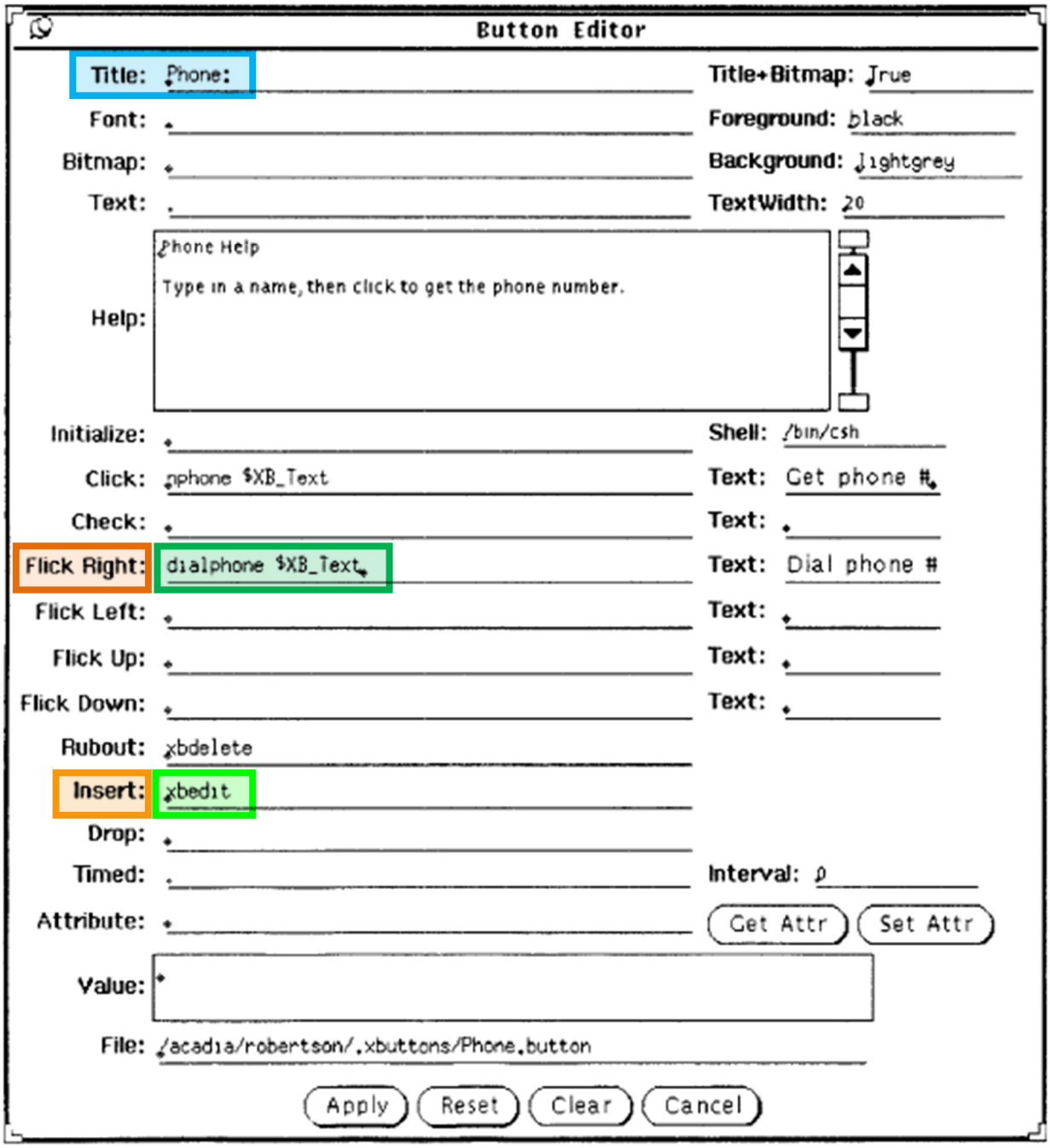


Figure 3: Structured Button Editor.

Ex-1005, FIG. 3 (annotated).

d. [1c] "wherein the function is activated by a multi-step operation comprising (i) an object touching the touch sensitive area at a location where the representation is provided and then (ii) the object gliding along the touch sensitive area away from the touched location,"

Robertson accepts multi-step "pen-based gestural input[s]" from the user to activate a function, which a POSITA would understand corresponds to the user touching a particular button (representation) on a touch-sensitive user interface, e.g., a touch screen, with a pen/stylus (or finger), then performing a movement while touching the touch sensitive area (e.g., screen) with the pen. Ex-1005, §§ 3-3.1, FIGS. 2-3; Ex-1003, ¶¶105-110. Robertson explains that "gesture[s] must start in an XButton," but "can move outside the XButton" while performed. Ex-1005, §§ 3.1, 4.2; Ex-1003, ¶106. A "gesture parser interprets … [the] pen movement and classifies it as one of a small set of easily differentiated gestures (flick left, flick right, flick up, flick down, click, rubout, check, or insert)." Ex-1005, § 3.1. After identifying the gesture, "the XButton executes the appropriate action." Ex-1005, § 3.1; Ex-1003, ¶106.

Robertson's "dialphone" function is activated by a multi-step operation of placing a pen on the phone button, then sliding the pen to the right along the touch-sensitive interface to perform a "flick right" gesture. Ex-1005, §§ 3-3.1; Ex-1003, ¶107. This multi-step operation is shown below, where the user touches the "Phone" button (**blue**, representation) with the pen/stylus and moves (glides) the

pen to the right, away from the initial touched location, which is tracked by the gesture parser to show the line drawn to the right (**orange**). Ex-1005, §§ 3-3.2, 4.2; Ex-1003, ¶107.

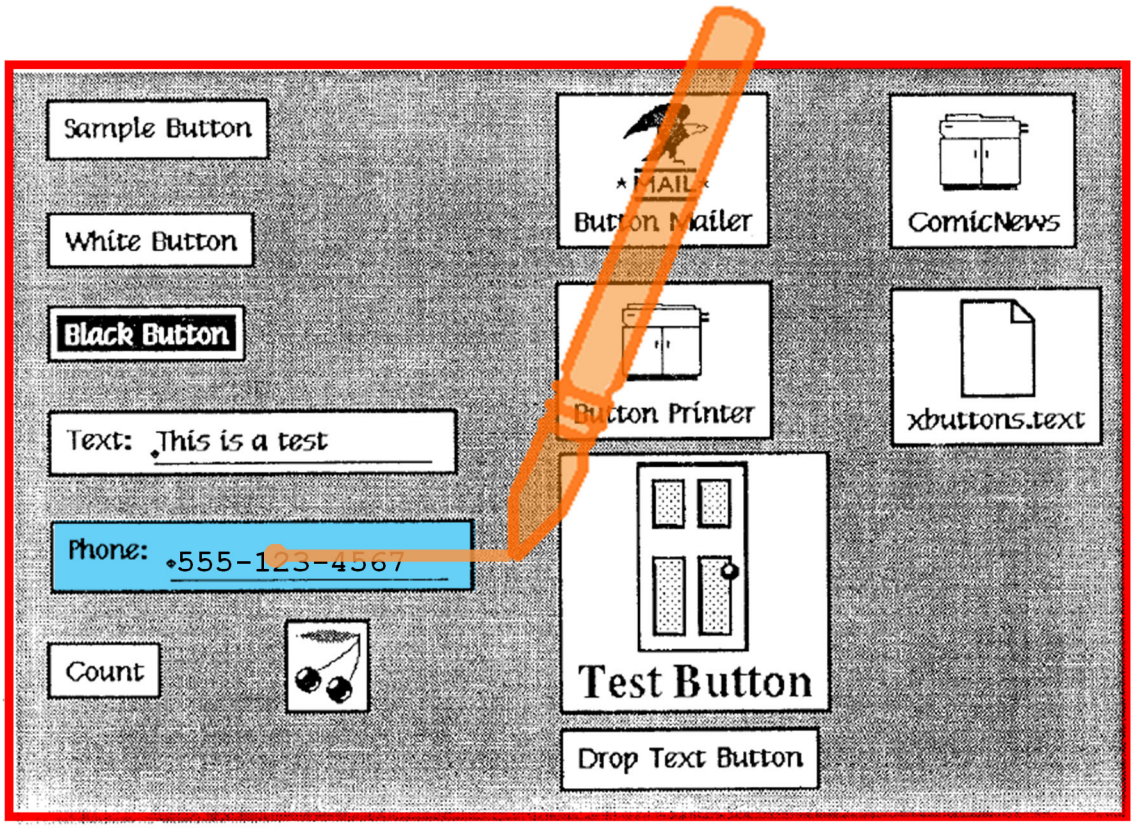


Figure 1: Sample Set of XButtons.

Ex-1005, FIG. 1 (annotated); *also* FIG. 2 ("Dial Phone #" function activated by "Flick Right" gesture), FIG. 3 (assigning "dialphone" function to "Flick Right" gesture).

The "xbedit" function to open a button editor is activated by a multi-step operation of touching a pen (or finger) on the phone button, then sliding the pen

away from the initial touched location in the shape of a caret to perform an "Insert gesture." Ex-1005, § 3.2; Ex-1003, ¶108. For this multi-step operation, the user touches the "Phone" button (**blue**) with the pen/stylus and glides (moves) the pen diagonally up and right, then diagonally down and right in a caret shape (^, **orange**), which is tracked by the gesture parser and drawn on the screen. Ex-1005, §§ 3-3.2, 4.2; Ex-1003, ¶108.

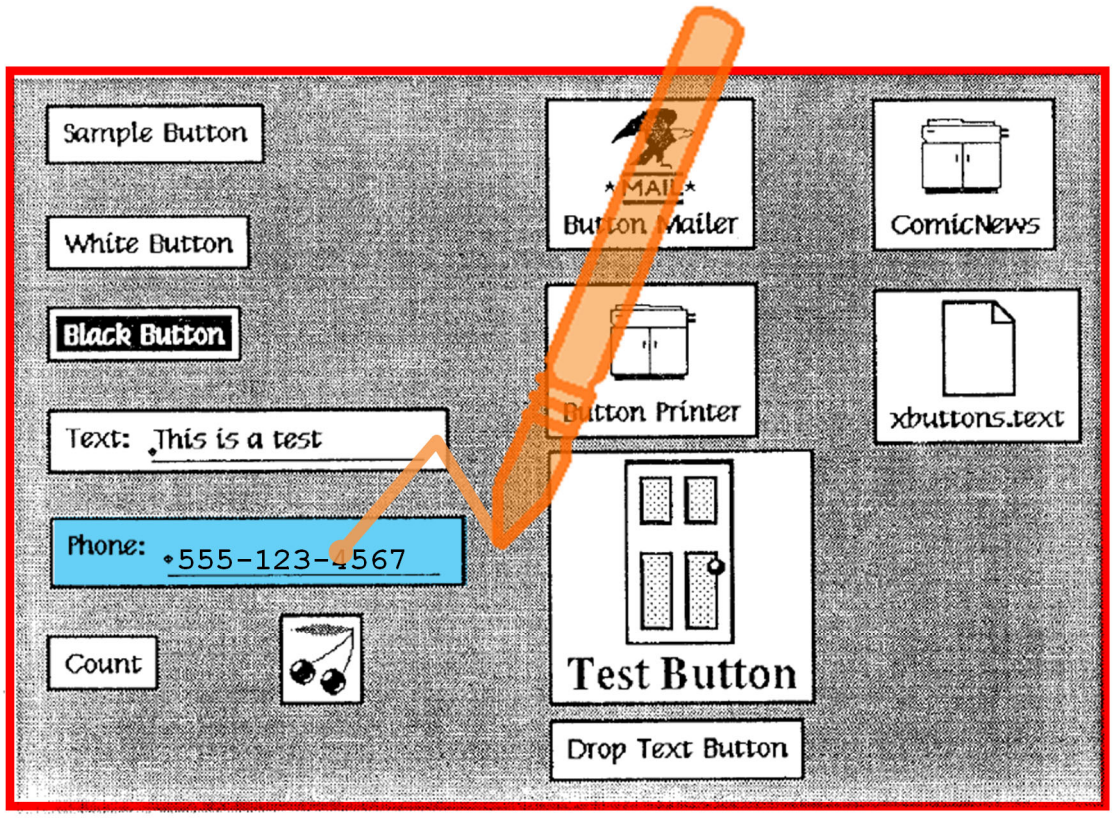


Figure 1: Sample Set of XButtons.

Ex-1005, FIG. 1 (annotated).

Although the multi-step activations for the "dialphone" and "xbedit" functions are detailed, Robertson's teachings are broader, applying to numerous multi-step gestures for activating functions, including "flick left, flick right, flick up, flick down, click, rubout, check, [and] insert" gestures. Ex-1005, § 3.1; Ex-1003, ¶109.

Buttons can be "tailorable," and corresponding actions/functions to configure buttons can use "the operating system command language," or other languages, including Unix and Lisp. Robertson does not limit itself to these languages, and a POSITA would understand Robertson's teachings could apply in numerous well-known gesture-based environments, e.g., touch-screen laptops, PDAs, and other mobile devices. Ex-1005, §§ 1.2, 3.2-3.3; Ex-1003, ¶110. Robertson's button editor "specifies the [button] action language" in "the 'Shell' field," confirming that different action languages are possible depending on the device (or operating system). Ex-1005, § 3.3; Ex-1003, ¶110. Thus, a POSITA would understand Robertson's teachings apply to different types of devices using gestural inputs. Ex-1003, ¶110. A POSITA would understand from Robertson that a button can have access to "the full range of actions" a user can perform on a device, regardless of operating system or language. Ex-1005, § 3.2 ("every action that is possible from the interface (e.g., from the menu) can be assigned to a gesture"); *also* §§ 1 ("wide range of … behaviors is possible" for button

functions), 3-3.1; Ex-1003, ¶110. Robertson states, "it is obvious how to change all the properties of the XButton," showing a POSITA that Robertson envisions different operations and functions being activated by gesture-based inputs, as in Figure 3. Ex-1005, §§ 1.2, 3-3.2, FIG. 3; Ex-1003, ¶110.[4] A POSITA would understand Robertson uses "Insert" (caret) to open the button editor menu, and therefore other gestures, e.g., "flick up," "flick down," or "check," could also open menus, panels, or dialogs, like an address book, list of available programs (like the Windows "Start" program menu), or list of available file directories or folders (e.g., music or photos). Ex-1003, ¶110.

e. [1d] "wherein the representation of the function is not relocated or duplicated during the gliding."

As explained for [1c] (Section V.A.3.d), Robertson's buttons on the touch-sensitive screen are not relocated or duplicated when the user slides the pen away

[4] Although Robertson discusses making buttons user-editable and independent of applications (Ex-1005, §§ 1-1.2), a POSITA would have known that Robertson's gesture-based activations would also apply to buttons within applications that are not user-editable and apply broadly across devices accepting gesture-based inputs, e.g., touch-screen laptops, PDAs, and other mobile handheld devices. Ex-1003, ¶110 n.3.

from the touched location. Ex-1003, ¶¶111-112. A POSITA would have recognized that the button is not relocated or duplicated during gliding because Robertson states that the gesture "can move outside the XButton," indicating that the button is fixed during the gesture, and Robertson provides no indication of any duplication or relocating. Ex-1005, § 4.2; Ex-1003, ¶112.

4. [Claim 2] "wherein the function, when activated, causes the user interface to display icons representing different services or settings for a currently active application."

Robertson's XButtons run with a program or application on the user device, called the "XButtons Server." Ex-1003, ¶¶115-118; Ex-1005, §§ 4-4.1.[5] When "xbedit" is activated, it communicates with the XButtons Server application on the device, causing the user interface to display a button editor with services and settings for the button that is part of the active XButtons Server application, including settings (yellow) for the button title, image (bitmap), background color, and foreground color; and services (pink), like "Get Attr[ibute]," "Set Attr[ibute]," "Reset," and "Clear," as shown. Ex-1005, §§ 3.2-3.3 (describing other settings), FIG. 3; Ex-1003, ¶116.

[5] When implementing Robertson's teachings in non-XButton software, a similar application would be running to control the button's functions. Ex-1003, ¶¶93, 115.

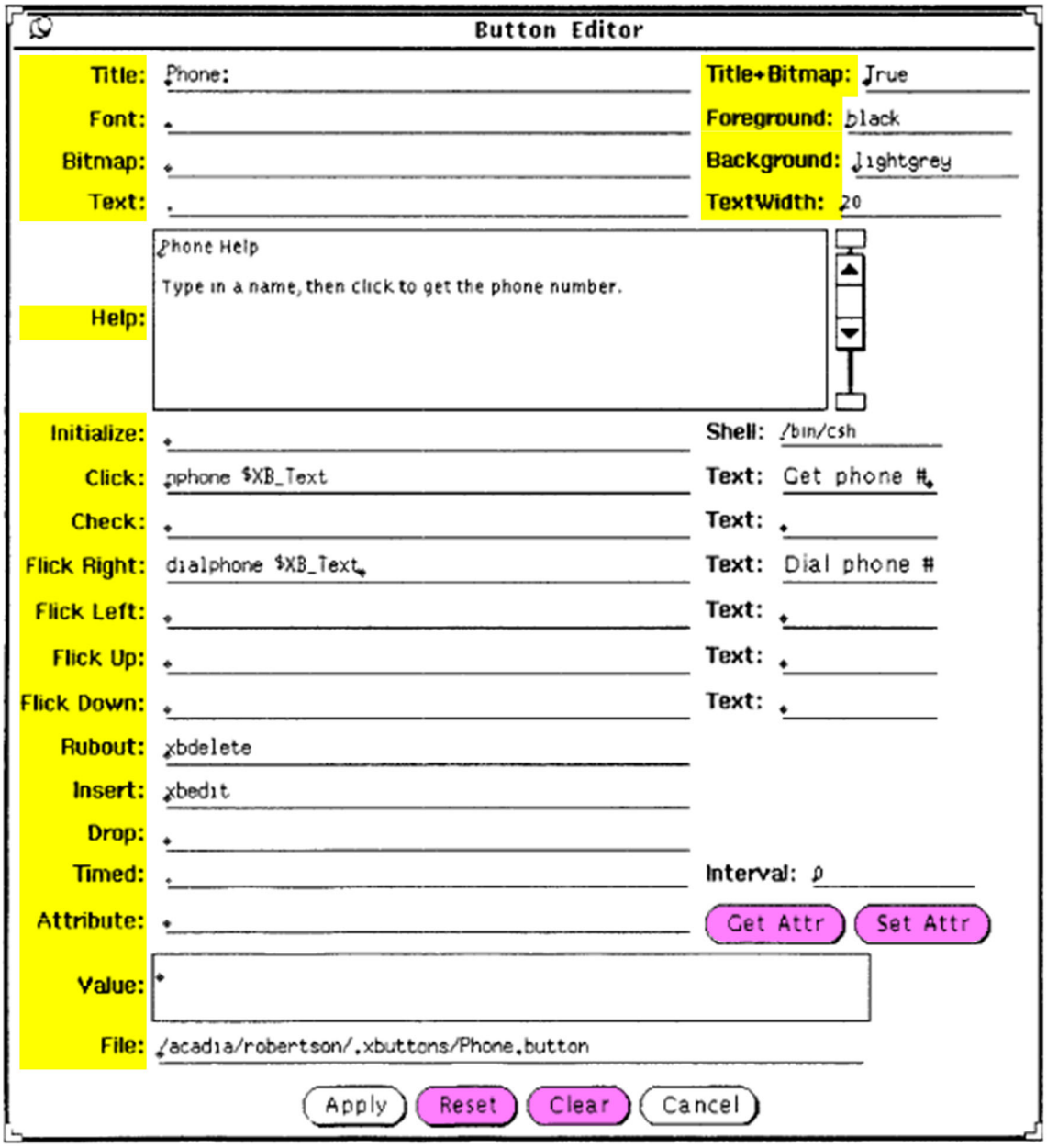


Figure 3: Structured Button Editor.

Ex-1005, FIG. 3 (annotated).

Robertson's buttons are "similar to familiar icons" and include icons with button text, like the eagle and printer icons in Figure 1. Ex-1005, §§ 3, 4.2 ("Xerox document icon"), FIG. 1; Ex-1003, ¶117.

A POSITA would have found it obvious to include icons in Robertson's list of different services or settings for the active application (e.g., XButtons Server application, button editor application, phone application, desktop application, or other application) to facilitate the user's ability to understand the available services or settings presented. Ex-1003, ¶118.

i. Motivation

A POSITA would have been motivated to display icons with various services or settings to intuitively indicate the meaning of fields or options (services or settings) available to the user in Robertson's button editor. Ex-1003, ¶119. Robertson displays buttons, icons, and text together to indicate button functions. Ex-1005, § 3.

A POSITA would have been motivated to display icons (alone or with text) representing services and settings to help users understand and visually locate the available services and settings, making user interactions more efficient. Ex-1003, ¶120. The button editor may display icons corresponding to available fonts and foreground and background colors, prompting the user to select settings supported by the user interface. A POSITA would have been motivated to provide icons of

the gestures, not just words, so the user can assign functions to the gesture he/she feels most naturally aligns with the function. *Id.*

Similarly, a POSITA would have been motivated to include icons representing various services in Figure 2, like a "?" icon for "Help," envelope for "Mail," phone for "Dial phone #," and icons for each gesture, to illustrate the service in a compact, intuitive manner. Ex-1003, ¶121. A POSITA would have recognized that displaying such icons would allow users to review current settings without opening each setting, simplifying user interactions with a configuration menu, like the button editor. *Id.*

Where gestures are assigned to other functions, e.g., opening menu functions (*supra* § V.A.3.d ([1c])), a POSITA would have also been motivated to configure the menus to display both text and icons to identify the service or setting listed in the menu, as commonly used in word processing or spreadsheet programs. Ex-1003, ¶122. A POSITA would have been motivated to include such icons (alone or with text) indicating a service or setting to facilitate the user's ability to identify the service (e.g., program or application, "search" function, etc.) or setting (e.g., font color, screen brightness or contrast, etc.). *Id.*

ii. Expected success

A POSITA would have expected success in displaying icons to indicate services or settings for an active application, like Robertson's button editor, button

menu, or other activated menu function, because using icons to identify a corresponding service or setting was routine and conventional. Ex-1003, ¶123. Robertson displays icons to represent buttons, e.g., "Button Printer," "Button Mailer," shown in Figure 1. Ex-1005, FIG. 1. Using icons alone or with text was commonplace for many applications, including menus in word processing programs (e.g., font type, font color, etc.), services (e.g., save, open, new document), spreadsheet programs, application lists (e.g., Windows "Start" menu), file directories (icons for different file types, e.g., music, spreadsheets, photos), as known to POSITAs. Ex-1003, ¶123. Accordingly, displaying icons for services or settings in an activated menu, like the editor in Figure 3 or button menu in Figure 2, for an active application, would provide ordinary, expected results of displaying an icon in the displayed list using known functions operating as expected. *Id.*

5. [Claim 3] "wherein the user interface is characterised in, that a selection of a preferred service or setting is done by tapping on a display icon corresponding to the preferred service or setting."

As explained for claim 2 (Section V.A.4), Robertson renders obvious displaying icons to indicate services and settings, e.g., available button editor options, button menu, or other activated function (e.g., application menu).

Robertson renders obvious tapping the displayed icon to select a service or setting. Ex-1003, ¶¶126-128.

"[C]licking" and "pressing" were simple operations to execute a button function or option displayed on touch-sensitive user interfaces, like touch screens. Ex-1005, §§ 1.1-1.2, 2, 3-3.1; Ex-1003, ¶127. Robertson's icons are similar to buttons and can be activated by pen-based clicking or pressing (tapping) on the icons. Ex-1005, §§ 1, 2 ("Icons are like degenerate buttons."), 3 (buttons are "similar" to icons).

A POSITA would have found it obvious to select a preferred service or setting being displayed for an application, as presented in an activated menu (*supra* § V.A.4 (claim 2)) by tapping the icon using a pen ("clicking" or "pressing" icon with pen) because tapping is a simple, fast operation to select a desired option that is intuitive to users. Ex-1003, ¶128.

i. Motivation

A POSITA would have been motivated to allow users to select a displayed icon representing a service or setting by tapping the icon because tapping using a pen/finger was a conventional "click" operation for touch-sensitive user interfaces. Ex-1003, ¶129. A POSITA would have been motivated to use tap-based selection based on Robertson's "click" on a button icon to perform an associated action. Ex-1005, § 3; Ex-1003, ¶129.

A POSITA would have specifically been motivated to allow the user to tap a setting icon in an activated menu, like the "foreground" or "background" settings, to edit the corresponding setting, e.g., available foreground/background colors, preventing the user from entering a setting unsupported by the device. Ex-1003, ¶130.

A POSITA would also have found it obvious to use tapping to select a preferred service from a menu (e.g., "Help," "Edit," "Mail," "Comment" in Figure 2), a simple operation for a user to select a desired service. Ex-1003, ¶131. A POSITA would have been motivated to allow a user to tap an icon to select a preferred service, e.g., "Help," because tapping corresponds to the conventional interface selection familiar to users. *Id.*

ii. Expected success

A POSITA would have expected success in selecting a preferred service or setting by tapping a displayed icon because tapping was a conventional selection action known to POSITAs for touch-sensitive user interfaces. Robertson teaches "clicking" and "pressing" icons using a pen to trigger an action. Allowing a user to press (tap) on a displayed icon to select the preferred setting or service would result in executing a selection action in response to a user's tap gesture operating as expected. Ex-1003, ¶132.

6. [Claim 4] "wherein the function, when activated, causes the user interface to display a keyboard and a text field."

Robertson's "Insert" (caret) gesture opens a button editor with text fields (yellow) next to "Help," "Text," and other settings, as shown. Ex-1005, §§ 3.2-3.3.

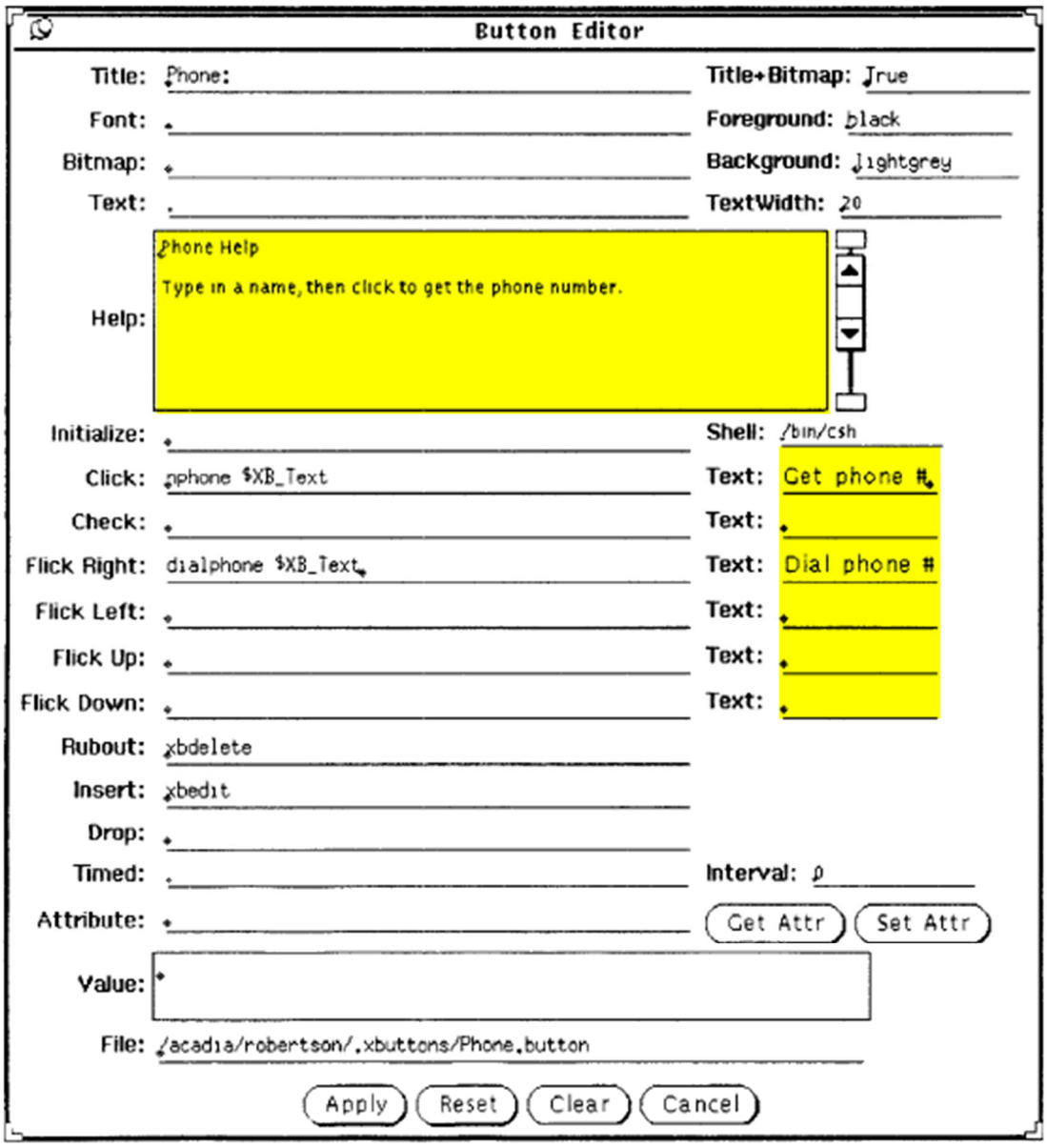


Figure 3: Structured Button Editor.

Ex-1005, FIG. 3 (annotated).

It would have been obvious to a POSITA to cause the user interface to display a keyboard with a text field while the button editor is activated so users can input and edit text, e.g., "Help" explanation or "Text" descriptions, based on Robertson's descriptions of "editable type-in field[s]" that users can "type" in. Ex-1005, §§ 1.2, 3, FIG. 3; Ex-1003, ¶136.

Maddalozzo renders obvious displaying an on-screen touch-sensitive keyboard (**purple**) along with a text field (**orange**) to input and edit text, as shown. Ex-1003, ¶137; Ex-1006, 6:36-61.

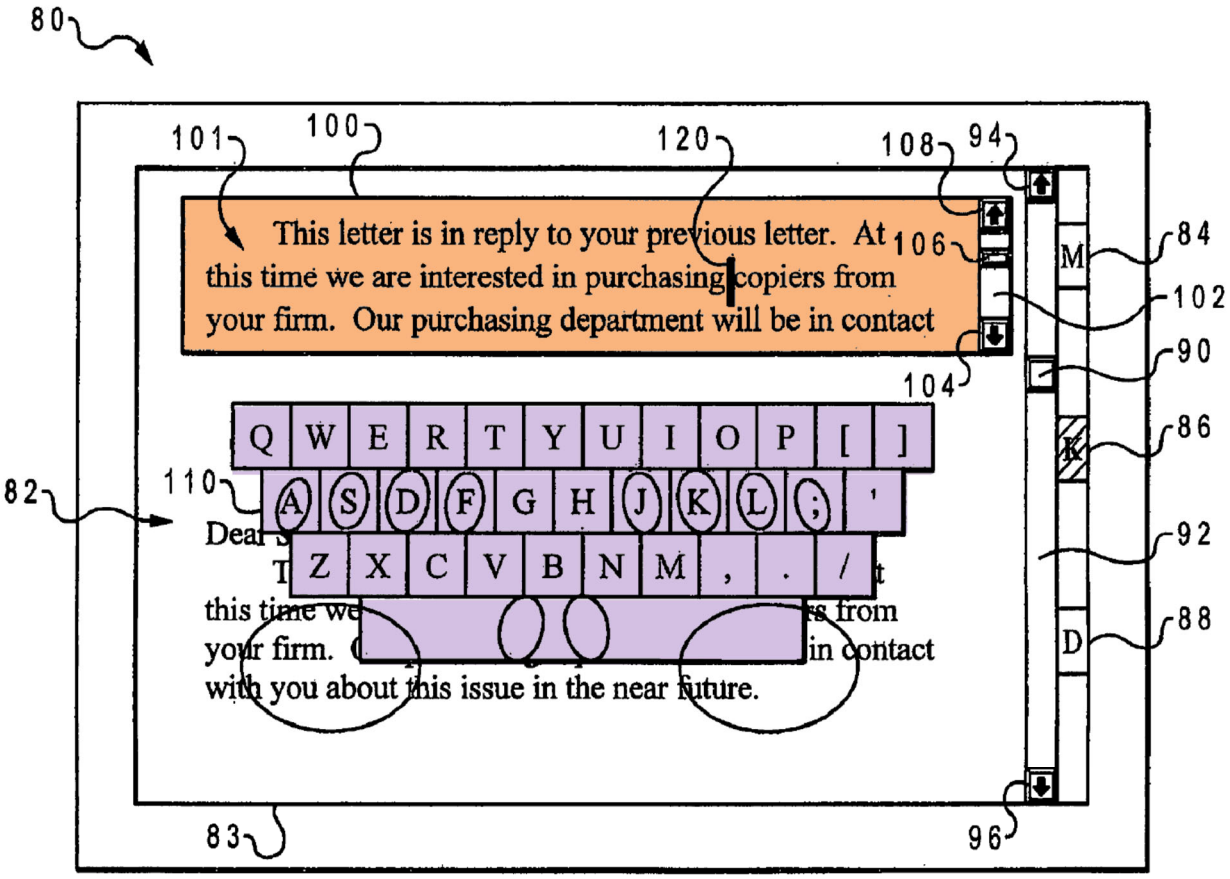


Fig. 6

Ex-1006, FIG. 6 (annotated).

A POSITA would have found it obvious to display Maddalozzo's touch-sensitive keyboard with the text field for entering and editing text ("Help" text or "Text" labels) to eliminate the need for an external keyboard and so the user can see the text field while typing. Ex-1003, ¶138.

i. Motivation

Robertson discloses button editor text fields where users input and edit text. Ex-1005, §§ 3.2-3.3; Ex-1003, ¶139. Robertson also discloses "type-in" fields for entering button text. Ex-1005, §§ 1.2, 3, FIG. 3. A POSITA would have been motivated to display a touch-sensitive keyboard when users input and edit text fields in the button editor, e.g., taps on a text field to select it. Ex-1003, ¶139. A POSITA would have been motivated to display a touch-sensitive on-screen keyboard to eliminate the need for an external keyboard, simplifying construction by removing mechanical components and creating a more portable device. *Id.* A POSITA would also have been motivated to display the touch-sensitive keyboard and text field to simplify the user entering and editing text, because the user can see text being typed/edited while typing. *Id.*

ii. Expected success

A POSITA would have expected success in displaying a touch-sensitive keyboard and text field when Robertson's button editor is activated because displaying a keyboard when inputting text on a touch-sensitive device (touch screen) was routine and conventional. Ex-1003, ¶140. A POSITA would have known touch-sensitive keyboards were conventional for pen-based input devices, like touch screens, before the '879 patent filing. Ex-1003, ¶140. Dutta (Ex-1011, [0005]-[0007]) confirms displaying touch-screen keyboards and text fields when

inputting text on mobile handheld computing devices was well-known to POSITAs. Ex-1003, ¶140. Displaying a touch-sensitive keyboard and text field on Robertson's screen when the button editor is activated involved only the known application of conventional technologies operating in the ordinary and predictable manner. *Id.*

7. [Claim 5] "wherein said text field is used for inputting and editing of text through said keyboard."

As in claim 4 (Section V.A.6), Robertson and Maddalozzo render obvious the text field used for inputting and editing text through the keyboard, e.g., inputting and editing text in the "Help" or "Text" fields. Ex-1003, ¶143.

8. [Claim 13] "wherein the user interface is characterised in, that said representation of said function is located at the bottom of said touch sensitive area."

Robertson discloses buttons (representations of functions) (**blue**) located at the bottom of the touch-sensitive screen (**red**), e.g., "Drop Text Button" in annotated Figure 1. Ex-1003, ¶¶146-150. A user can open or activate the button editor (function) for the "Drop Text Button" using an "insert" (caret) gesture. *Supra* § V.A.3.d ([1c]); Ex-1005, §§ 3-3.2; Ex-1003, ¶146.

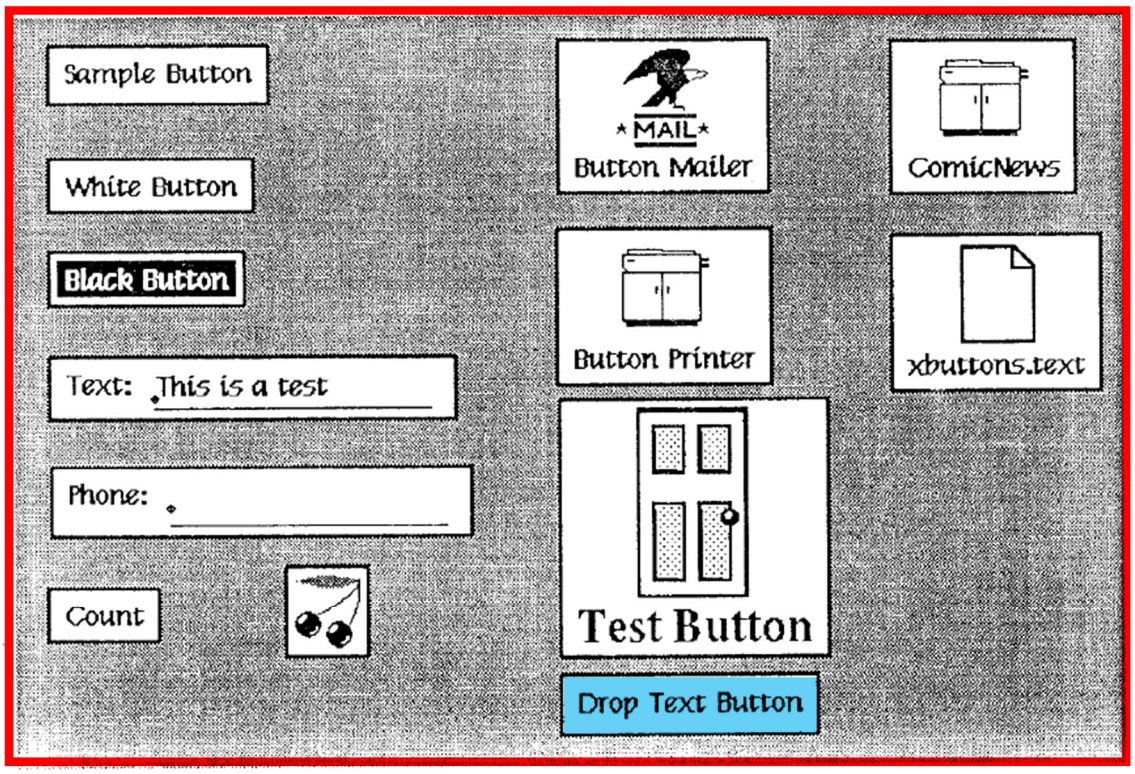


Figure 1: Sample Set of XButtons.

Ex-1005, FIG. 1 (annotated).

A POSITA would understand that any of Robertson's buttons, e.g., "Phone," could be located at the bottom of the touch sensitive area because buttons can be "mov[ed] … to a new location." Ex-1005, §§ 3, 3.3; Ex-1003, ¶147.

A POSITA would have found it obvious to move the "Phone" button to the bottom of the touch-sensitive user interface to make it accessible to the user, to free space for other applications to be displayed, and have it in an easy-to-locate area of the screen, conventional for frequently used buttons. Ex-1003, ¶¶148-149. A POSITA would have also been motivated to place commonly used buttons, e.g.,

"Phone," buttons to open an application list, menu buttons, or search buttons, at the bottom of a display for mobile handheld devices, like laptops and PDAs. Ex-1003, ¶149. For example, Dutta (Ex-1011), Figure 3, shows such buttons at the bottom of the touch screen. Ex-1003, ¶149.

A POSITA would have expected success in locating Robertson's gesture-activated buttons at the bottom of a touch screen because it would result in the ordinary, expected function of specifying the location on the screen using known, conventional functions operating as expected. Ex-1003, ¶150.

9. [Claim 15] "characterised in, that said computer program code is adapted to function as a shell upon an operating system."

Robertson discloses "the default [button action language] is the operating system command language." Ex-1005, § 1.2; Ex-1003, ¶153. "[T]he full range of actions the user can do from the Shell can be encapsulated into on-screen buttons," and the button editor "specifies the action language (the 'Shell' field)." Ex-1005, §§ 3.2-3.3; Ex-1003, ¶153. Robertson explains "for X, this is the Unix Shell," but a POSITA would understand other shells and operating systems (e.g., Palm OS and Symbian OS, which accept gesture-based inputs) are possible. *Supra* §§ V.A.3.a, V.A.3.d ([1Preamble], [1c]); Ex-1005, § 1.2; Ex-1003, ¶153. Robertson identifies other shell command languages, like Lisp, for controlling button operations. Ex-1005, §§ 1.2, 3.2; Ex-1003, ¶153.

10. [Claim 16] "wherein the representation is finger-sized."

Robertson uses an object, e.g., pen or stylus, to make gestures on a touch-sensitive user interface, such as touch screens, to activate functions. Ex-1005, §§ 3-3.1; Ex-1003, ¶156. A POSITA would have recognized that fingers and styluses were interchangeable for touch-sensitive devices, and thus would have found obvious making Robertson's buttons finger-sized. Ex-1003, ¶¶155-157.

Maddalozzo also renders obvious that Robertson's gesture-activated button is finger-sized to simplify interactions. Ex-1003, ¶157. Maddalozzo's user interacts with a touch screen using fingers, and buttons can be sized according to a width of the user's fingers. Ex-1006, 1:23-25, 5:31-36, 6:48-61, 9:48-53; Ex-1003, ¶157. A POSITA would have found it obvious to incorporate finger-based inputs and finger-sized representations into Robertson's touch-sensitive user interface to enable finger-based gestures for activating Robertson's button functions. Ex-1003, ¶157.

i. Motivation

Robertson and Maddalozzo both teach touch-based input devices to activate functions. Ex-1005, §§ 3-3.1; Ex-1006, 5:31-39, 6:28-35, 6:58-61, 7:12-23; Ex-1003, ¶158. Robertson teaches "pen-based gestural input[s]" and Maddalozzo teaches a finger-based user interface with finger-sized buttons. Ex-1005, § 3.1; Ex-1006, 9:48-53; Ex-1003, ¶158. A POSITA would have been motivated to make

Robertson's buttons (representations) finger-sized and activatable by a user's fingers, like Maddalozzo, to allow users to control the interface without a pen/stylus, because a pen/stylus may be lost. Ex-1003, ¶158. Further, a POSITA would have been motivated to use finger-based input to decrease manufacturing costs for a stylus. *Id.* Moreover, finger-sized buttons and finger-based input allow users to interact with the touch screen using both hands rather than only the hand holding the stylus, enhancing the user's experience interacting with the touch screen. *Id.*

ii. Expected success

It was known to POSITAs that finger-based inputs were interchangeable with pens/styluses to provide inputs on touch-sensitive screens. Ex-1003, ¶159. *See* Ex-1012 (Ausems), [0014], [0037], [0046] (describing using finger or stylus touch-screen interactions). Finger-sized buttons were known in mobile handheld units with touch screens for inputs by stylus or finger. Ex-1006, 9:48-53; Ex-1003, ¶159. Making Robertson's buttons finger-sized would require only determining the button size without changing the function, involving only the application of conventional technologies in predictable ways. Ex-1003, ¶159.

11. [Claim 17] "wherein the location where the representation is provided does not provide touch functionality for a different function."

The location of the "Phone" button (representation) includes only a phone function and not touch functionality for a different function, e.g., printing. Ex-1005, § 3; Ex-1003, ¶¶162-165.

If Patent Owner argues Robertson does not disclose this claim, a POSITA would have found it obvious to limit Robertson's button to one function to simplify the user interface. Ex-1003, ¶163. Robertson allows a user to define gestures and actions for buttons and states "it is obvious" to change button properties. Ex-1005, §§ 1.2, 3.3, FIG. 3; Ex-1003, ¶163. Figures 2 and 3 show several unassigned gestures assigned, and Robertson suggests "it [was] obvious" to provide a button with one function and use one gesture to activate it. Ex-1003, ¶163. A POSITA would have found it obvious to define a file explorer or application button (like in Dutta) to browse available documents or applications on the device with only that one function to open browsing. Ex-1003, ¶163; Ex-1011 (Dutta), FIG. 3 ("APPLICATIONS" button with function to only browse applications).

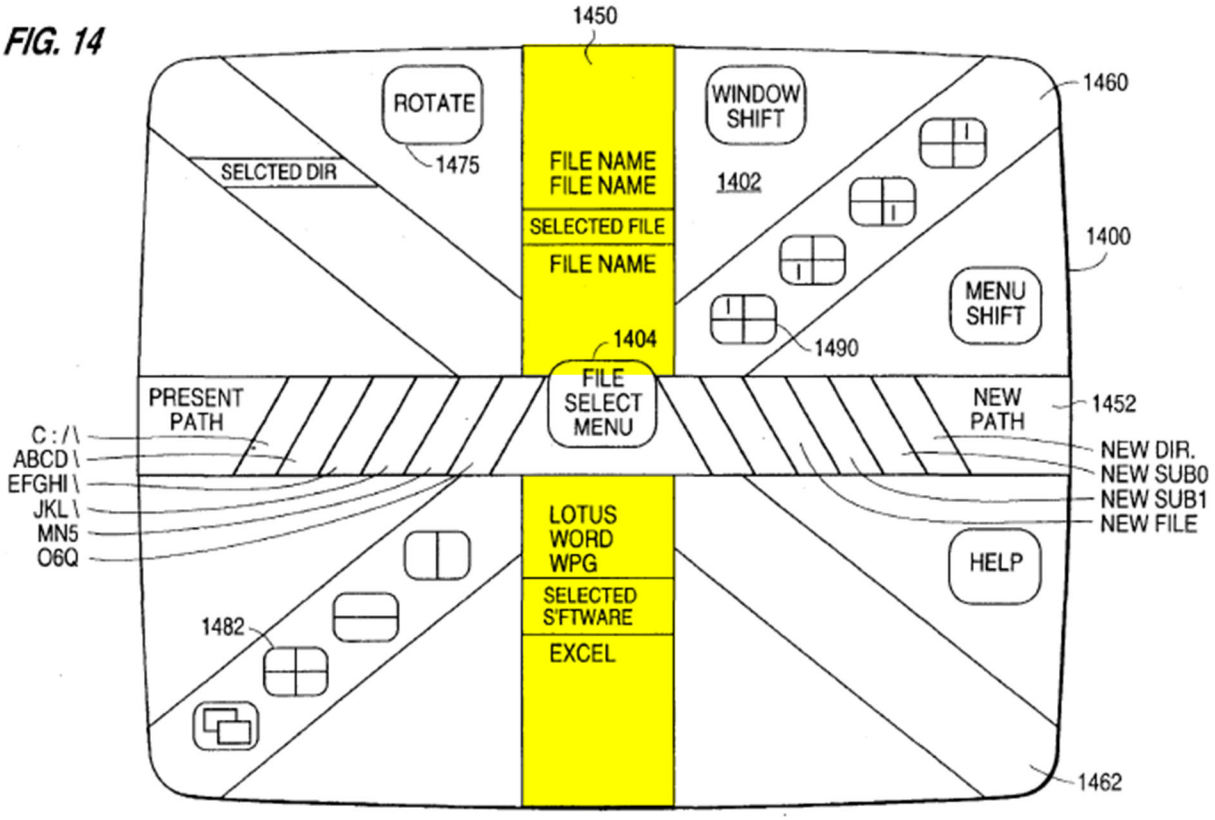
A POSITA would have been motivated to provide only one function for a file explorer or application search button to simplify the button interface and to use a multi-step gesture (e.g., "flick," "insert," "check") to prevent accidental activation, e.g., for buttons performing irreversible actions. Ex-1003, ¶164.

A POSITA would have expected success providing only one touch functionality for the location where the representation is provided. Ex-1003, ¶165. Robertson allows a user to define gestures and actions for buttons, and states "it is obvious how to change" all button properties. Ex-1005, §§ 1.2, 3.3, FIG. 3; Ex-1003, ¶165. Providing only one function for a button merely requires not defining more than one gesture and functionality, involving known functions operating as expected. Ex-1003, ¶165.

B. [Ground 2] Claims 6-7, and 9 are rendered obvious by Robertson, Maddalozzo, and Vayda

1. Overview of Vayda

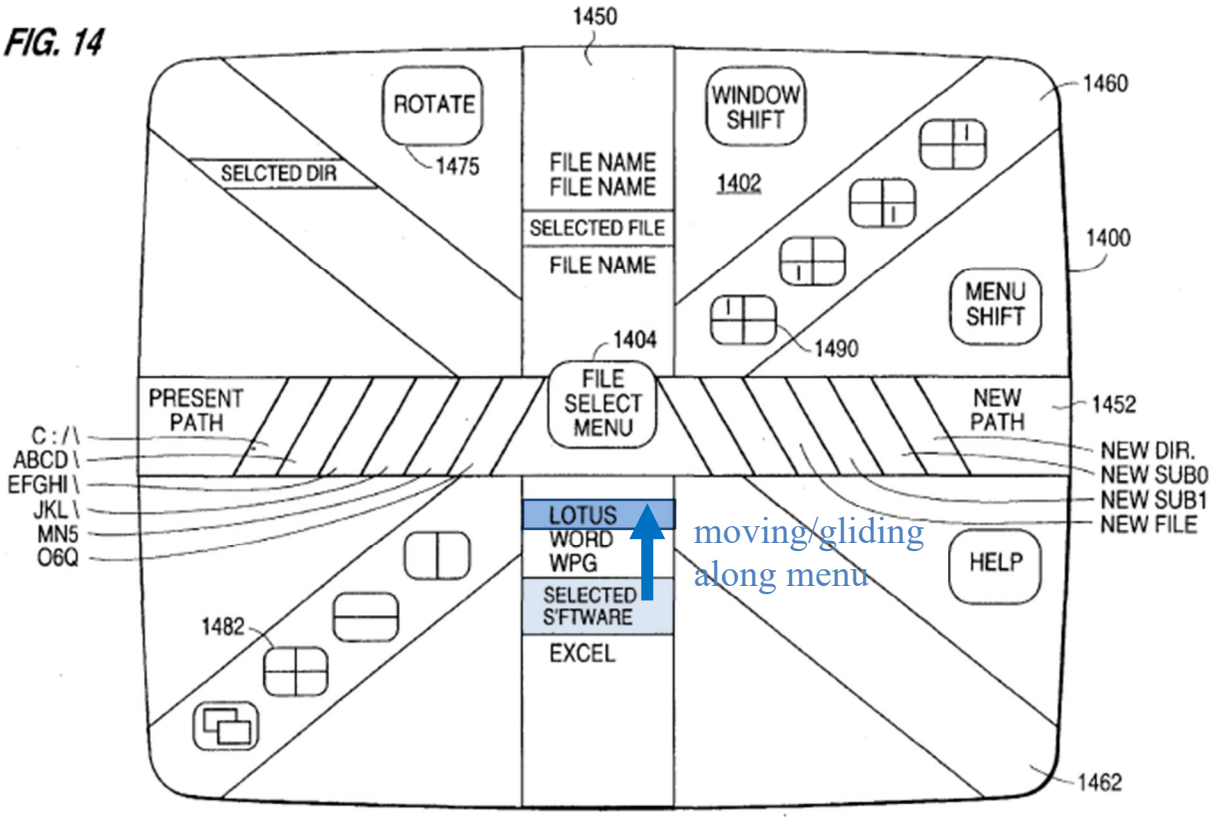
Vayda discloses a computing device that displays a menu screen activated by touch-screen input. Ex-1007, Abstract, 6:13-18, FIG. 14. Figure 14 illustrates a "File Management Menu" with vertical command bar 1450, displaying a list of applications and files on the device that the user can select using a "touch screen" or "stylus" input device. Ex-1007, FIG. 14, 4:45-50, 10:19-20.



Ex-1007, FIG. 14 (annotated).

Vayda discloses navigating menu items using a moveable marker ("scroll position") highlighting a "selected" file or application. Ex-1007, 15:40-42, FIGS. 14-15. The scroll position can be glided along the command bar by an object (e.g., a stylus or a finger) to "indicate [highlight] the currently selected command along that command bar" ("SELECTED FILE" or "SELECTED S[O]FTWARE") shown in annotated Figure 14. Ex-1007, 15:40-56. In operation, the user can "scroll through the commands on [the] command bar" "by movement of user input device 106 in either direction along the bar desired" by moving or gliding input device 106 (e.g., finger or stylus on a touch screen) along vertical command bar 1450,

which moves the highlighted scroll position to "indicate the currently selected" item (i.e., position scrolled to by user) along the list menu. Ex-1007, 15:40-56, 4:41-51; Ex-1003, ¶168. This is shown in annotated Figure 14.



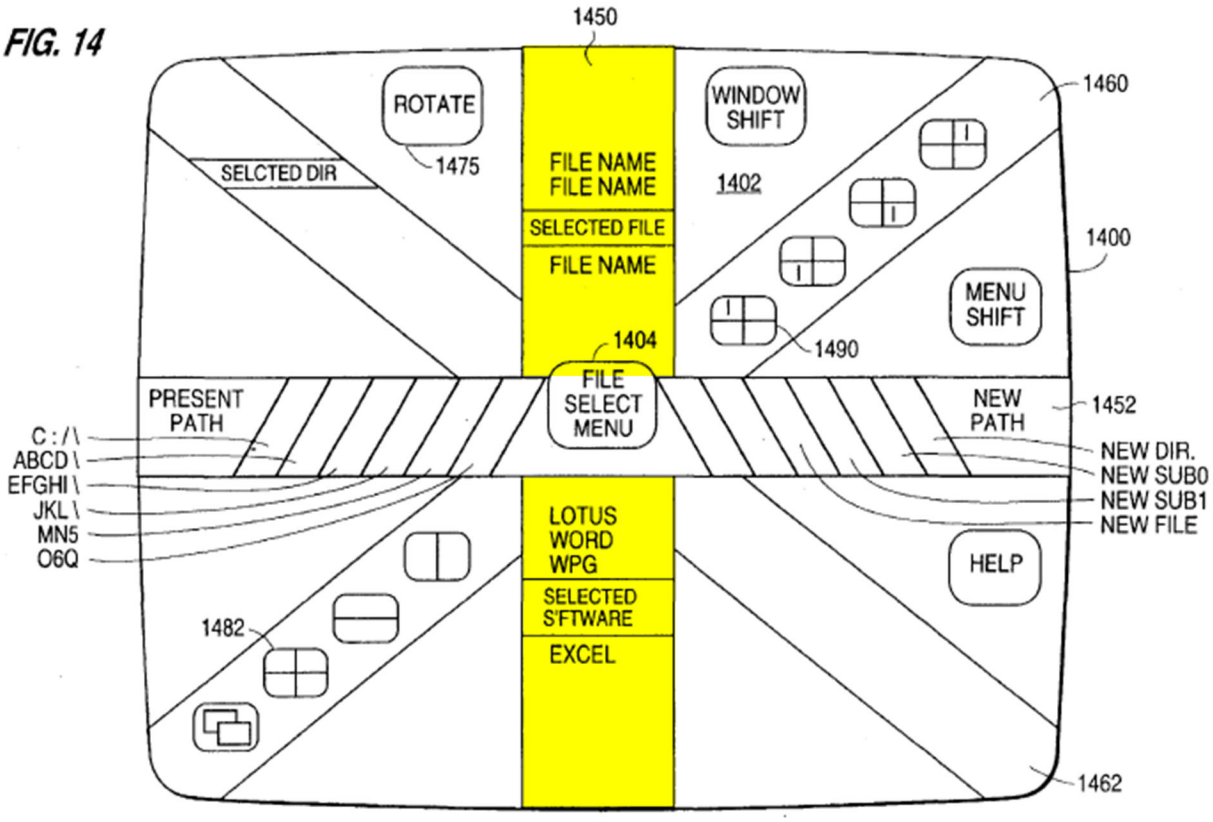
Ex-1007, FIG. 14 (annotated).

2. [Claim 6] "wherein the function, when activated, causes the user interface to display a list with a library of available applications and files on the mobile handheld computer unit."

As explained for claims 1, 2, and 17 (Sections V.A.3, V.A.4, V.A.11), a POSITA would understand Robertson suggests and renders obvious any applicable command being assigned to a gesture, including opening menus, opening a

program menu list, search functions, or file explore/directory functions. Ex-1003, ¶¶170-172; Ex-1005, §§ 3.2-3.3.

Vayda discloses displaying a menu screen activated, like Robertson, by a touch input, like a touch screen. Ex-1007, 4:45-51, 6:13-18; Ex-1003, ¶171. As explained in Section V.B.1, Figure 14 shows vertical command bar 1450 of the menu displaying a list of available applications and files that the user can select using a "touch screen" or "stylus" input device. Ex-1007, FIG. 14, 4:45-51, 10:19-20.



Ex-1007, FIG. 14 (annotated).

A POSITA would have found it obvious to open a list of available files and applications, as taught by Vayda, in Robertson's user interface to enable the user to find a file or application to open or execute on the mobile handheld computer unit using simple gestures without the need for an external input device, e.g., mouse or keyboard. Ex-1003, ¶172.

i. Motivation

Both Robertson and Vayda describe graphical user interfaces for gesture-based touch-sensitive devices. *Supra* §§ V.A.1 (Robertson), V.B.1 (Vayda); Ex-1005, §§ 2, 3-3.1; Ex-1007, 4:45-51, 6:13-18; Ex-1003, ¶173. Robertson discloses a broad framework of gesture-based inputs to activate functions, like opening menus, directories, etc. (including commands defined by users). *Supra* § V.A.3 (claim 1); Ex-1005, §§ 2, 3, 3.2; Ex-1003, ¶173. A POSITA would have been motivated to use Robertson's touch-based gestures, e.g., "flick up," "flick down," or "check," to open a menu (like vertical command bar 1450) showing available files and applications to allow a user to access a desired file, directory, or application using gestures on a handheld device, eliminating the need for an external keyboard or mouse, while still permitting the user a broad range of navigation. Ex-1005, §§ 1.2, 3-3.2; Ex-1003, ¶173. A POSITA would have been motivated to open a list of available files and applications using gesture-based operations on a touch-screen mobile device, because Vayda explains gesture-

driven menus "provide operation of menu driven programs in a more efficient and less time consuming manner." Ex-1007, 1:51-53, 16:41-44, 6:13-18 (identifying the activation for a select/execute command menu as a touch screen), 17:5-10 (cross-configuration menus can be implemented in touch screens); Ex-1003, ¶173.

ii. Expected success

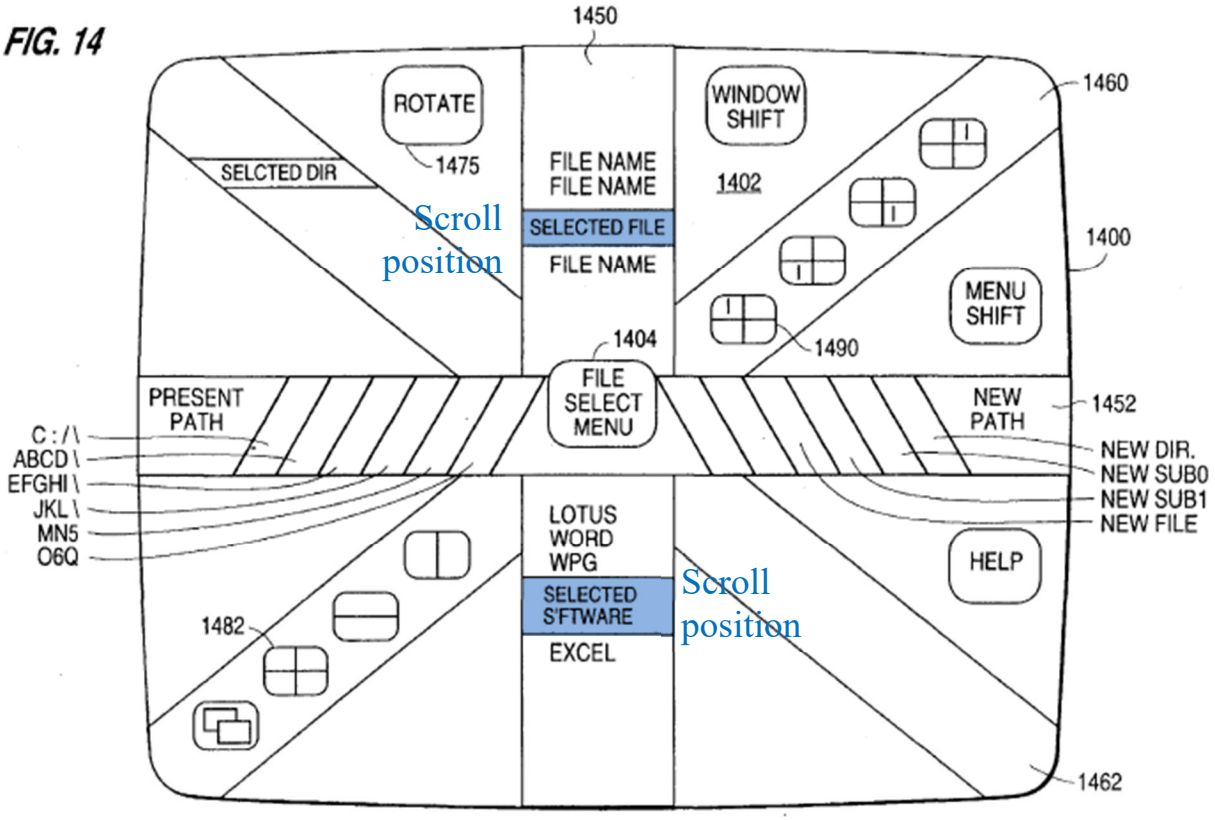
Robertson activates a function using a multi-step gesture from "the full range of actions" available in the "command" or "action" language, like searching the device (e.g., by typing a name) or displaying available settings and services. Ex-1005, §§ 3-3.3, FIG. 3; Ex-1003, ¶174. A POSITA would have expected success using gestures to display a list with a library of available applications and files, as in Vayda, to provide a more convenient and efficient gesture-based activation for users on a touch screen. Ex-1003, ¶174. It was well-known to display a list of available files and applications on a mobile handheld computer unit, like conventional Windows Start Menu program lists, file folders/directories, and "search" functions on conventional laptops and PDAs. Ex-1003, ¶174; *see* Ex-1009 (Tarpenning), 6:42-49, Table 1 (describing a library menu displaying a "list of the titles currently stored on the device" and "a list of additional programs available on the device"). Dutta (Ex-1011) confirms that opening lists of applications or searching a device for files was known to POSITAs, as shown by "APPLICATIONS," "FIND," and "MENU" buttons at the bottom of the touch

screen in Figure 3. Ex-1003, ¶174. Modifying Robertson's gesture-based user interface to activate a function to display a list with a library of available applications and files would have resulted in the ordinary operation of opening a list or menu of files and applications as was known. *Id.* Therefore, displaying a list or menu of available files and applications in Robertson's user interface merely combines known functions operating as expected. *Id.*

3. [Claim 7] "wherein the user interface is characterised in, that a selection of an application or file is done by gliding the object along said touch sensitive area so that a representation of a desired one of said application or file is highlighted, raising said object from said touch sensitive area, and then tapping on said touch sensitive area."

As discussed in claim 6 (Section V.B.2), Vayda discloses and renders obvious displaying a list with a library of available applications and files on the mobile handheld computer unit. Ex-1007, FIG. 14; Ex-1003, ¶¶176-182.

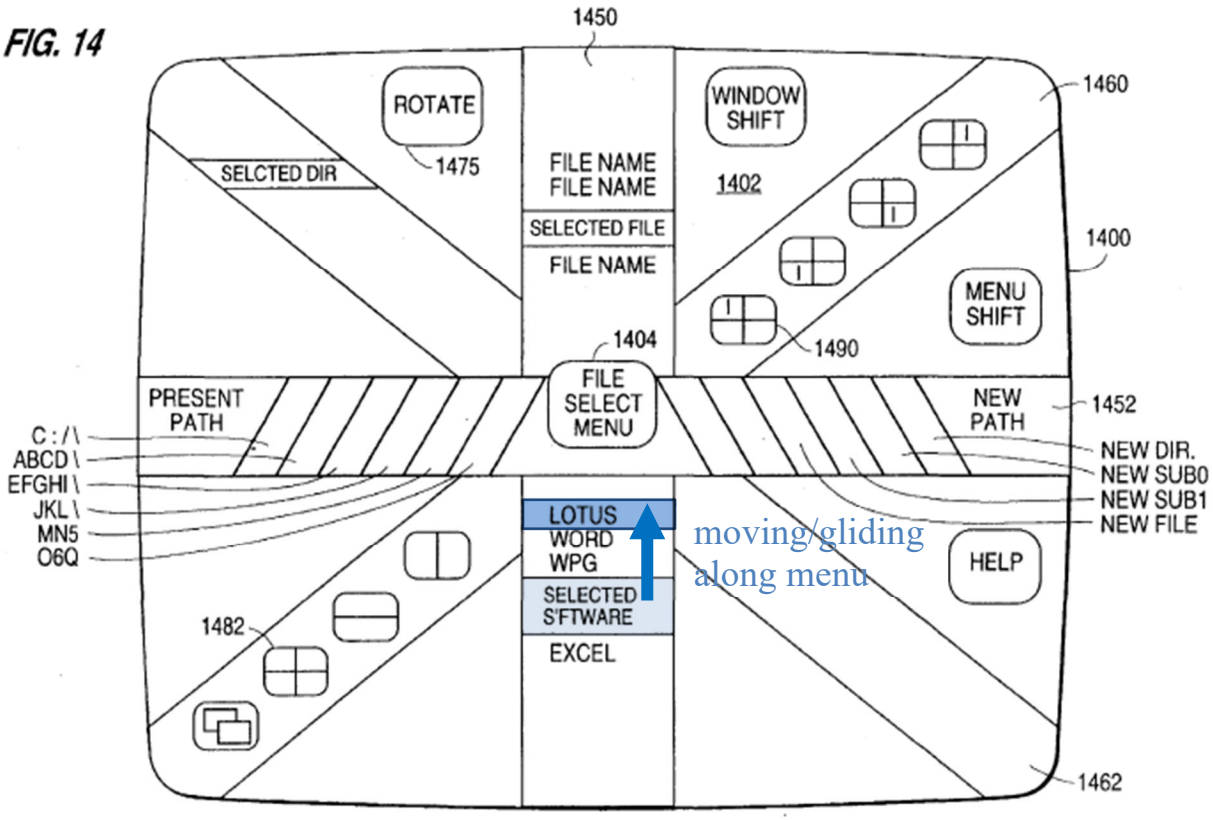
Vayda discloses menus showing a "scroll position" (e.g., "scroll position[] 1564" (Figure 15), highlighted "selected" item (Figure 14)) that "indicate[s] [highlights] the currently selected command along that command bar," as shown. Ex-1007, 15:35-56, FIGS. 14-15; Ex-1003, ¶177.



Ex-1007, FIG. 14 (annotated).

As explained in Section V.B.1, the user "scroll[s] through the commands on [the] command bar" "by movement of user input device 106 in either direction along the bar desired" by moving or gliding input device 106 (e.g., finger/stylus on a touch screen) along vertical command bar 1450, which moves the highlighted scroll position to "indicate the currently selected" menu item (i.e., position scrolled to by user), shown below. Ex-1007, 15:40-56, 4:41-51; Ex-1003, ¶178. Gliding a stylus or finger along vertical command bar 1450 will "snap" or "scroll through the commands on [the] command bar," which a POSITA would understand to mean

highlighting each of the files or applications shown in command bar 1450 as the stylus moves over them on the touch screen, as shown. Ex-1007, 15:40-56.

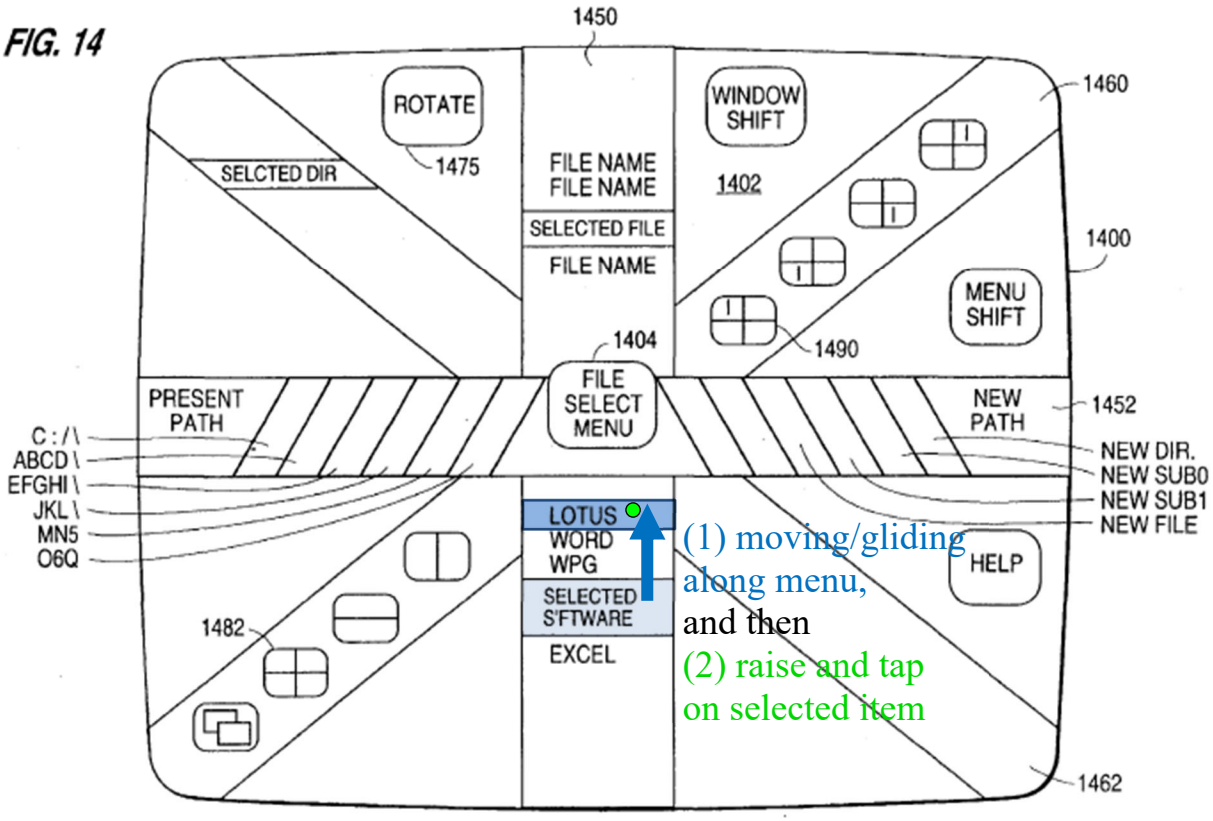


Ex-1007, FIG. 14 (annotated).

Robertson discloses tapping ("clicking," "pressing") to execute an action. Ex-1005, §§ 1.1-1.2, 2, 3.1; Ex-1003, ¶179.

Vayda also teaches selecting the highlighted file or application by then "clicking" the file or application. Ex-1007, 7:64-67 (executing command by first "mov[ing]" then "click[ing]" the user device), 8:15-17 (executing command by "snapping and [then] clicking" the user device), 15:7-10; Ex-1003, ¶180. This "confirmation click" is "a separate operation in addition to the snap" to execute the

selected menu item. Ex-1007, 8:42-49; Ex-1003, ¶180. As explained for claim 3 (Section V.A.5), a POSITA would understand a "tap" of a user's finger or stylus on a touch screen as a conventional "click" operation for touch screens. Ex-1003, ¶180. Thus, a POSITA would understand Vayda's "confirmation click" to disclose raising and then tapping the stylus back on the screen (e.g., a "tap" action). *Id.* This sequence for selecting (highlighting) by moving (gliding) the stylus or finger and then confirmation clicking (tapping) the selection is shown below. *Id.*



Ex-1007, FIG. 14 (annotated).

A POSITA would have found it obvious to include a confirmation tap, as in Robertson and Vayda, to execute a selected file or application after the scrolling

(gliding) operation highlights the item, as in Vayda. Ex-1003, ¶182. A POSITA would understand the confirmation tap to select the item as a natural pause to separate the gliding gesture from the subsequent selection tap. *Id.*

i. Motivation

A POSITA would have been motivated to include Vayda's scroll highlighting, followed by a confirmation tap to select an application or file, to avoid an unintentional selection when the user accidentally scrolls to an undesired location along the list of files and applications, or if the user pauses scrolling at an intermediate location in the menu. Ex-1003, ¶183. A POSITA would have been motivated to include this functionality in a menu or list activated by Robertson's gestures to allow an opportunity to confirm the desired file or application has been selected while navigating through the menu or list, especially with novice users or smaller devices. Ex-1003, ¶183; Ex-1007, 8:23-28. A POSITA would have been motivated to incorporate Vayda's gliding scroll operation to preserve screen space on smaller devices because no additional buttons or scroll bars are needed to scroll through the list and because scrolling using small buttons can be difficult. Ex-1003, ¶183; *see* Ex-1017 (Agulnick), 10:14-16 ("[I]n the case of scrolling, it is much easier to flick within a large window than to tap a relatively small button.").

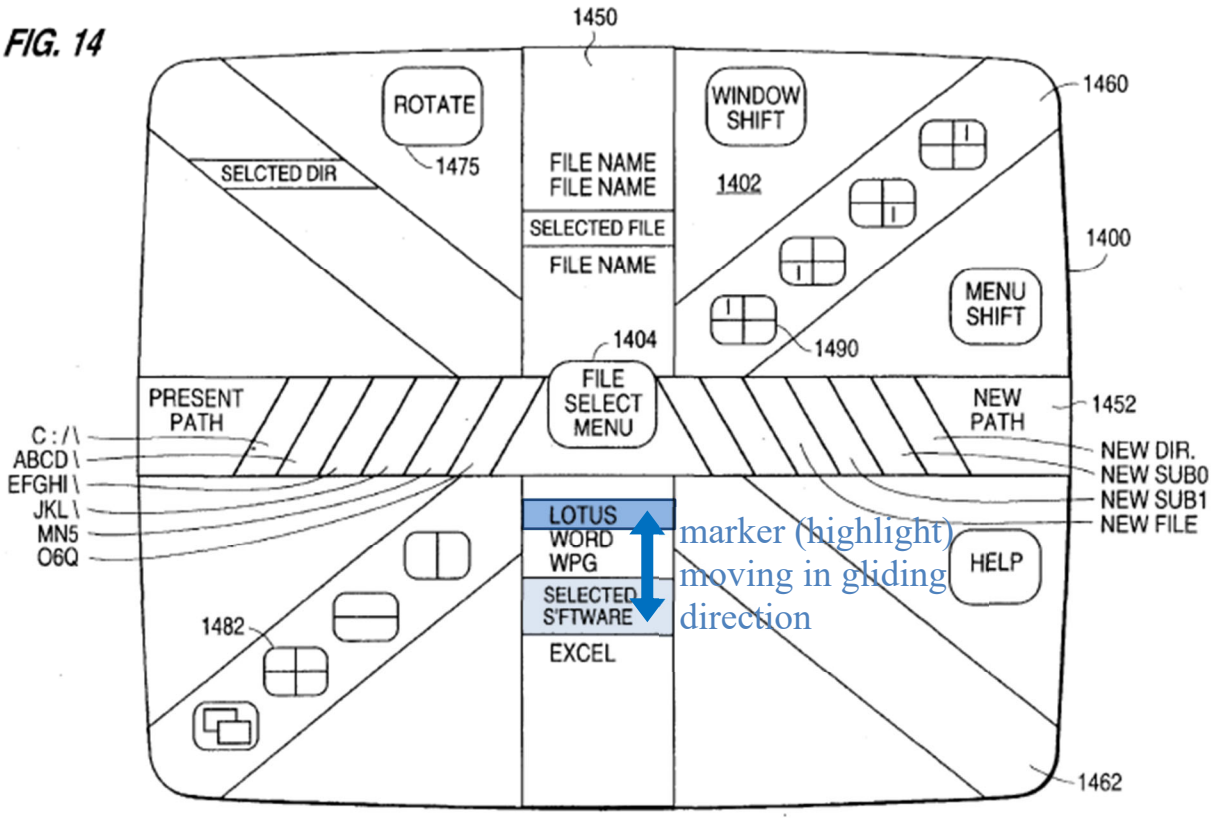
ii. Expected success

A POSITA would have expected success incorporating scrolling to select an application or file in a menu or list and then performing a confirmation tapping gesture because all of these actions—gliding, highlighting the glided-to item, and tapping to select the highlighted item—were conventional and well-known functions to a POSITA. Ex-1003, ¶184; Ex-1009 (Tarpenning), 7:44-51 (gliding a stylus through a list of items and highlighting the glided-to item by an icon next to it); Ex-1005, § 3.1 ("pen-based gestural input[s]); Ex-1007, 15:40-56, 4:41-51 (gliding to scroll and highlight menu items); Ex-1017 (Agulnick), 8:14-18 (a horizontal stroke is a "flick" gesture). Robertson's and Vayda's "click" gestures (tapping using a stylus or finger, *supra* § V.A.5 (claim 3)) were conventional for selecting an item from a list. Ex-1003, ¶184. Thus, incorporating Vayda's glide to highlight a menu item, then performing a confirmation click, as in Vayda and Robertson, in a displayed menu or list on a touch-screen interface combines known functions operating as expected. *Id.*

4. [Claim 9] "wherein the user interface is characterised in, that, one item in said list is highlighted by a moveable marking, and the user interface enables list navigation whereby gliding the object along the touch sensitive area in a direction towards the top of said list or towards the bottom of said list causes said marking to move in the same direction without scrolling the list."

Per Sections V.B.1 and V.B.3 (claim 7), Vayda teaches a scroll position (moveable marking) moving along a list of files and applications (vertical command bar 1450) towards the top of the list or the bottom of the list to highlight a "selected" application or file (item). Ex-1003, ¶¶186-189. Vayda also explains this functionality enables list navigation by "mov[ing]" (gliding) the object (e.g., stylus or finger) along the touch sensitive area of the list "***in either direction*** along the bar desired" towards the top or bottom of the list "to ***scroll through the commands*** on that command bar," which a POSITA would understand as moving the highlighted selection bar with the user's movement, as shown. Ex-1007, 15:40-56 (emphases added), 4:41-51; Ex-1003, ¶186. For example, when the user moves his or her finger or stylus (object) from the "SELECTED S[O]FTWARE" position up towards the listed "LOTUS" application, the highlighted scroll position moves up ("snap[s]") with the movement. Ex-1007, 15:35-56, FIG. 14; Ex-1003, ¶186. The reverse is also true; if the user glides or moves the stylus or finger back down towards the "EXCEL" application, the highlighted scroll position moves towards

the bottom of the list with the gliding movement. Ex-1007, 15:35-56; Ex-1003, ¶186.



Ex-1007, FIG. 14 (annotated).

Alternatively, the marker (highlight) may be moved up or down the list by flicking a finger or stylus multiple times towards the top or bottom of the list (to "snap" up or down the list, respectively), which causes the marker (highlight) to move in the desired direction up or down, as explained in Section V.B.3 (claim 7). Ex-1003, ¶187.

A POSITA would understand that this causes the scroll position to move along vertical command bar 1450 without scrolling the list. Ex-1003, ¶188. This

non-scrolling of the list is confirmed by Vayda's "[a]lternative[]" operation where the list scrolls through the scroll position (Ex-1007, 15:58-63), rather than moving the scroll position along the list. Ex-1003, ¶188.

Per Sections V.B.1 and V.B.3 (claim 7), Vayda renders obvious that one item in the list is highlighted by a moveable marking (scroll position), and the list is navigated by gliding the object (finger or stylus) along the touch sensitive area towards the top or bottom of the list, causing the marking to move in the same direction without scrolling the list. Ex-1003, ¶189.

C. [Ground 3] Claim 12 rendered obvious by Robertson, Maddalozzo, and Bedford-Roberts

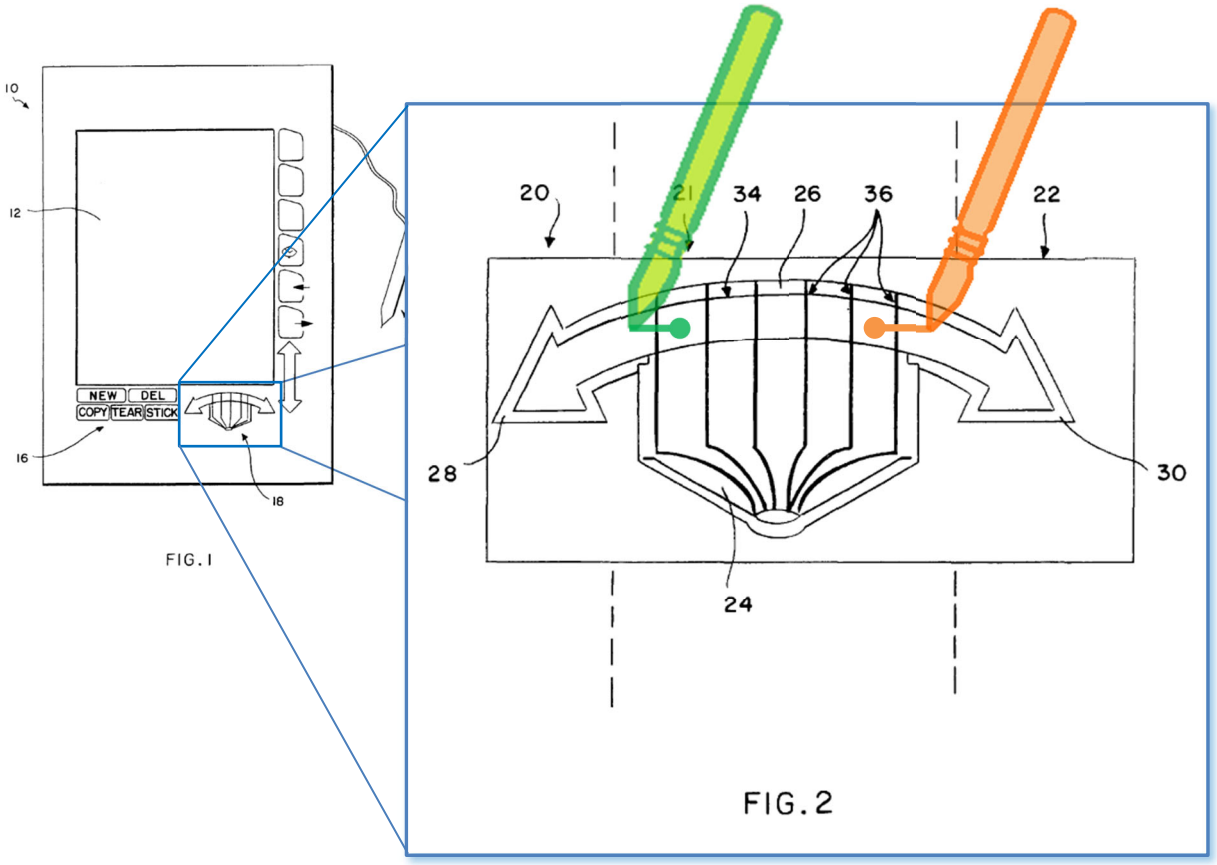
1. [Claim 12] "wherein the user interface is characterised in, that an active application, function, service or setting is advanced one step by gliding the object along the touch sensitive area from left to right, and that the active application, function, service or setting is closed or backed one step by gliding the object along the touch sensitive area from right to left."

Robertson discloses moving a finger or stylus from left to right ("flick right") or from right to left ("flick left") to activate commands. Ex-1005, § 3.1; Ex-1003, ¶¶192-195.

Bedford-Roberts teaches using a touch-sensitive screen "to make freehand input[s]," like gestures, using a pen or finger to control the device. Ex-1008, Abstract, 1:42-58, FIGS. 1-2. Bedford-Roberts teaches using left-to-right and right-

to-left gliding movements on a book icon to advance one page (step) and back up one page (step), respectively, in an active e-book application. Ex-1008, 2:50-55, 2:65-3:29, FIGS. 1-2; Ex-1003, ¶193.

As shown in annotated composite Figures 1 and 2, when a "user moves the pen 14 along the body 34 of the arrow, or indeed anywhere within the region 21, successive pages of stored information are displayed, depending on the direction of movement along the arrow body 34." Ex-1008, 2:50-53; Ex-1003, ¶194. For example, when the pen is touched down in the arrow body and passes over a page delimiter 36, "the next page in the direction of movement of the pen 14 is displayed." Ex-1008, 2:54-55; Ex-1003, ¶194. If the pen goes down over the book icon and is then moved (glided) from left-to-right across a delimiter 36 (right direction), the book advances to the next page (**orange**). And if the pen is moved from right-to-left across the delimiter (left direction), the book backs up to the previous page (**green**). Ex-1008, 2:50-55, 2:65-3:29; Ex-1003, ¶194. This is confirmed by the pseudocode in column 3, which checks whether the pen is "dragged" (glided) across the delimiter to turn a page. Ex-1008, 2:65-3:29; Ex-1003, ¶194.



Ex-1008, FIGS. 1-2 (annotated).

A POSITA would have found it obvious to use Bedford-Roberts's left-to-right and right-to-left gestures to advance and back up an active application, function, service, or setting by implementing one of Robertson's gesture-based buttons using the "flick right" and "flick left" gestures in an e-book application. Ex-1003, ¶195.

i. Motivation

A POSITA would have been motivated to use Bedford-Roberts's right-to-left and left-to-right gestures (akin to Robertson's "flick" gestures) to advance and

back up one step (e.g., page) in an active e-book application on touch-screen mobile devices. Ex-1003, ¶196. Motivation would come from Agulnick's explanation: "it is much easier to flick within a large window than to tap a relatively small button." Ex-1017, 10:14-16; Ex-1003, ¶196. A POSITA would also have been motivated to use such glide gestures to turn a page on the active application to create a more natural operation for users that is more convenient than tapping a button and prevents accidentally "double tapping" the button because the gesture is more natural. Ex-1003, ¶196.

ii. Expected success

A POSITA would have expected success using Robertson's and Bedford-Roberts's left-to-right and right-to-left glide gestures to advance and back up one page (step) in an active e-book application because such gestures were known to POSITAs, especially for e-book readers, e.g., buttons within the e-book application. Ex-1003, ¶197; Ex-1005, §§ 3.1-3.2, 1.1 (buttons may be part of applications); Ex-1008, 2:50-55; *see also* Ex-1017, 8:14-18, 10:14-16; Ex-1016, 13:42-47, 13:53-55. Assigning left-to-right and right-to-left gestures to advance and back up one step would result in expected operations behaving conventionally. Ex-1003, ¶197.

D. [Ground 4] Claims 1, 4-6, 13, and 15-17 are rendered obvious by Tarpenning

1. Tarpenning

Tarpenning discloses a handheld electronic reading device having a "touch-sensitive display and graphical user interface." Ex-1009, Abstract. A touch-sensitive display 34 (**red**) displays user interface elements having associated functions. Ex-1009, FIGS. 6-13. The user may interact with various "fixed icons" (displayed elements), like book menu key 84 (**blue**) or library menu key 86, by touching or pressing the icon using a stylus/finger to activate the corresponding functions, causing the user interface to display book menu 85 (**green**) or a library menu (not shown), respectively. Ex-1009, 6:9-14, 6:41-49, Table 1, FIG. 6.

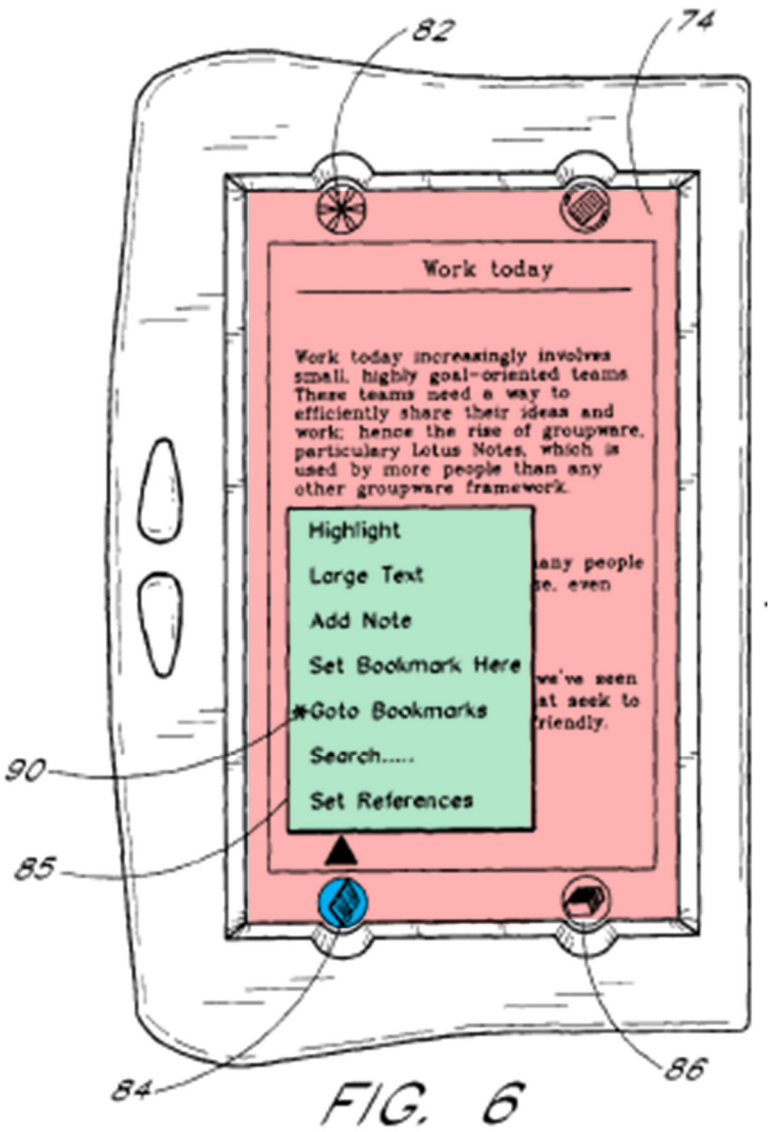
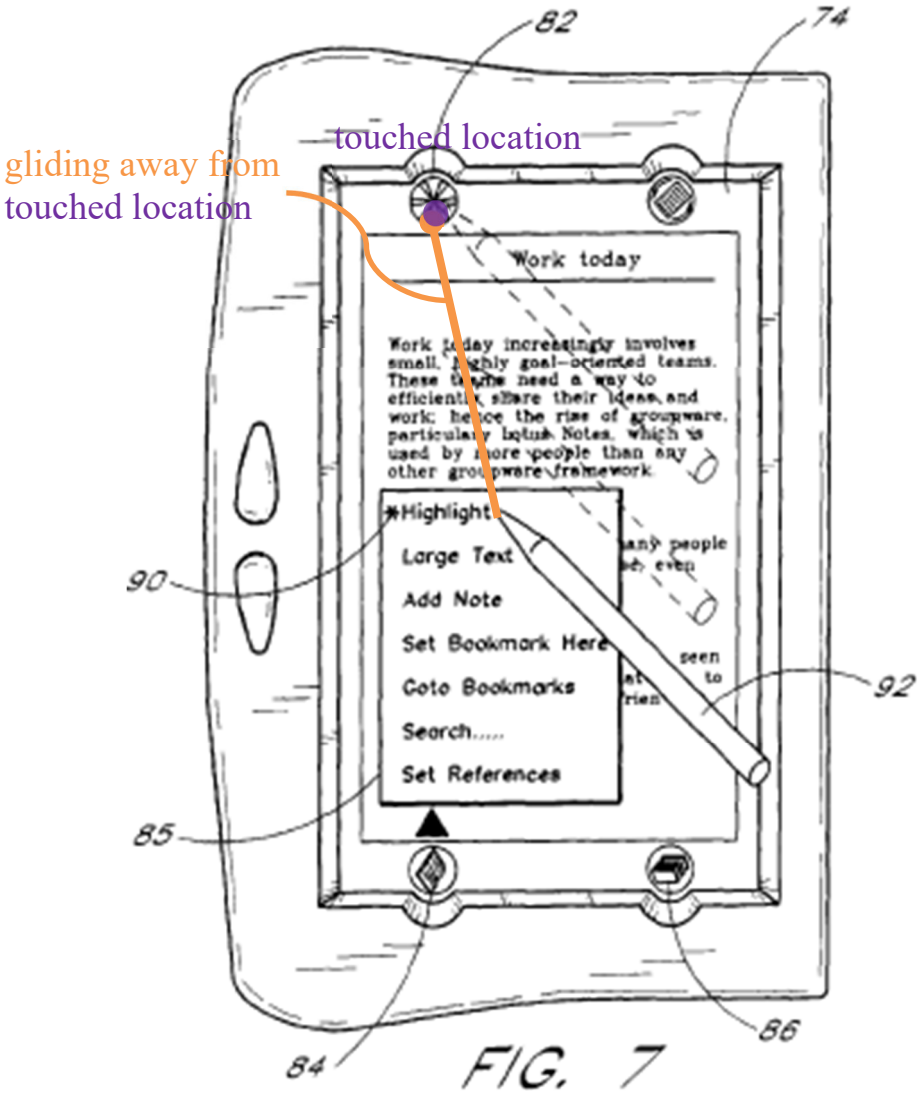


FIG. 6

Ex-1009, FIG. 6 (annotated).

Another way to activate a key's function is a touch-then-glide operation, for example, touching fixed-icon hotkey 82 at a location (**purple**) with an object (stylus 92 or user's finger), then dragging or gliding (**orange**) the object to activate the hotkey's "assignment" function, as shown. Ex-1009, 7:44-48, FIG. 7. After activating the "assignment" function, a new function for the hotkey is assigned

when the user lifts the object at the target to be assigned (e.g., "Highlight" in book menu 85). Ex-1009, 7:44-53, FIG. 7.



Ex-1009, FIG. 7 (annotated).

2. Claim 1

Tarpenning renders obvious claim 1.

a. Element [1Preamble]

Tarpenning's device 30 includes a portable reading device. Ex-1009, 3:44-48; Ex-1003, ¶¶204-207.

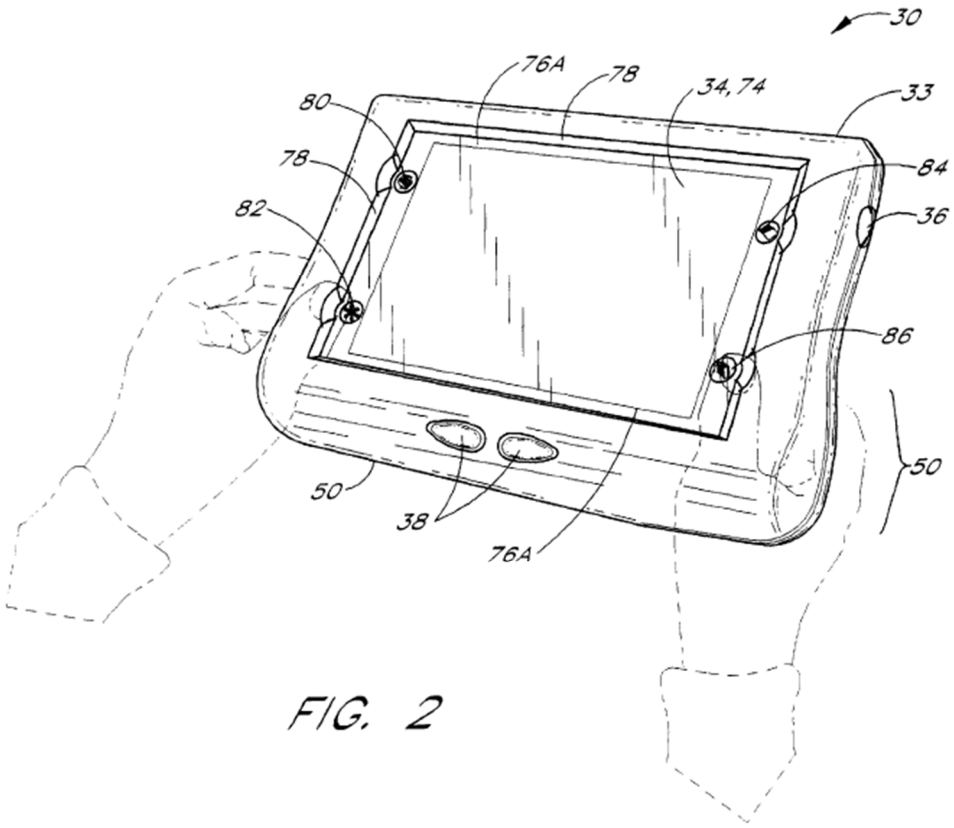
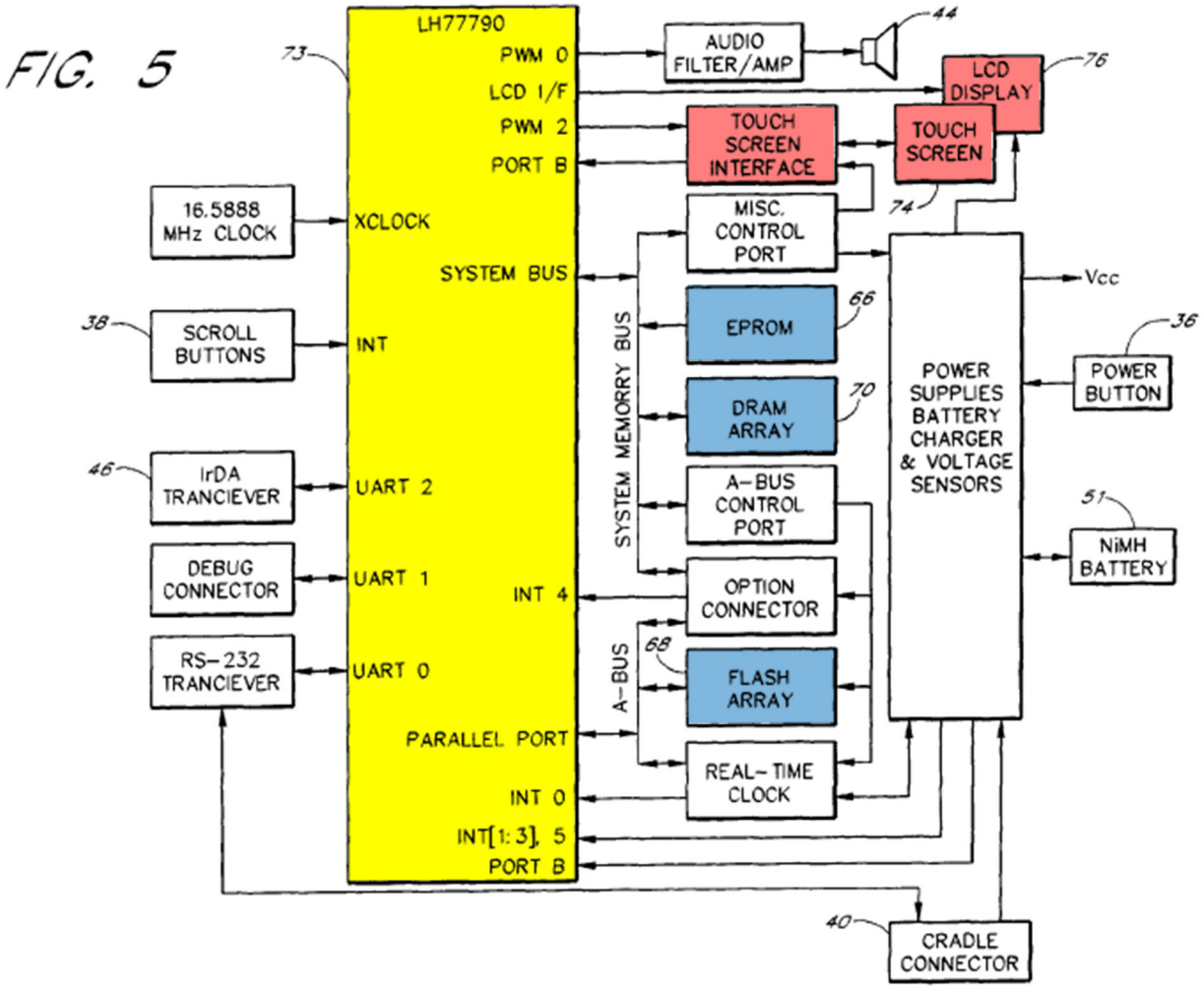


FIG. 2

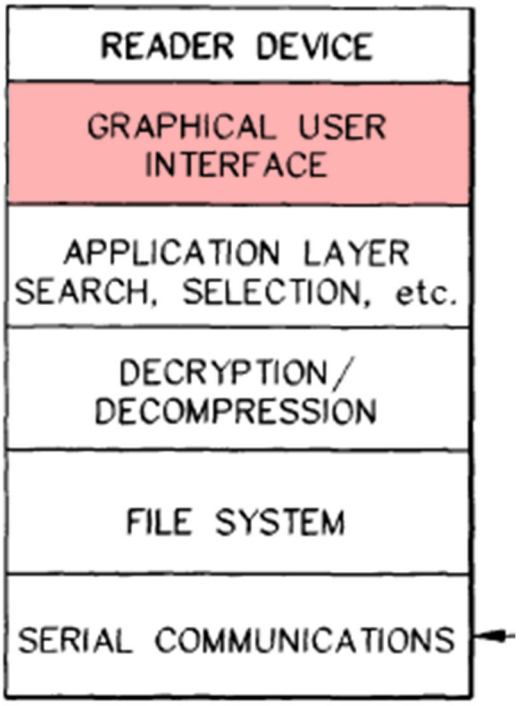
Ex-1009, FIG. 2.

Device 30 corresponds to a mobile handheld computer unit because it "is controlled by a Sharp LH77790 microprocessor 73" (yellow), which "accesses three types of memory" (collectively memory, blue) storing a computer program with computer program code, e.g., "system software, the user and device keys, titles and software applications," which, when read by device 30, allows the

computer to present the user interface. Ex-1009, 5:46-53, FIGS. 4-5; Ex-1003, ¶206. The user interface (**red**) uses touch-sensitive screen 34, as shown. Ex-1009, 3:44-48, 3:51-56, FIGS. 4-6; Ex-1003, ¶206.



Ex-1009, FIG. 5 (annotated).



Ex-1009, FIG. 4 (excerpted; annotated).

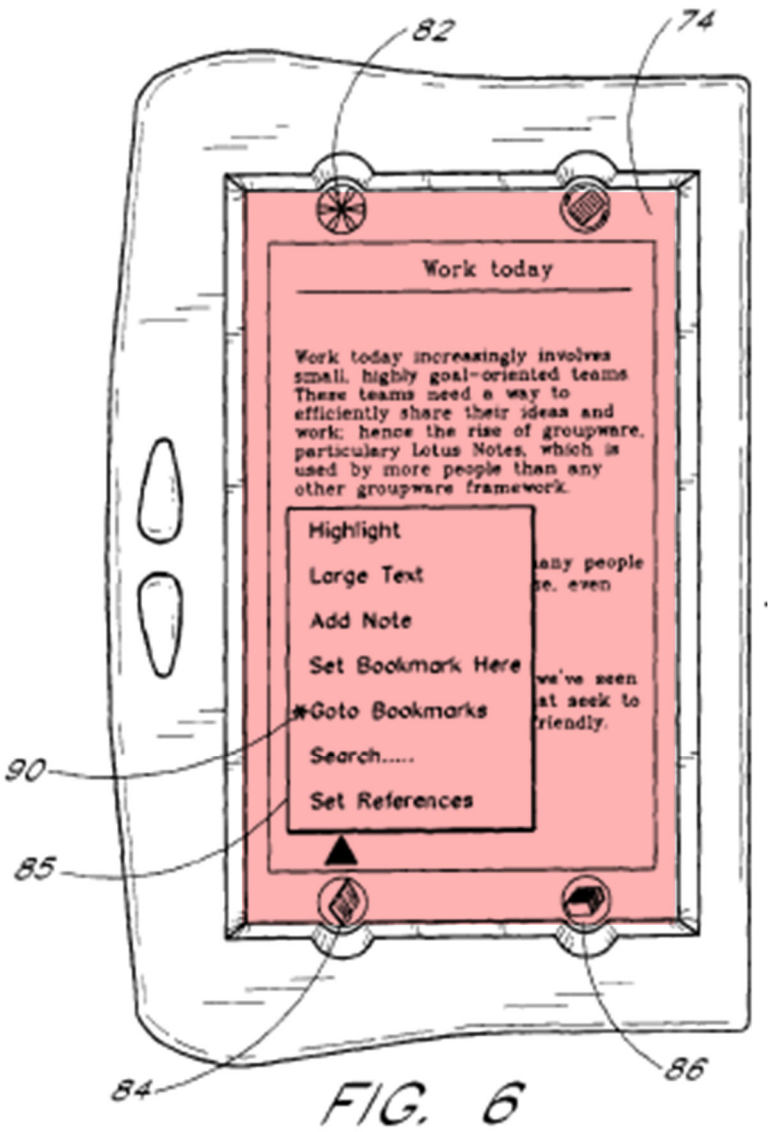


FIG. 6

Ex-1009, FIG. 6 (annotated).

b. Element [1a]

Device 30 includes touch-sensitive display 34 (touch sensitive area, **red**) including touch screen 74 having book menu key 84 and library menu key 86 (representations, **blue**), which are fixed icons displayed on touch screen 34, as shown. Ex-1009, 3:66-4:2, 6:9-16, FIG. 2; Ex-1003, ¶¶208-211.

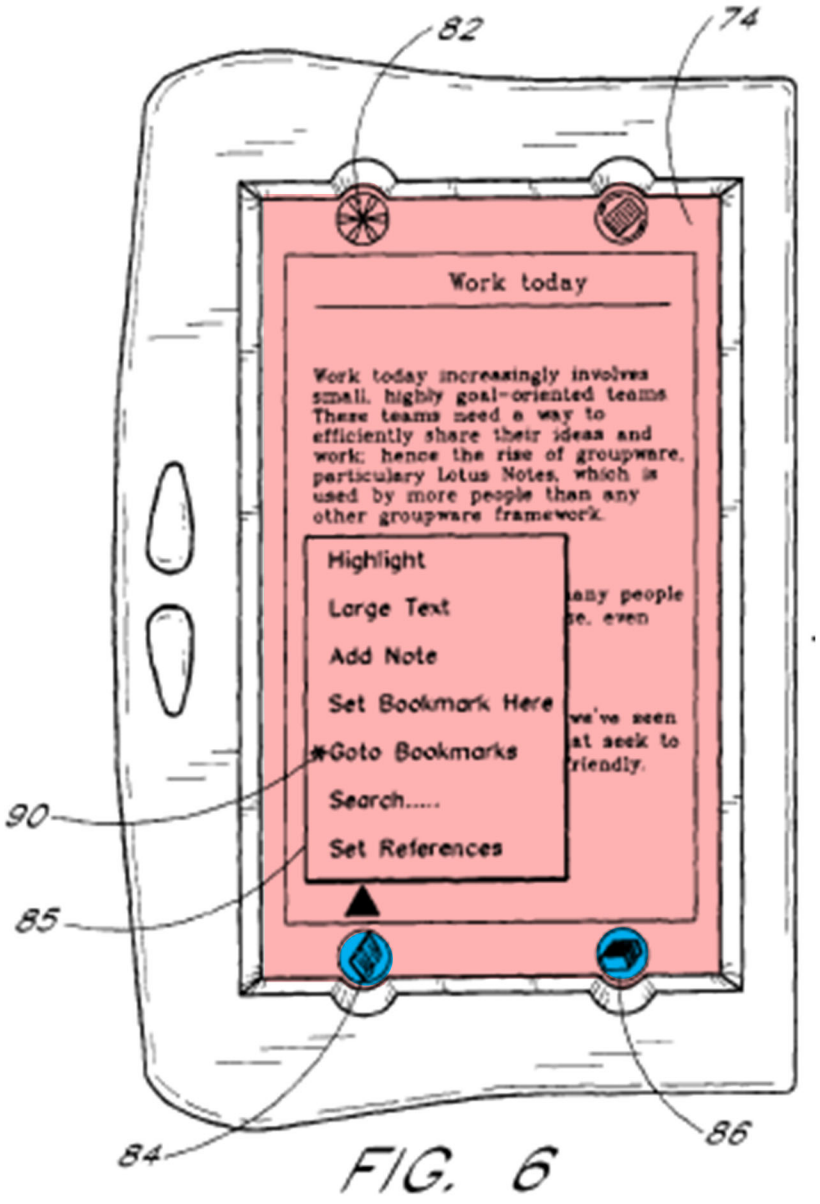

82
74
Work today
Work today increasingly involves small, highly goal-oriented teams These teams need a way to efficiently share their ideas and work; hence the rise of groupware, particulary Lotus Notes, which is used by more people than any other groupware framework.
Highlight
Large Text
Add Note
Set Bookmark Here
*Goto Bookmarks
Search.....
Set References
90
85
84
86

FIG. 6

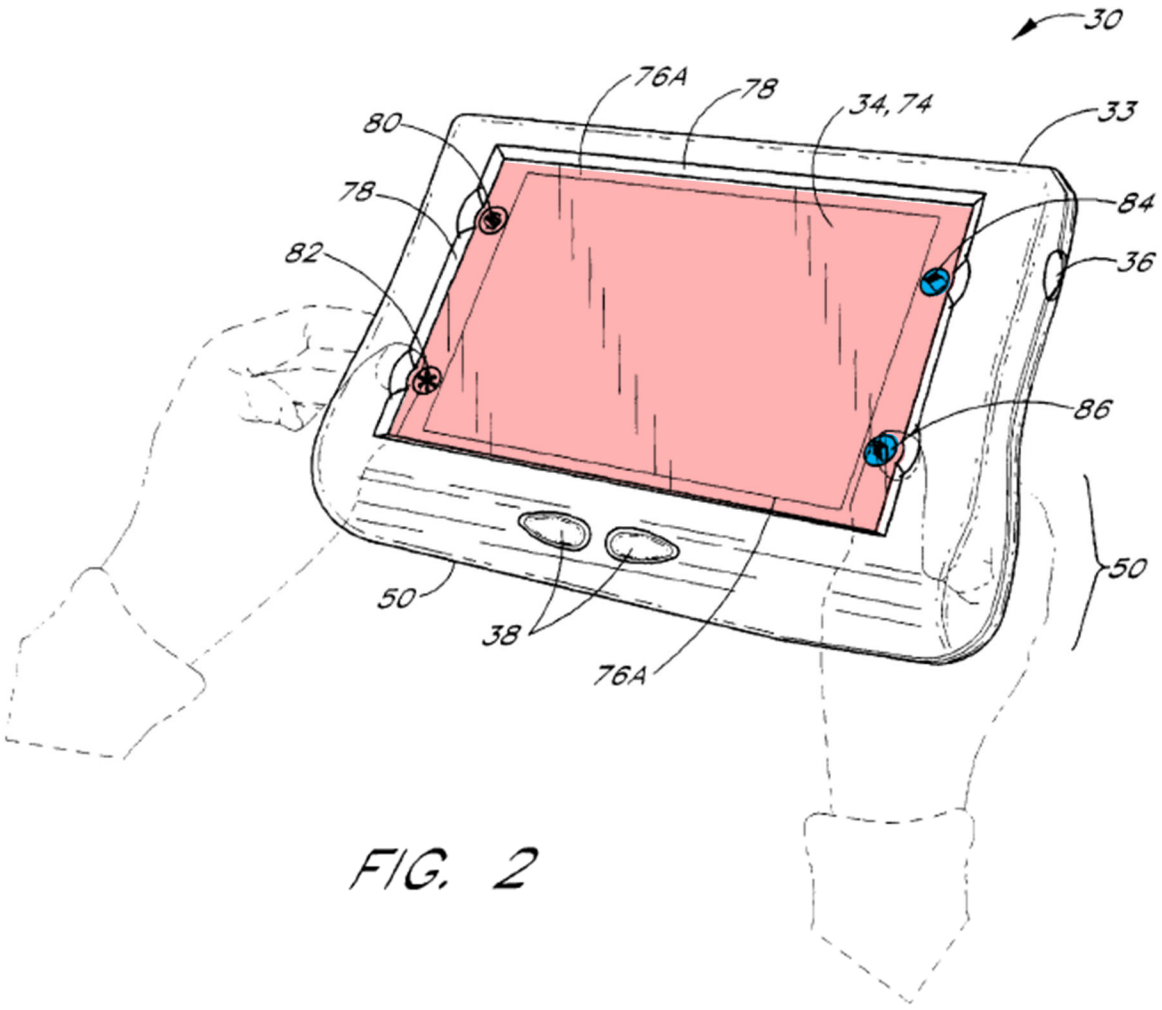


FIG. 2

Ex-1009, FIGS. 2, 6 (annotated).

Book menu key 84 and library menu key 86 are representations (**blue**) for their respective menu display functions that, when activated, present interactive menus to users. The book menu function (**green**) for book menu key 84, shown in annotated Figure 6, and the library menu function (not shown) for library menu key 86, are activated and displayed by the user pressing (e.g., touching) the key with a stylus or finger. Ex-1009, 6:9-14, 6:41-43; Ex-1003, ¶210.

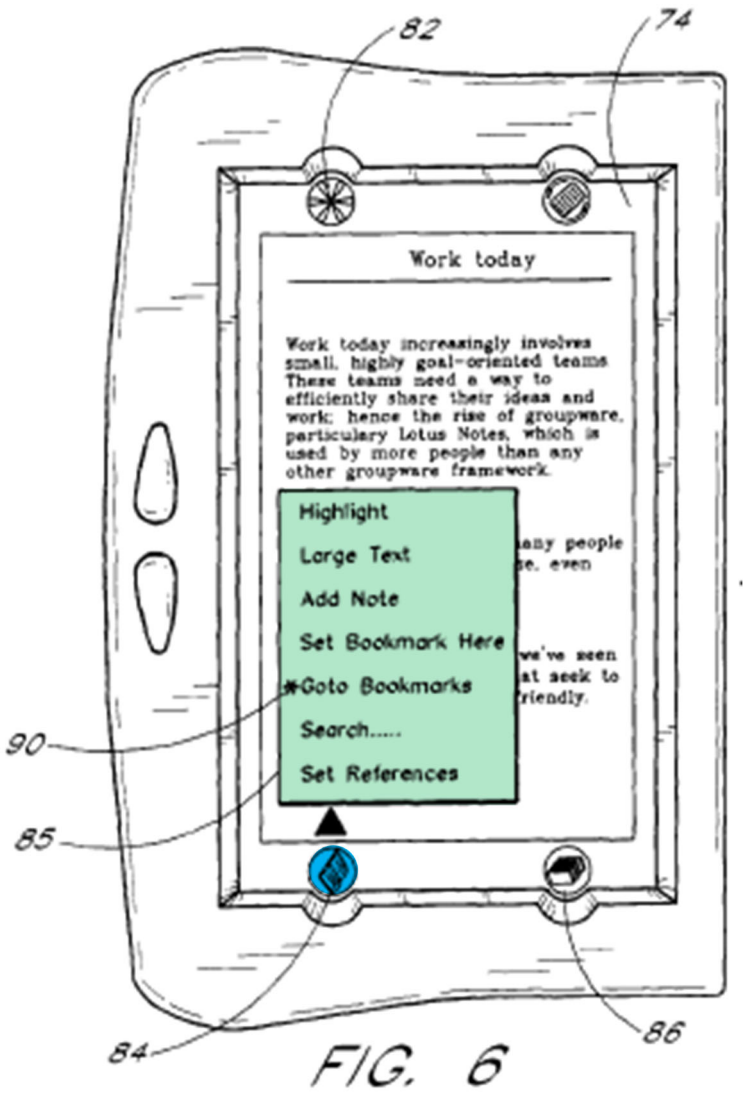


FIG. 6

Ex-1009, FIG. 6 (annotated).

A POSITA would understand book menu key 84 and library menu key 86 as "soft" keys activated by touching the key with an object (e.g., finger, stylus). Ex-1003, ¶211. A POSITA would understand that "pressing" or "touching" book menu key 84 (or library menu key 86) describes the user touching the key "with

the stylus 92 (or the user's finger)," as in Figures 2[6] and 7. Ex-1009, 7:44-48, 12:44-45; Ex-1003, ¶211; *also* Ex-1009, 6:9-14, 6:20-23, 6:35-37, 7:44-48, 12:44-45.

c. Element [1b]

Book menu key 84 and library menu key 86 (representations, **blue**) each consist of only one option for displaying book menu 85 (**green**) or the library menu (not shown), respectively, which is touching the key with a stylus or finger. *Supra* § V.A.3.b ([1a]); Ex-1009, 6:41-43, FIG. 6; Ex-1003, ¶212.

[6] The key touched by the user's right thumb in Figure 2 is mislabeled "86" instead of "84." *Compare* Ex-1009, FIG. 2, *with* Ex-1009, FIGS. 6-7, 9-10, 12-13, 17-18.

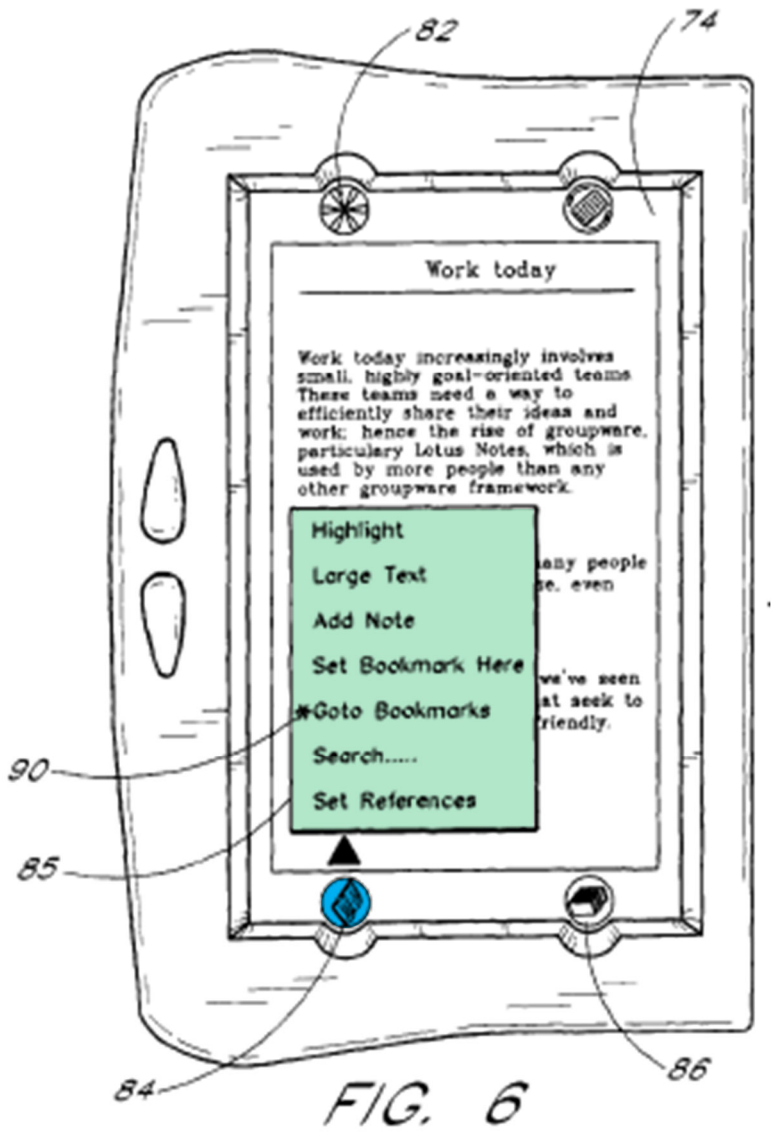


FIG. 6

Ex-1009, FIG. 6 (annotated).

d. Element [1c]

As explained for [1a] and [1b] (Sections V.D.2.b-V.D.2.c), the user presses (touches) the touch-sensitive screen (**red**) at book menu key 84 (or library menu key 86) (representations, **blue**) to activate the menu function (**green**) of the

corresponding key, as shown. Ex-1009, 6:9-14, 6:41-43, FIGS. 2, 6; Ex-1003, ¶¶214-220.

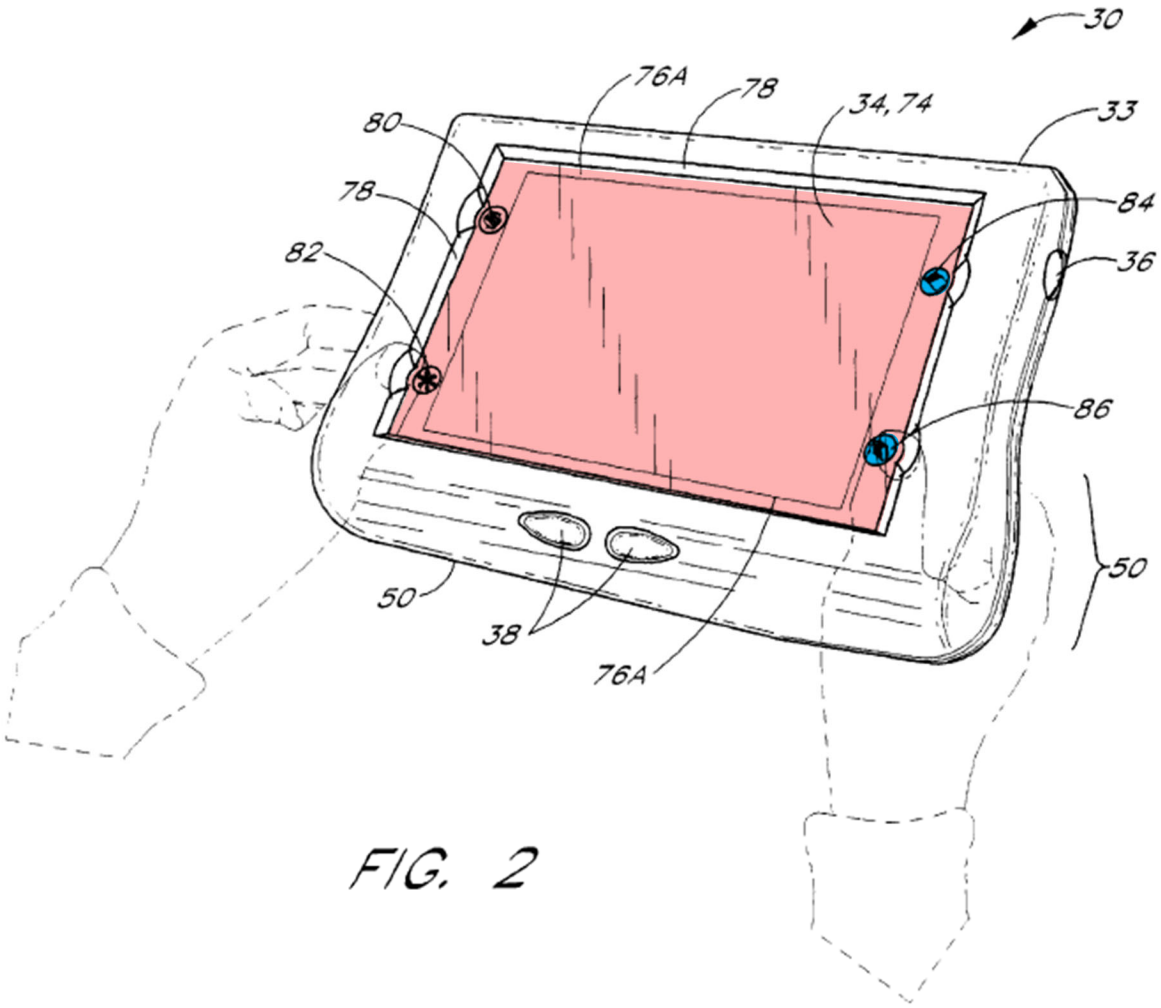


FIG. 2

Ex-1009, FIG. 2 (annotated; touching book menu key 84 (mislabeled "86")).

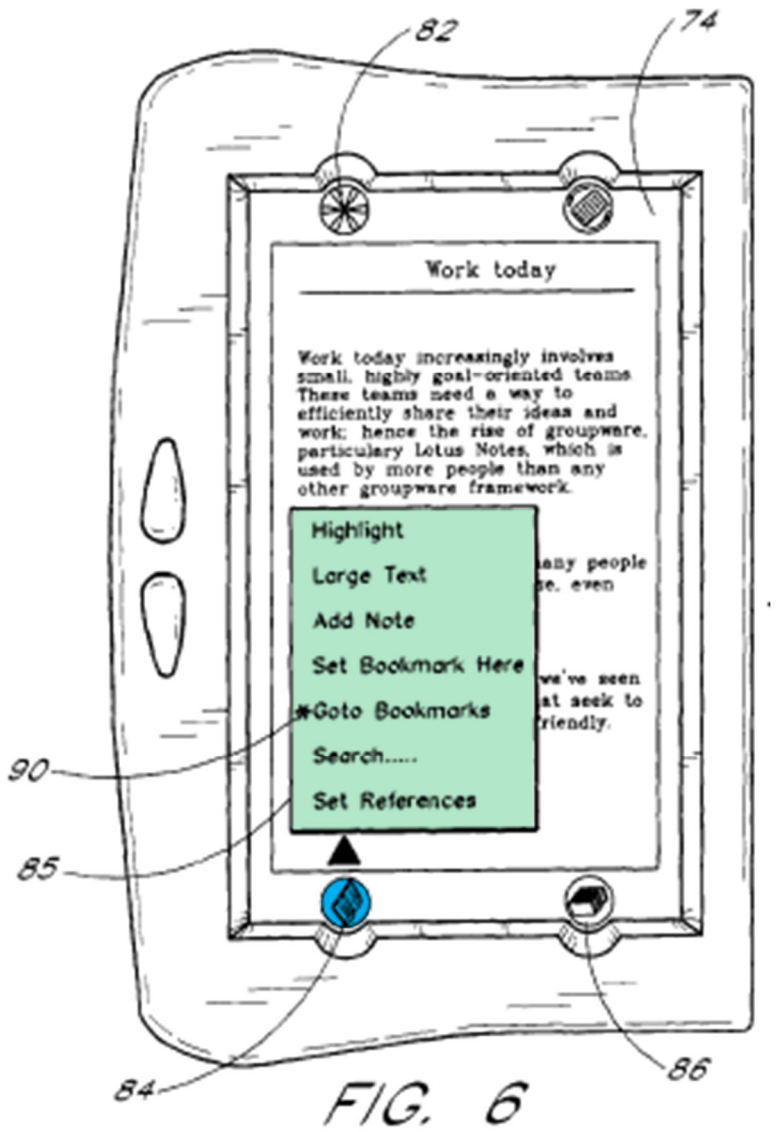
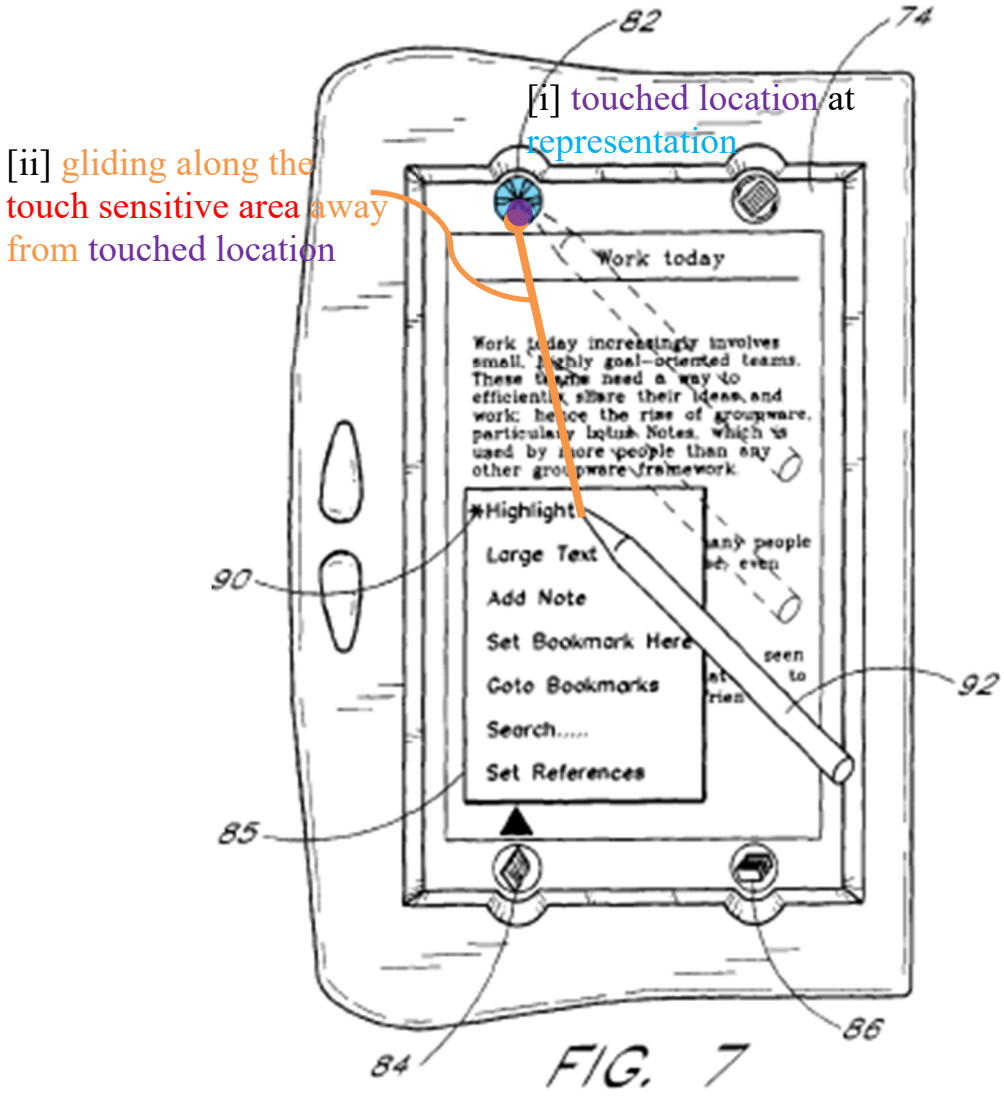


FIG. 6

Ex-1009, FIG. 6 (annotated).

Tarpenning also discloses using a multi-step touch-then-glide gesture operation to activate hotkey 82's "assignment" function, in which "the user … [i] touches the hotkey 82 with the stylus 92 (or the user's finger)," and then [ii] "drags [glides] the stylus to the target item." Ex-1009, 7:44-48, 6:35-40, FIG. 7; Ex-1003, ¶216. The assignment is completed when the user lifts the stylus at the

desired operation to be assigned. Ex-1009, 7:44-48, FIG. 7; Ex-1003, ¶216. Annotated Figure 7 shows this multi-step function activation, where the user touches (**purple**) the representation of hotkey 82 (**blue**) with a stylus or finger (object), and then drags or glides (**orange line**) the stylus or finger to activate the "assignment" function, which is completed when the user reaches the target item to be assigned (e.g., "Highlight" in book menu 85). Ex-1009, 6:35-40, 7:44-53, FIG. 7; Ex-1003, ¶216.



Ex-1009, FIG. 7 (annotated).

A POSITA would have found it obvious to replace Tarpenning's touch operation to activate the menu display functions of book menu key 84 and library menu 86 with the disclosed multi-step touch-then-glide activation gesture. Ex-1003, ¶217.

A POSITA would have found it obvious to use Tarpenning's touch-then-glide operation to [i] touch book menu key 84 (representation, **blue**) at the location (**purple**) of the menu key using a stylus or finger, then [ii] glide the stylus or finger up in the direction of the arrow indicated in Figure 6 (**orange line**) to activate Tarpenning's function of displaying book menu 85 (**green**), as shown in annotated Figure 6. Ex-1003, ¶218.

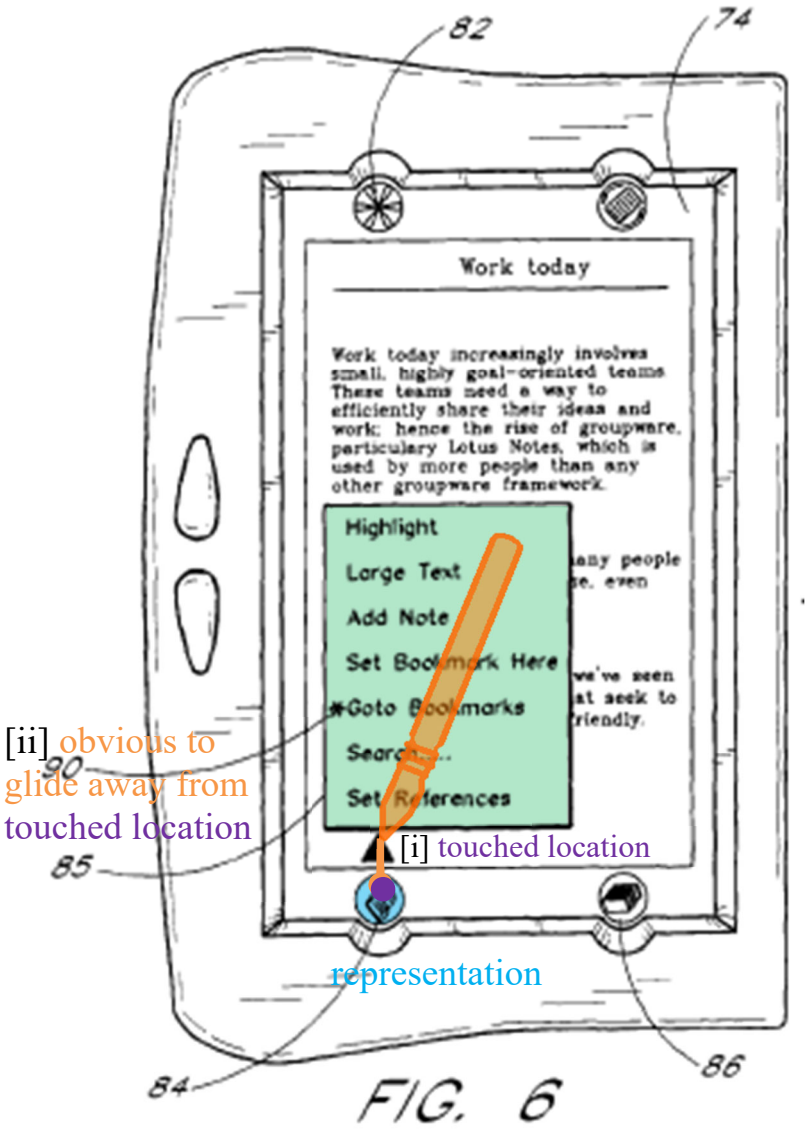


FIG. 6

Ex-1009, FIG. 6 (annotated).

A POSITA would have found this multi-step activation operation obvious to prevent accidental activation of the menu functions resulting from accidentally touching the icon. Ex-1003, ¶219.[7]

i. Motivation

A POSITA would have been motivated to substitute a touch operation on book menu key 84 and library menu key 86 with Tarpenning's disclosed touch-then-glide gesture to activate the function of displaying book menu 85 and the library menu, respectively, to prevent users from accidentally opening the menu when they inadvertently touch the screen with their finger or heel of their hand, which would lead to frustration. Ex-1003, ¶221.

A POSITA would have also been motivated to implement Tarpenning's touch-then-glide activation to activate the menu display functions of menu keys 84 and 86 to allow users to more accurately open sub-menus by gliding up to the desired sub-menu location without lifting the stylus or finger off the screen, which results in faster, more efficient operations for a user. Ex-1003, ¶222; *see* Ex-1017

[7] The '879 patent does not ascribe any unexpected or unique operation or particular benefit of activating a function using a touch-then-glide gesture instead of a touch alone. *See* Ex-1001; Ex-1003, ¶220.

(Agulnick), 10:14-16 (for scrolling, "it is much easier to flick within a large window than to tap a relatively small button").

The touch-then-glide gesture was known to POSITAs for mobile handheld computer devices, like touch-screen laptops, PDAs, etc., and thus represented a known design option for POSITAs. Ex-1003, ¶223; *supra* §§ V.A.3.d ([1c]) (describing Robertson's touch-then-glide "flick," "check," and "insert" gestures), V.B.1-V.B.4 (Vayda), V.C.1 (Bedford-Roberts).

ii. Expected success

A POSITA would have expected success substituting Tarpenning's touch-then-glide gesture to activate the menu display function of book menu key 84 or library menu key 86. Ex-1003, ¶224. Tarpenning's mobile handheld computer unit already uses both touch and touch-then-glide operations to activate key functions. Ex-1009, 6:41-43, 7:39-48. These types of operations were well-known before the filing date of the '879 patent. Ex-1005, §§ 3-3.2 (describing "flick," "check," and "insert" touch-then-glide gestures to activate functions); Ex-1007, 6:42-45 (describing activating "EDIT" command as either "click" or "snap" gesture); Ex-1008, 2:50-55, 2:65-3:29, FIGS. 1-2 (touch-then-drag to turn page(s)); Ex-1003, ¶224. Replacing the touch operation with a touch-then-glide gesture to activate the menu display functions of book menu key 84 and library menu 86

would result in the ordinary operation of displaying a menu in response to a known activation gesture. Ex-1003, ¶224.

e. Element [1d]

Tarpenning discloses [1d].

Book menu key 84 and library menu key 86 (representations of respective menu display functions) are "fixed icons" and not relocated or duplicated during activation. Ex-1009, 6:9-14; Ex-1003, ¶¶225-226.

It would have been obvious to use Tarpenning's touch-then-glide to activate the menu display functions of menu keys 84 and 86. *Supra* § V.A.3.d ([1c]); Ex-1003, ¶226. A POSITA would have recognized that the representation of the menu keys (fixed icons) would not be relocated or duplicated during the glide because the menus open next to the keys, as in Figure 6 where the representation is not relocated or duplicated. Ex-1009, FIG. 6; Ex-1003, ¶226.

3. Claim 4

Tarpenning renders obvious the menu display function, when activated, causing the user interface to display a keyboard and text field. Ex-1003, ¶¶228-234.

As explained in [1c] (Section V.D.2.d), the activated functions display book menu 85 or the library menu. Ex-1003, ¶229. Book menu 85 can also display the "Search/Lookup" sub-menu item (yellow), where the user can enter a "text string"

for searching the displayed e-book "using a pop-up keyboard." Ex-1009, Table 1, FIG. 6; Ex-1003, ¶229.

TABLE 1-continued

Menu Item	Description
Book Menu/ Search	Displays a sub-menu of search options (see below).
Book Menu/ Search/ Lookup	Searches the displayed title for a text string. The string can be selected before invoking this function, or can be entered using a pop-up keyboard.
Book Menu/ Search/ "other tags"	Takes the user to the publisher-defined tags. These tags are incorporated into the Search menu automatically.
Book Menu/ Set References	Allows the user to designate one or more reference titles to be used in place of a default reference title.
Library Menu/ Bookshelf	Displays of list of the titles currently stored on the device. From this list the user can delete titles or select a title for reading.
Library Menu/ Desk Supplies	Displays a list of additional programs available on the device.
Library Menu/ Set Citation Destinations	Allows user to create lists of destinations (email addresses, fax numbers, etc.) for routing selected passages. The destination lists can alternatively be generated on the PC and transferred to the device.
Library Menu/ "titles"	The library menu always displays the four most recently used literary titles. Touching a title will cause the title to be opened.

Ex-1009, Table 1 (excerpted; annotated).

A POSITA would understand entering a "text string" using a "pop-up keyboard" would display a "text field" with the keyboard to show the text being typed so the user can confirm correctness and make changes Ex-1003, ¶230; Ex-1009, Table 1. Using a text field to input and edit a text string with an on-

screen keyboard was known to POSITAs. Ex-1003, ¶230. For example, Dutta (Ex-1011), Figure 4, shows a text field and on-screen keyboard on a handheld mobile computing device (e.g., Palm device), which confirms the known operation for mobile touch-screen devices. Ex-1003, ¶230; *also* Ex-1006 (Maddalozzo), 6:52-55, 7:7-11, FIG. 6.

Tarpenning discloses "Add Note" (**yellow**) with book menu 85, which a POSITA would understand, when activated, allows the user to add a note to a displayed text or page. Ex-1009, FIG. 6, Table 1; Ex-1003, ¶231.

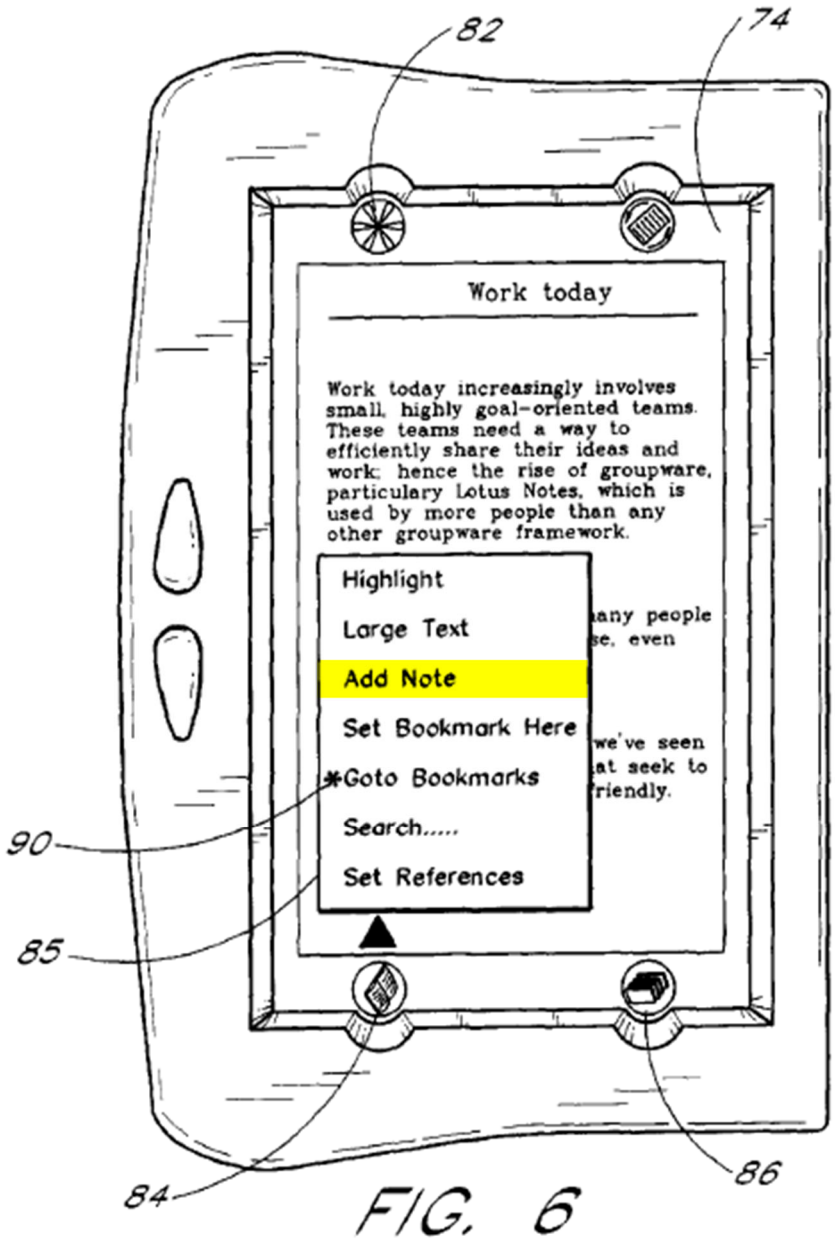


FIG. 6

Ex-1009, FIG. 6 (annotated).

TABLE 1

Menu Item	Description
Book Menu/ Highlight	Toggles mode which allows user to touch words, lines and paragraphs to permanently highlight them.
Book Menu/ Large Text	Toggles page display between normal, and large fonts.
Book Menu/ Add Note	Causes a user note to be attached to the selected text, or if no text is selected, to the current page.
Book Menu/ Set Bookmark Here	Sets a bookmark to the selected text, or if no text is selected, to the current page. The first few words of the selection are used as the default identifier of the bookmark.
Book Menu/ Goto Bookmarks	Displays a list of the bookmarks for the title, including both bookmarks defined by the publisher (such as "index" and "table of contents") and bookmarks defined by the user. From this list, the user can either goto or delete a displayed bookmark.

Ex-1009, Table 1 (excerpt; annotated).

A POSITA would understand the "Add Note" operation being similar to "Add Annotation." Ex-1009, 2:24-25, 10:32-36; Ex-1003, ¶232. "If the user then selects the 'Add Annotation' item, a pop-up keyboard (not shown) and editing screen are displayed for allowing the user to type in an annotation," which a POSITA would also apply to "Add Note." Ex-1009, 10:33-36, 2:24-25 ("[T]he user can type-in an optional annotation using a pop-up keyboard."); Ex-1003, ¶232.

A POSITA would understand displaying an on-screen "pop-up keyboard" and "editing screen" for the "Search/Lookup" operation and to add a note/annotation would present a text field for the user to input and edit the note text. A POSITA would have found it obvious to present the text field to input and edit the text with the "pop-up" keyboard so the user can see the note being input and edit the note to ensure correctness, as was conventional. Ex-1003, ¶233.

A POSITA would have found it obvious to display a text field and a "pop-up keyboard" in the user interface (e.g., next to or above book menu 85) so the user can input and edit text in the text field with the book menu for searching the title or adding a note. Ex-1003, ¶234.

i. Motivation

The "Search/Lookup" and "Add Note" menu items use a displayed keyboard and text field for receiving text input. Ex-1003, ¶235. A POSITA would have been motivated to display the keyboard and text field alongside book menu 85 so the user can access the text entry operation as part of the menu or sub-menu to simplify menu selection by keeping features together when presented. Ex-1009, Table 1; Ex-1003, ¶235. A POSITA also would have been motivated to simplify using the "Search/Lookup" and "Add Note" functions requiring a keyboard and text field by presenting them alongside the corresponding menu items in the user interface. Ex-1003, ¶235.

ii. Expected success

A POSITA would have expected success displaying an on-screen keyboard and text field when the book menu is activated, like when using "Search/Lookup" or "Add Note." Ex-1003, ¶236. Tarpenning teaches displaying an on-screen or "pop-up" keyboard and text field when the user activates the "Search" and "Add Note" functions. Ex-1009, Table 1; Ex-1003, ¶236. Displaying a keyboard and text field to input and edit text when the menu is activated, like for "Search" and "Add Note," was known to POSITAs and would result in the expected operation of displaying these elements in the user interface. Ex-1003, ¶236.

4. Claim 5

As explained for claim 4 (Section V.D.3), a POSITA would understand or have found obvious using Tarpenning's text field for the user to input and edit the text of a note or annotation through the displayed "pop-up keyboard" because Tarpenning's device 30 does not include a physical keyboard for the user to type on. Ex-1003, ¶¶238-239.

5. Claim 6

As explained for [1c] (Section V.D.2.d), Tarpenning renders obvious the library menu display function. Tarpenning also renders obvious the function, when activated, causing the user interface to display a list with a library of available applications and files. Ex-1003, ¶¶241-244.

Selecting "Bookshelf" (**yellow**) in the activated library menu displays a "list of the titles [files] currently stored on the device [30]" and "the user can … select a title for reading," e.g., in a sub-menu or dialog box alongside the library menu. Ex-1009, 5:51-53, Table 1; *supra* § V.D.2.d ([1c]). Furthermore, selecting "Desk Supplies" (**yellow**) in the activated library menu displays "a list of additional programs [applications] available on the device," e.g., in a sub-menu or dialog box alongside the library menu. Ex-1009, 6:42-49, Table 1; Ex-1003, ¶242. If the list size exceeds the sub-menu or dialog size, up/down arrows were conventional to scroll or page through the list. Ex-1003, ¶244.

TABLE 1-continued

Menu Item	Description
Book Menu/ Search	Displays a sub-menu of search options (see below).
Book Menu/ Search/ Lookup	Searches the displayed title for a text string. The string can be selected before invoking this function, or can be entered using a pop-up keyboard.
Book Menu/ Search/ "other tags"	Takes the user to the publisher-defined tags. These tags are incorporated into the Search menu automatically.
Book Menu/ Set References	Allows the user to designate one or more reference titles to be used in place of a default reference title.
Library Menu/ Bookshelf	Displays of list of the titles currently stored on the device. From this list the user can delete titles or select a title for reading.
Library Menu/ Desk Supplies	Displays a list of additional programs available on the device.
Library Menu/ Set Citation Destinations	Allows user to create lists of destinations (email addresses, fax numbers, etc.) for routing selected passages. The destination lists can alternatively be generated on the PC and transferred to the device.
Library Menu/ "titles"	The library menu always displays the four most recently used literary titles. Touching a title will cause the title to be opened.

Ex-1009, Table 1 (excerpt; annotated).

Tarpenning's library menu display function, when activated, thus causes the user interface to display a list with a library of available applications (programs,

Desk Supplies) and files (titles, Bookshelf) as lists of the library menu.[8] A POSITA would understand a user can switch between displaying the Bookshelf files or Desk Supply applications by selecting the other menu item. Ex-1003, ¶243.

Alternatively, it would have been obvious to a POSITA to display a list with a library of "titles currently stored on the device" (available files) and "programs available on the device" (available applications) when the library menu is activated using section dividers within the library menu and, if the list size exceeds display size, page up/down arrows to scroll the menu, which saves the user time because the user can see available titles and programs without choosing from different sub-menus. Ex-1003, ¶244.

i. Motivation

A POSITA would have been motivated to display a list with a library of book "titles" (available files) and "programs" (available applications) in list sub-menus or dialogs of the "Bookshelf" and "Desk Supplies" selections while the library menu is activated to allow the user to search and choose from either menu if they cannot find the item they are looking for in one of the lists, which saves the

[8] Dependent claim 8, which recites that "at any given time said list presents ***only files*** or ***only applications***…," confirms that the files and applications may be displayed separately. Ex-1001, 7:18-23 (emphases added).

user time and effort by not having to reopen the library menu to search the other list. Ex-1009, 6:42-49, Table 1; Ex-1003, ¶245.

Alternatively, a POSITA would have been motivated to present the titles (files) and programs (applications) as a single list, with customary section dividers to aid visual organization, rather than separate lists, to allow the users to view all of the files and applications in one place to prevent users from having to search multiple lists. Ex-1003, ¶246.

ii. Expected success

A POSITA would have expected success displaying a list with a library of available applications and files as a sub-menu, dialog box, or divided section of Tarpenning's library menu. Ex-1003, ¶247. Programs (applications) and title (files) are displayed when the "Desk Supplies" and "Bookshelf" functions are activated. Ex-1009, Table 1; Ex-1003, ¶247. Sub-menu lists, like the "Search" option, and dialog boxes are also known. Ex-1009, Table 1; Ex-1003, ¶247. Displaying the list of files and programs as a sub-menu list or dialog box would have applied only well-known functions of getting and populating a list option, which has been a conventional tool for POSITAs. Ex-1003, ¶247. A POSITA would recognize this would present a sub-menu, dialog box, or divided section list of the activated library menu resulting in the ordinary, expected operation. Ex-1003, ¶247.

6. Claim 13

Tarpenning discloses book menu key 84 and library menu key 86 (representations, **blue**) located at the bottom of touch-sensitive display 34 (**red**), as shown. Ex-1009, FIG. 6; Ex-1003, ¶249; *supra* § V.D.1 (Tarpenning).

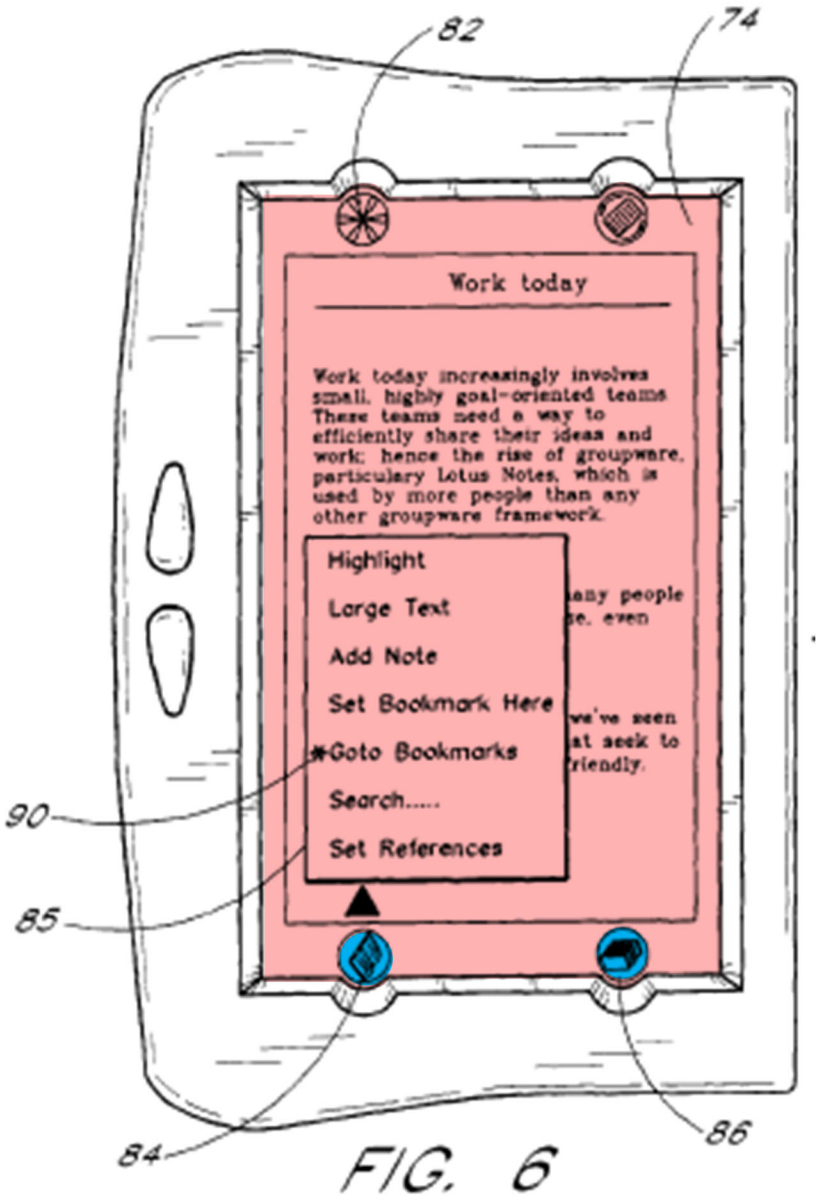


FIG. 6

Ex-1009, FIG. 6 (annotated); *also* FIGS. 7, 12, 13.

7. Claim 15

Tarpenning teaches device 30 and its programs displayed by the user interface are adapted to function as a shell upon an operating system, like "a proprietary operating system developed by NuvoMedia Inc." and "Windows CE™ operating system." Ex-1009, 5:46-53, 5:59-63; Ex-1003, ¶¶251-252.

8. Claim 16

Tarpenning's Figure 2 shows menu keys 84 and 86[9] (representations, **blue**) are the size of the user's thumb (**orange**) (finger-sized) and pressable by the user's finger. Ex-1009, FIG. 2, 6:41-43, 7:44-48; Ex-1003, ¶¶254-255.

[9] Book menu key 84 and library menu key 86 are reversed in Figure 2, but both are finger-sized. *Supra* § V.D.2.b ([1a]).

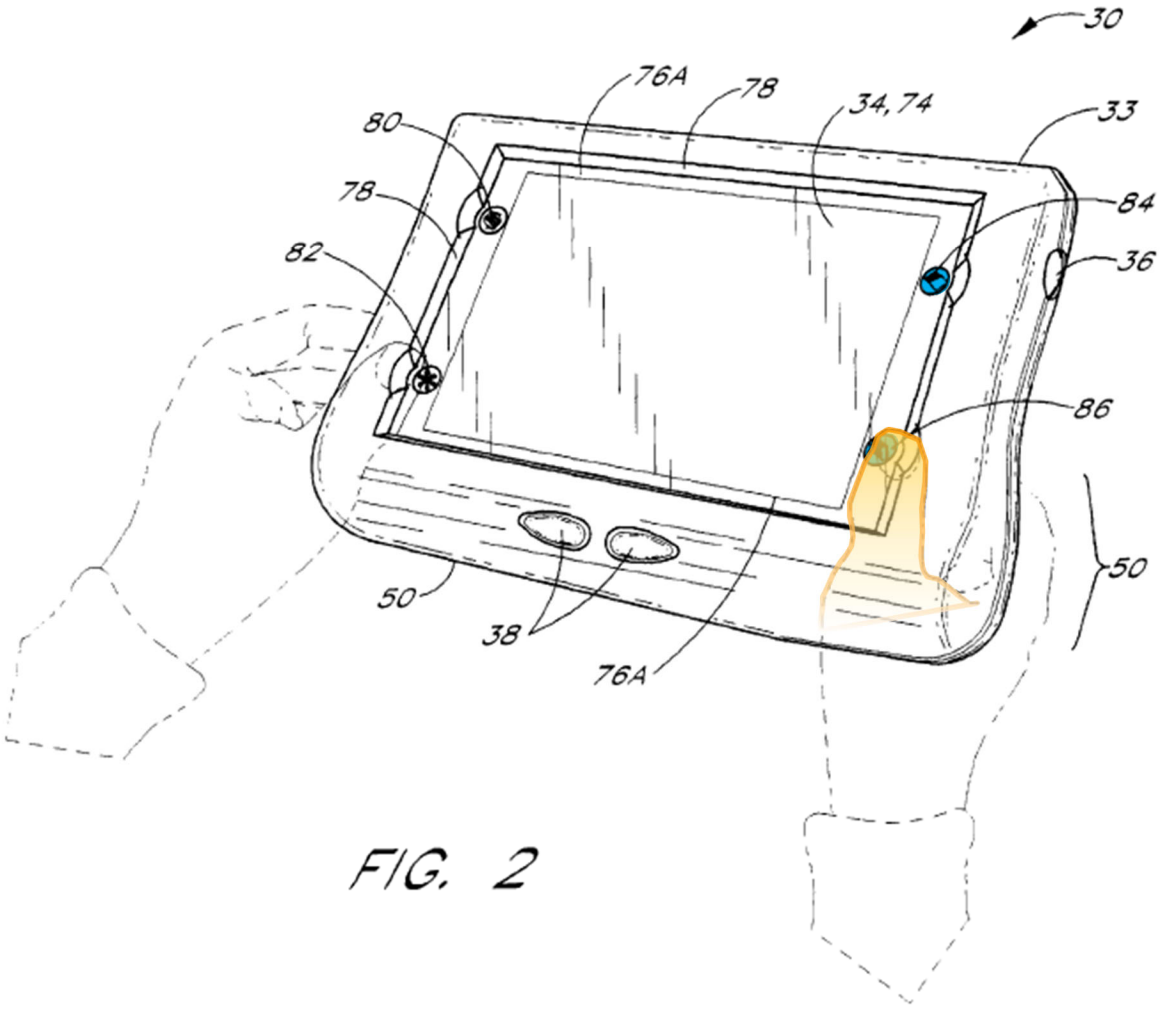


FIG. 2

Ex-1009, FIG. 2 (annotated).

9. Claim 17

The locations of book menu key 84 and library menu key 86 (representations) do not provide touch functionality for functions other than opening the book menu and library menu, respectively, as explained for [1b] (Section V.D.2.c). Ex-1009, 6:41-43; Ex-1003, ¶257.

E. [Ground 5] Claims 2-3, 7, and 9 are rendered obvious by Tarpenning and Vayda

1. Claim 2

As explained for [1c] (Section V.D.2.d), activating the function of book menu 84 causes the user interface to open book menu 85, including a list of various different services or settings for a currently active e-book application (e.g., "Highlight," "Large Text," "Add Note," "Search," "Set Bookmark Here"). Ex-1009, Table 1, FIG. 6; Ex-1003, ¶¶260-262. Tarpenning also discloses icons 80, 82, 84, 86 representing various services. Ex-1009, 6:9-14, 6:20-23, 6:35-37, 7:44-48, 12:44-45; Ex-1003, ¶260. Vayda renders obvious displaying icons representing different services or settings in Tarpenning's book menu 85. Ex-1003, ¶260.

Vayda teaches the menus may "display[] word labels, icons or both as command selectors." Ex-1007, 10:62-64; *also* Ex-1007, 7:14-18, 7:53-58. Figure 7 shows an activated menu with command icons 711, 713, 715, 716 representing different services or settings ("OPEN," "NEW FILE," "SAVE IN MEMORY (HARD DRIVE)," "CLOSE"). Ex-1007, 10:21-42.

A POSITA would have found it obvious to use icons to supplement or replace text for services and settings listed in book menu 85, like a magnifying glass for "Search" and pencil for "Add Note," to provide visual aids to allow users

to quickly locate and identify functions. Ex-1007, 10:26-29; Ex-1003, ¶262; *see* Ex-1007, 11:4-14.

i. Motivation

A POSITA would have been motivated to include icons in book menu 85 because icons were well-known to help users identify and quickly locate menu items, as confirmed by Tarpenning's icons and Vayda's teaching that "menu icon[s] … become associated with the commands and command sets they represent" over time. Ex-1007, 11:4-14, FIG. 7; Ex-1009, 6:9-19, FIGS. 2, 6-13; Ex-1003, ¶¶263-266.

A POSITA alternatively would have been motivated to replace text representing the service or setting with icons to make book menu 85 more compact and obscure less of the underlying e-book content or include more items without enlarging the menu. Ex-1003, ¶¶264-266; Ex-1007, 10:62-11:4, 7:14-18, 11:4-14.

ii. Expected success

A POSITA would have expected success supplementing or replacing the text items in book menu key 84 with icons. Ex-1003, ¶267. As explained in Section V.A.4.ii (claim 2), using icons to supplement or replace text in menu lists was well-known. Ex-1003, ¶267; Ex-1009, 6:9-14, 6:20-23, 6:35-37, 7:44-48, 12:44-45; Ex-1006 (Maddalozzo), FIG. 3 (icons on touch screen); Ex-1005 (Robertson), FIG. 1 (icons supplementing text); Ex-1017 (Agulnick), 7:25-28, FIG.

4 (icons 130 and 140 representing "notebook" and "accessory" services). Thus, displaying icons in book menu key 84 would result in the ordinary, expected operation. Ex-1003, ¶267.

2. Claim 3

Tarpenning and Vayda render obvious selecting a preferred service or setting by tapping a displayed icon corresponding to the preferred service or setting. Ex-1003, ¶¶ 269-273; *supra* § V.E.1 (claim 2).

As explained in Section V.D.2.d ([1c]), Tarpenning discloses tapping icons 84, 86 to activate them and selecting menu list items, which a POSITA would understand could be tapped using a stylus or finger. Ex-1003, ¶271.

As explained in Sections V.B.2-V.B.3 (claims 6 and 7), Vayda displays menus where the menu items can be selected by "clicking" (tapping) the menu icons using a touch screen or stylus, which a POSITA would understand would be tapping the screen. Ex-1007, 6:13-18, 15:14-22, FIG. 14; Ex-1003, ¶272.

It would have been obvious to a POSITA to allow users to select services or settings from book menu 85 by tapping the desired menu item (words or icon) to facilitate ease of operation. Ex-1009, 7:51-57; Ex-1007, 6:13-18, 15:14-22; Ex-1003, ¶273.

i. Motivation

A POSITA would have been motivated to allow users to tap an icon in book menu 85 because it uses a familiar selection technique (tapping), making the user's selection easier. Ex-1003, ¶274.

Vayda teaches the desirability of using a confirmation click for novice users to avoid erroneous inputs, which a POSITA would have been motivated to use to confirm a user's highlighted menu item when scrolling through a menu. *Supra* § V.B.3 (claim 7); Ex-1007, 8:23-26; Ex-1003, ¶275.

ii. Expected success

A POSITA would have expected success tapping an icon in book menu 85 to select a preferred service or setting from the menu. Tarpenning activates services by tapping icons. Ex-1009, 6:9-14, 6:20-23, 6:35-37, 7:44-48, 12:44-45. Tapping to select a menu or list item was conventional for touch-screen interfaces, like Tarpenning's. *Supra* §§ V.A.5.ii, V.B.3. Allowing a user to tap on a displayed icon to select a preferred service or setting would result in the ordinary execution of a conventional selection action as expected. Ex-1003, ¶276.

3. Claim 7

As explained in Section V.B.3, Vayda teaches and renders obvious the subject matter of claim 7 to highlight a menu item using a "scroll position" as the

user's finger moves along the menu, and selecting the item using a confirmation tap. Ex-1003, ¶278.

A POSITA would have found it obvious to highlight a selected file (title) or application (program) in Tarpenning's library menu, like in sub-menus or dialog boxes described in claim 6 (*supra* § V.D.5), for the same reasons as in Section V.B.3.i-ii (claim 7), which apply equally to Tarpenning and Robertson. Ex-1003, ¶279.

4. Claim 9

As explained in Section V.B.4, Vayda teaches and renders obvious the subject matter of claim 9 by highlighting a menu item as the user's finger moves towards to top or bottom of the menu. Ex-1003, ¶281.

A POSITA would have found it obvious to highlight a selected file (title) or application (program) in Tarpenning's library menu as the user's finger or stylus moves along the list, like the sub-menus in claims 6 and 7 (*supra* §§ V.D.5, V.E.3), for the same reasons discussed in Section V.B.4.i-ii, which apply equally to Tarpenning and Robertson. Ex-1003, ¶282.

F. [Ground 6] Claim 12 is rendered obvious by Tarpenning and Bedford-Roberts

1. Claim 12

As explained in Section V.C.1 (claim 12), Bedford-Roberts teaches and renders obvious the subject matter of claim 12. Ex-1003, ¶¶284-285.

A POSITA would have found it obvious to advance or back up the displayed page of Tarpenning's e-book reader using the intuitive left-to-right and right-to-left gliding motions on Tarpenning's e-book reader for the same reasons discussed in Section V.C.1.i-ii, which apply equally to Tarpenning. Ex-1003, ¶285.

VI. Non-Institution Under 35 U.S.C. § 325(d) Would Be Improper

The Board should not deny institution based on *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 (Feb. 13, 2020) (precedential), and *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8 (Dec. 15, 2017) (precedential).

Under *Advanced Bionics*, step 1, and *Becton, Dickinson* factors (a), (b), and (d), none of the prior art in this Petition—Robertson, Maddalozzo, Vayda, Agulnick, Tarpenning—was before the Examiner in prosecution. *See* Ex-1002; Ex-1001.

Even under a broad reading of *Becton, Dickinson* (*Advanced Bionics*, Paper 6, at 10), denial would be improper because the prior art in this Petition is not substantially the same as, or cumulative of, any prior art considered during prosecution. The examiner allowed the claims after adding "wherein the representation of the function is not relocated or duplicated during the gliding." Ex-1002, 611. Robertson and Tarpenning are not cumulative or substantially the same as the art relied on during examination because neither reference teaches

relocating or duplicating the representation during gliding, and each consists of only one option for activating a function. *Supra* §§ V.A.3.d-V.A.3.e, V.D.2.d-V.D.2.e.

Because the references cited in this Petition are not cumulative of the prosecution prior art, *Becton, Dickinson* factors (c)-(f) are inapplicable.

VII. Non-Institution Under 35 U.S.C. § 314(a) Would Be Improper

A. Denial of institution under *Apple v. Fintiv* would be improper

The Board should institute this proceeding under 35 U.S.C. § 314(a) because all of Factors 1-6 of *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11, at 5-6 (Mar. 20, 2020) (precedential), favor institution. There is no co-pending litigation involving Petitioner, which minimizes or moots the *Fintiv* factors.

Under Factor 1, there is no co-pending litigation involving Petitioner. To Petitioner's knowledge, all litigations involving the '879 patent are stayed, which favors institution. Ex-1025; Ex-1026.

Under Factors 2-4, there is no trial date to impact Factor 2 and no investment under Factor 3, which favor institution. To the best of Petitioner's knowledge, the grounds in this Petition are not overlapping with any other proceeding involving the '879 patent.

Under Factor 5, Petitioner is not a defendant in a district court action, which favors institution.

Under Factor 6, the merits of this Petition are strong, weighing in favor of institution. *3Shape A/S v. Align Tech., Inc.*, IPR2020-00223, Paper 12, at 34 (May 26, 2020).

B. Denial of institution under *General Plastic* and *Valve Corp.* would be improper

Under *General Plastic Industrial Co. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (Sept. 6, 2017) (precedential), and *Valve Corp. v. Electronic Scripting Products, Inc.*, IPR2019-00062, Paper 11 (Apr. 2, 2019) (precedential), the Board should not exercise its discretion to deny institution.

Under Factor 1, Petitioner here is not the same as in IPR2021-00144 and is not a co-defendant of either petitioner in IPR2021-00144. Petitioner has not been sued for patent infringement of the ’879 patent. Moreover, this Petition challenges additional claims 7 and 9, which are not challenged in IPR2021-00144; thus, there is not complete overlap of the claims.

Under Factor 2, Petitioner has no knowledge whether the petitioners of IPR2021-00144 knew of the prior art in this Petition because Petitioner has not communicated with them regarding the prior art.

Under Factor 3, Patent Owner filed a preliminary response in IPR2021-00144. The Board has not yet rendered an institution decision in IPR2021-00144 when this Petition is being filed.

Under Factors 4 and 5, Petitioner became aware of the prior art asserted in this Petition around May 3, 2021, and worked diligently to file its Petition thereafter.

Under Factors 6 and 7, efficiency favors institution. Petitioner only recently learned of the '879 patent and is not a party to any litigation involving it. The Board's timeline (if any) in IPR2021-00144 will not be affected.

VIII. Mandatory Notices

A. Real Parties-in-Interest Under 37 C.F.R. § 42.8(b)(1)

Google LLC is the real party-in-interest for this Petition.[10]

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

To the best knowledge of Petitioner, the '879 patent is or has been involved in the following district court litigations and PTAB proceeding:

Name	Number	Court	Filed
Neonode Smartphone LLC v. Samsung Electronics Co. Ltd.	6:20-cv-00507	W.D. Tex.	June 8, 2020
Neonode Smartphone LLC v. Apple Inc.	6:20-cv-00505	W.D. Tex.	June 8, 2020
Samsung Electronics America, Inc. and Apple Inc. v. Neonode Smartphone LLC	IPR2021-00144	PTAB	Nov. 6, 2020

[10] Google LLC is a subsidiary of XXVI Holdings Inc., a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest here.

In addition, pending U.S. Application No. 16/796,880 claims priority to the '879 patent.

C. Lead and Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Lead Counsel	Back-Up Counsel
Erika Harmon Arner (Reg. No. 57,540) erika.arner@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP Two Freedom Square 11955 Freedom Dr. Reston, VA 20190-5675 Tel: 571-203-2700 Fax: 202-408-4400	Kevin D. Rodkey (Reg. No. 65,506) kevin.rodkey@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 271 17th Street, NW Suite 1400 Atlanta, GA 30363-6209 Tel: 404-653-6400 Fax: 202-408-4400 Yi Yu (Reg. No. 69,397) yi.yu@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP Two Freedom Square 11955 Freedom Dr. Reston, VA 20190-5675 Tel: 571-203-2700 Fax: 202-408-4400

Lead Counsel	Back-Up Counsel
	Wei Yuan (Reg. No. 71,772) Wei.yuan@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Tel: 202-408-4000 Fax: 202-408-4400

D. Service Information Under 37 C.F.R. § 42.8(b)(4)

Please address correspondence to lead and back-up counsel at the addresses above and Google-Neonode-IPR@finnegan.com. Petitioner consents to electronic service by e-mail.

IX. Standing

Petitioner certifies that the '879 patent is available for *inter partes* review and Petitioner is not barred or estopped from requesting *inter partes* review to challenge the claims on the grounds herein.

X. Conclusion

Petitioner has established a reasonable likelihood of prevailing for the challenged claims and requests the Board institute *inter partes* review and cancel each challenged claim as unpatentable.

Date: June 14, 2021

Respectfully submitted,

/Erika H. Arner/
Erika H. Arner
Lead Counsel
Reg. No. 57,540

CERTIFICATE OF COMPLIANCE

The undersigned hereby certifies that the foregoing **Petition for *Inter Partes* Review** contains 13,993 words, excluding those portions identified in 37 C.F.R. § 42.24(a), as measured by the word-processing system used to prepare this paper.

/Erika H. Arner/
Erika H. Arner
Lead Counsel
Reg. No. 57,540

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing **Petition for *Inter Partes* Review** was served on June 14, 2021, by Priority Mail Express or by means at least as fast and reliable as Priority Mail Express at the following address of record for the subject patent. The associated **Exhibits 1001 through 1029** and the **Power of Attorney** were also served on June 14, 2021.

Marc Berger
Soquel Group, LLC
P.O. Box 2063
Santa Cruz, CA 95063

A courtesy copy has also been mailed to lead counsel for Patent Owner in IPR2021-00144 at:

Robert M. Asher
Sunstein LLP
100 High Street
Boston, MA 02110-2321

Date: June 14, 2021

/Daniel E. Doku/
Daniel E. Doku
Litigation Legal Assistant
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, LLP