UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,

Petitioner,

v.

MESSAGELOUD, INC.,

Patent Owner.

Case No. IPR2025-01429
U.S. Patent No. 10,516,775

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 10,516,775

TABLE OF CONTENTS

I. PRELIMINARY STATEMENT 1

II. STATE OF THE ART 1

III. THE '775 PATENT 3

A. The '775 Patent Specification 3

B. Prosecution History 5

C. Claim Construction 6

IV. STATEMENT OF PRECISE RELIEF REQUESTED 6

V. LEVEL OF ORDINARY SKILL 7

VI. GROUND 1: CLAIMS 1-24 ARE UNPATENTABLE OVER *BOELTER* AND *GRUBER* (GROUND 1A) OR *BOELTER*, *GRUBER*, AND *POLAK* (GROUND 1B) 8

A. Prior Art Overview 8

1. *Boelter* 8

2. *Gruber* 10

3. *Polak* 10

B. Combining *Boelter*, *Gruber*, and/or *Polak* 11

C. Independent Claim 1 17

1. 1[pre]: "A computer implemented method to be carried out with a processor, a memory, and a touch screen, comprising:" 17

2. 1[a]: "determining without any input by a user that one or more different types of messages selected from the group consisting of a text message, an email, and a message from a messenger application have been received;" 18

3. 1[b]: "analyzing without any input by the user a content of the received text message, message from the messenger application, or email;".....19

4. 1[c]: "placing the received email, message from the messenger application, or text message in a queue to be read aloud in order of receipt time;".....21

5. 1[d]: "informing the user without any input by the user that the received text message, message from the messenger application, or email has been received from a sender by reading aloud at least an identity of the sender;".....25

a. Ground 1A.....26

b. Ground 1B.....27

6. 1[e]: "allowing the user a time to take an affirmative action to stop reading aloud to the user a body of the received text message, message from the messenger application, or email;".....28

7. 1[f]: "based on not receiving an input corresponding to the affirmative action from the user, reading aloud the body of the received text message, message from the messenger application, or email."".....32

D. Dependent Claims.....34

1. Claim 2: "The computer implemented method of claim 1, wherein analyzing the content comprises parsing the email into a subject, the sender, and the body, and parsing the text message or the message from the messenger application into the sender and the body."".....34

2. Claim 3: "The computer implemented method of claim 1, wherein the computer implemented method is carried out with an application that is configured to run in a foreground or in a background, and is further configured to switch back and forth between the foreground and the

background, and is further configured to read aloud the identity of the sender, and the body of the received email, text message, or message from the messenger application while the application is in the background." 36

3. Claim 4: "The computer implemented method of claim 1, further comprising prompting the user to call the sender without any input from the user." 39

4. Claim 5: "The computer implemented method of claim 1, wherein a different interface is presented on the touch screen depending on an activity of the user." 42

5. Claim 6: "The computer implemented method of claim 5, wherein the activity is driving a vehicle." 44

6. Claim 7: "The computer implemented method of claim 5, wherein the activity is working with a desktop, a laptop, or a tablet computer."....45

7. Claim 8: "The computer implemented method of claim 5, wherein the activity is exercise (Active Mode)." 46

8. Claim 9: "The computer implemented method of claim 1, wherein informing the user comprises reading aloud a name of the sender."....49

9. Claim 10: "The computer implemented method of claim 1, the method further comprising based on receiving the input corresponding to the affirmative action from the user, not reading aloud the body of the received text message, message from the messenger application, or email to the user." 49

10. Claim 11: "The computer implemented method of claim 1, further comprising receiving the input from the user without the user looking at the touch screen."....50

11. Claim 12: "The computer implemented method of claim 1, further comprising displaying an interface on the touch screen with no prompts for the user to touch."....50

12. Claim 13: "The computer implemented method of claim 1, further comprising receiving the input from the user based on gestures by the user."....52

13. Claim 14: "The computer implemented method of claim 13, wherein the input from the user is limited to tapping or swiping the touch screen without a need to look at the touch screen."....52

14. Claim 15: "The computer implemented method of claim 1, further comprising informing the user by voice of an option to call the sender." 53

15. Claim 16: "The computer implemented method of claim 15, further comprising, based on receiving an input from the user to call the sender, calling the sender."....54

16. Claim 17: "The computer implemented method of claim 2, wherein informing the user comprises informing the user of the identity of the sender and the subject of the received email, or the identity of the sender of the received text message or the message from the messenger application."....55

17. Claim 18: "The computer implemented method of claim 1, wherein after reading aloud the body of the received email, text message or message from the messenger application, the received email, text message or message from the messenger application is automatically dismissed and a status of the received email, text message or message from the messenger application is changed to read."....56

18. Claim 19: "The computer implemented method of claim 18, wherein after the received email, text message or message from the messenger application is read aloud and the status changed to read, the user is notified that a new email, text message, or message from the messenger application has been received."....59

19. Claim 20: "The computer implemented method of claim 1, wherein the reading aloud is carried out by an audio system in a vehicle." 60

20. Claim 21: "The computer implemented method of claim 1, wherein the input from the user is limited to a) pausing or rereading the reading of the received email, text message or message from the messenger application; b) dismissing the received email, text message, or message from the messenger application; c) deleting the received email; and d) calling the sender." 61

21. Claim 22: "The computer implemented method of claim 1, wherein the method is carried out on a device selected from the group consisting of: a smart phone, a mobile phone, a phablet, a tablet, a wearable device, a watch, and a vehicle operating device." 62

E. Independent Claim 23 63

1. 23[pre]: "A mobile phone, comprising: one or more processors; a memory, a touch screen; and one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for steps comprising:" 63

2. 23[a]: "determining without any input, by a user that one or more different types of messages selected from the group consisting of a text message, an email, and a message from a messenger application have been received;" 64

3. 23[b]: "analyzing without any input by the user a content of the received text message, message from the messenger application, or email;" 64

4. 23[c]: "placing the received email, message from the messenger application, or text message in a queue to be read aloud in order of receipt time;" 64

5. 23[d]: "informing the user without any input by the user that the received text message, email, or message from the messenger application has been received from a sender by reading aloud at least an identity of the sender;" 65

6. 23[e]: "allowing the user a time to take an affirmative action to stop reading aloud to the user a body of the received text message, message from the messenger application, or email;".....65

7. 23[f]: "based on not receiving an input corresponding to the affirmative action from the user, reading aloud the body of the received text message, message from the messenger application or email, and for the received email, further reading aloud a subject of the received email." 65

F. Independent Claim 24 67

1. 24[pre]: "A system for delivering one or more of a text message and an email and a message from the messenger application to a user engaged in an activity, the system comprising a processor, a memory, and a touch screen, the system configured to deliver the text message, the message from the messenger application, and the email to the user by:" 67

2. 24[a]: "running an application configured to deliver one or more of a text message, an email, or a message from the messenger application to a user while the user is engaged in an activity;" 68

3. 24[b]: "determining without any input by a user that one or more different types of messages selected from the group consisting the text message, the email, and the message from the messenger application have been received;" 68

4. 24[c]: "placing the received email, message from the messenger application, or text message in a queue to be read aloud in order of receipt time;" 68

5. 24[d]: "analyzing without any input by the user a content of the received text message, message from the messenger application, or email;" 69

6. 24[e]: "informing the user without any input by the user that the received text message, message from the messenger application, or email has been received from a sender by reading aloud at least an identity of the sender;" 69

7. 24[f]: "allowing the user a time to take an affirmative action to stop reading aloud to the user a body of the received text message, message from the messenger application, or email;" 69

8. 24[g]: "based on not receiving an input corresponding to the affirmative action from the user, reading aloud the body of the received text message, message from the messenger application, or email, and for the received email, further reading aloud a subject of the received email." 70

VII. MANDATORY NOTICES 70

A. Real Parties-in-Interest 70

B. Related Matters 70

C. Lead and Backup Counsel, and Service Information 71

VIII. GROUNDS FOR STANDING 72

IX. CONCLUSION 72

TABLE OF AUTHORITIES

Page(s)

Cases

Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.,
868 F.3d 1013 (Fed. Cir. 2017) 6

Penumbra, Inc. v. RapidPulse, Inc.,
IPR2021-01466, Paper 34 (Mar. 10, 2023) 7

Statutes

35 U.S.C. § 102(a)(1).....6

35 U.S.C. § 102(a)(2).....6

35 U.S.C. § 103.....7

LIST OF EXHIBITS

Exhibit	Description
Ex. 1001	U.S. Patent No. 10,516,775 ("the '775 patent")
EX1002	Prosecution File History of the '775 patent
EX1003	Declaration of Dr. Bederson ("Bederson")
EX1004	Curriculum Vitae of Dr. Bederson
EX1005	U.S. Patent Application Publication No. 2014/0303842 A1 ("*Boelter*")
EX1006	U.S. Patent Application Publication No. 2013/0275138 A1 ("*Gruber*")
EX1007	U.S. Patent Application Publication No. 2015/0350400 A1 ("*Polak*")
EX1008	U.S. Patent Provisional Application No. 61/942,059 ("*Polak Provisional*")
EX1009	*Ford SYNC - Reading Text Messages* (YouTube, June 21, 2013), https://www.youtube.com/watch?v=MaZZ7t3ppn0
EX1010	U.S. Patent Application Publication No. 2013/0275899 A1 ("*Schubert*")
EX1011	U.S. Patent Application Publication No. 2014/0195252 A1 ("*Gruber '252*")
EX1012	U.S. Patent Application Publication No. 2013/0172027 A1 ("*Sturges*")
EX1013	U.S. Patent No. 8,577,422 B1 ("*Ledet*")
EX1014	U.S. Patent No. 7,920,682 B2 ("*Byrne*")
EX1015	U.S. Patent Application Publication No. 2014/0273975 A1 ("*Barat*")
EX1016	Michael A. Nees and Bruce N. Walker, Chapter 2: *Auditory Displays for In-Vehicle Technologies*, Reviews of Human Factors & Ergonomics, Vol. 7, pp. 58-99, at 60 (2011)

Exhibit	Description
EX1017	Charles Arthur, *The History of Smartphones: Timeline*, Jan. 24, 2012, *available at* https://www.theguardian.com/technology/2012/jan/24/smartphones-timeline
EX1018	Ryan Blundell, *CrackBerry Guide to BlackBerry Voice to Text Apps*, Aug. 4, 2010, *available at* https://crackberry.com/crackberry-guide-blackberry-voice-text-apps
EX1019	Athima Chansanchai, *This Hands-Free Testing App Is Easy to Use, Accurate*, Mar. 9, 2012, *available at* https://www.nbcnews.com/tech/gadgets/hands-free-texting-app-easy-use-accurate-flna383127

I. PRELIMINARY STATEMENT

Apple Inc. respectfully requests *inter partes* review and cancellation of claims 1-24 of U.S. Patent No. 10,516,775.

The ’775 patent purports to invent systems that allow users to listen to their emails, texts, and other messages while driving, exercising, or engaging in other activities. But *Boelter*, *Gruber*, and *Polak* demonstrate that these concepts were already known and obvious. *Boelter* discloses presenting a queue of notifications, including both new email and text messages, to a driver without user input. *Gruber* and *Polak* each suggest reading these notifications to the user.

As shown by the grounds below, these references render all claims of the ’775 patent unpatentable.

II. STATE OF THE ART

The commercialization of smartphones—including the 2007 launch of the iPhone—revolutionized communication, enabling users to have their favorite messaging applications all on one device. EX1017, 1. Although constant connection came with many benefits, it also created problems, such as texting while driving. EX1018, 1-2. In response, those in the art developed ways to decrease the risks associated with messaging while driving. EX1018, 1-7.

By November 2014, the ’725 patent’s claimed priority date, those in the art had already developed methods for managing messages to minimize distracted

driving. *Boelter* discloses one such system, which collects emails, text messages, and other notifications into a single list and displays the list to a user after the system determines it is safe to do so:

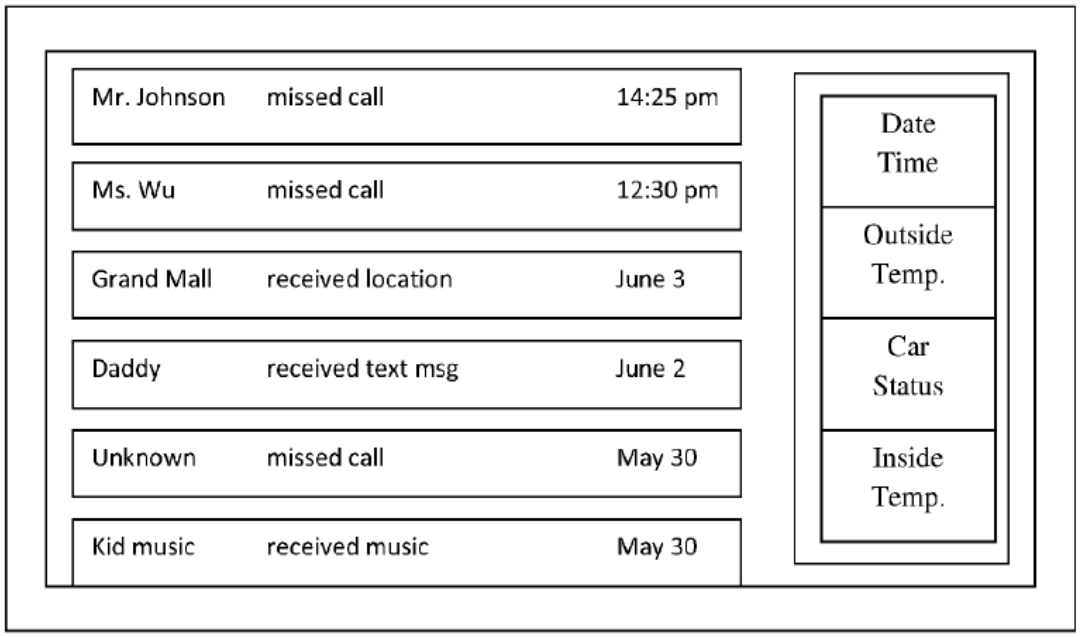


Fig. 4

Boelter, ¶¶[[0010], [0028], [0036], Fig. 4.

Reading groups of messages was also well known before November 2014. *Gruber* discloses that virtual assistants had been configured to read lists of text messages, emails, and other notifications. *Gruber*, ¶¶[0467]-[0472], [0682]-[0689]. *Gruber* and *Polak* describe how reading messages aloud minimizes users' need to read from a display while driving. *Polak*, ¶[0021]; *Gruber*, ¶¶[0166]-[0167],

[0191]. In fact, devices were even configured to automatically read messages to users without any user input. *Gruber*, [0409]-[0419], *Polak*, ¶[0019]; EX1018, 1-7.

Together, this art discloses the concepts later claimed in the ’775 patent.

III. THE ’775 PATENT

The ’775 patent was filed January 11, 2019, and claims priority to November 21, 2014.[1] EX1001, 1 (code (22); code (60)).

A. The ’775 Patent Specification

The ’775 patent discloses systems that allow a user to receive text messages, email, or other messages while engaged in an activity. EX1001, 1:19-34, 5:8-23, 8:22-28. Figure 22 illustrates the purported invention:

[1] Petitioner does not concede the claims are entitled to this priority date, but the relied-upon prior art references predate it. Petitioner reserves the right to challenge the priority date and to respond if Patent Owner attempts to antedate any of the prior art.

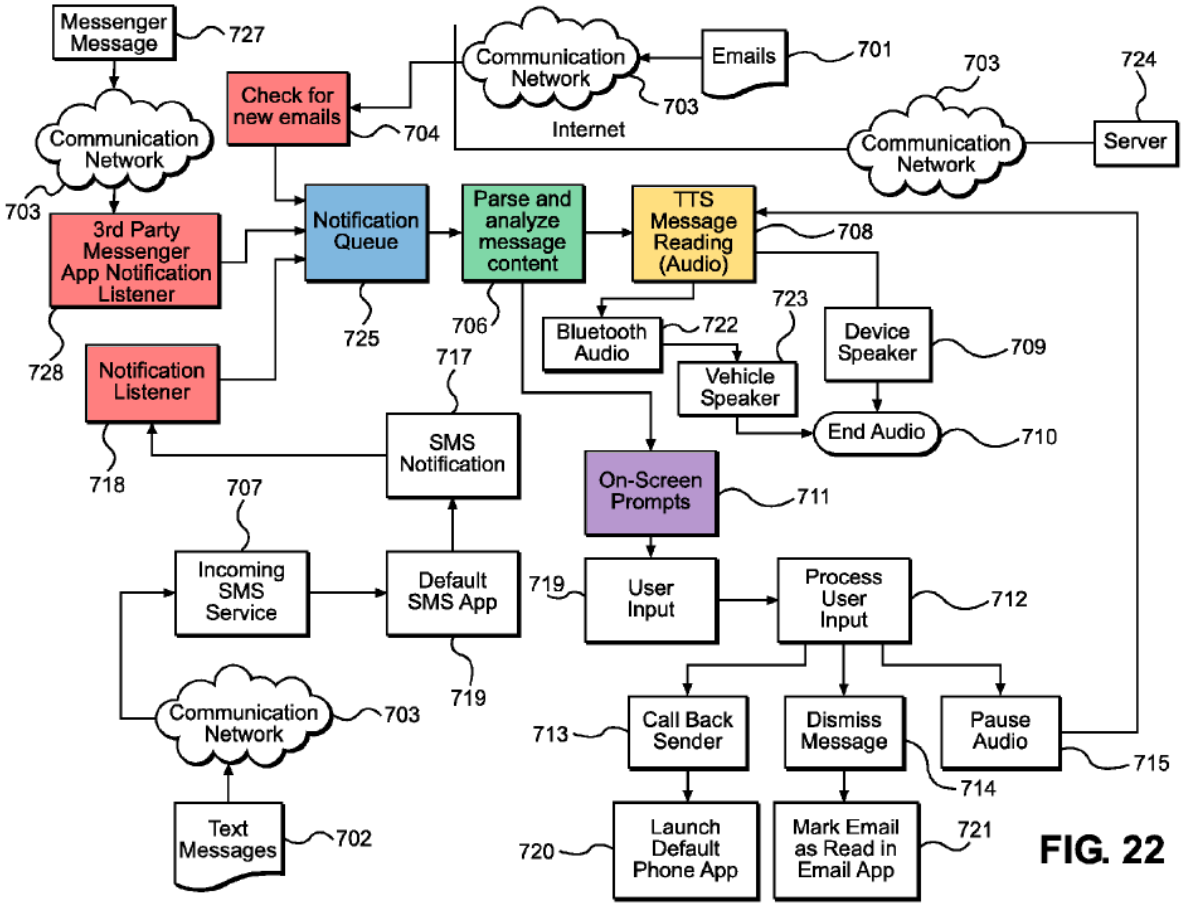


FIG. 22

EX1001, 15:7-16:29, Fig. 22 (annotated). The user's mobile phone or other device receives new messages (701, 702, 727). EX1001, 15:32-45. The new messages are placed in a notification queue 725. EX1001, 15:45-46. The messages are analyzed 706 by parsing the messages into sub-parts, including message type, sender, subject, and body. EX1001, 15:53-62. The messages are read aloud 708 to the user. EX1001, 15:62-64, 16:2-16:5. And the user may be prompted 711 to respond to the messages. EX1001, 16:5-13.

Figure 24 of the '775 patent illustrates a notification queue 726 containing email, text messages, and messenger messages:

:

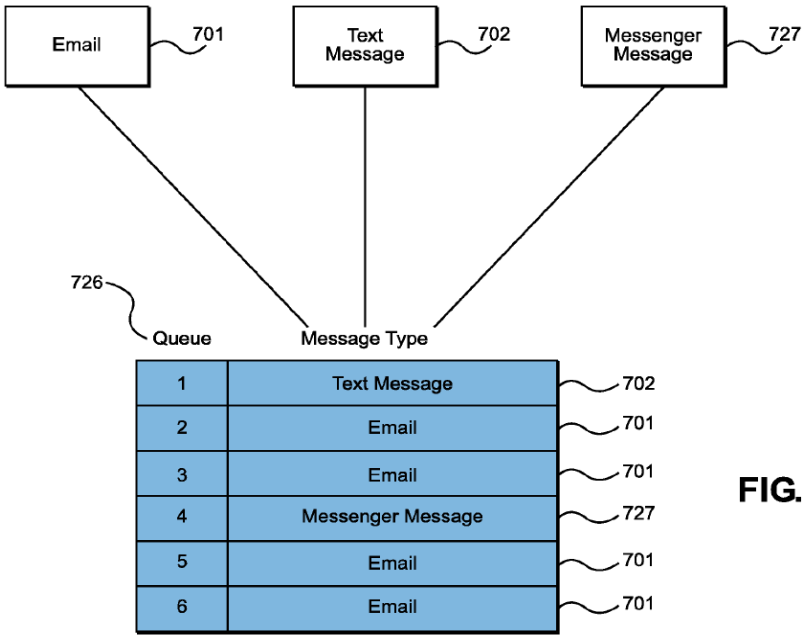


FIG. 24

EX1001, Fig. 24 (annotated), 15:45-53, 16:30-37.

B. Prosecution History

The '775 patent did not receive any prior art rejections. EX1002, 139-142. The Examiner rejected the pending claims based on nonstatutory double patenting. EX1002, 140-142. After the Applicant filed a terminal disclaimer (EX1002, 188), the Examiner allowed the claims, noting "[n]o new prior art has been found that suggests or renders obvious the limitations of independent claims 37, 59 and 60 disclosing the detailed method for analyzing, organizing, and informing a user without any input from the user, of a variety of incoming messages." EX1002, 230.

The Applicant filed an Amendment after Notice of Allowance that "correct[ed] typographical issues," which the Examiner accepted. EX1002, 264-268, 275.

C. Claim Construction

Claims must be construed only to the extent necessary to resolve a controversy. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). Here, no terms need construction to resolve the controversy in this forum.[2]

IV. STATEMENT OF PRECISE RELIEF REQUESTED

The following prior art discloses the elements of claims 1-24:

Exhibit	Prior Art	Filed	Published	Prior Art Status at least under AIA 35 U.S.C.
EX1005	*Boelter* (U.S. 2014/0303842 A1)	Aug. 31, 2011	Oct. 9, 2014	§102(a)(1)
EX1006	*Gruber* (U.S. 2013/0275138 A1)	June 8, 2013	Oct. 17, 2013	§102(a)(1)
EX1007	*Polak* (U.S. 2015/0350400 A1)*	Apr. 23, 2015	Dec. 3, 2015	§102(a)(2)

[2] Petitioner reserves the right to argue limitations in the challenged claims are indefinite, lack written description support, or are invalid for other reasons not presented here.

**Polak* claims priority to U.S. Provisional Application No. 61/942,059 ("*Polak Provisional*," EX1008), filed February 20, 2014. For prior art purposes, *Polak* is entitled to the *Polak Provisional*'s filing date as demonstrated by parallel citations below. *See Penumbra, Inc. v. RapidPulse, Inc.*, IPR2021-01466, Paper 34 (Mar. 10, 2023) (precedential). Further, while not required, the provisional application supports at least one claim of *Polak*. EX1003, ¶¶78-82.

And the following grounds render the claims obvious:

Ground	Claims	Statutory Basis	Prior Art
1A	1-24	§103	*Boelter* and *Gruber*
1B	1-24	§103	*Boelter*, *Gruber*, and *Polak*

V. LEVEL OF ORDINARY SKILL

A person having ordinary skill in the art ("ordinary artisan") would have had at least a bachelor's degree in computer science, computer architecture, or a similar field and two to three years of experience designing or analyzing messaging or notification systems. Additional relevant work experience can compensate for less education, and vice versa. EX1003, ¶60.

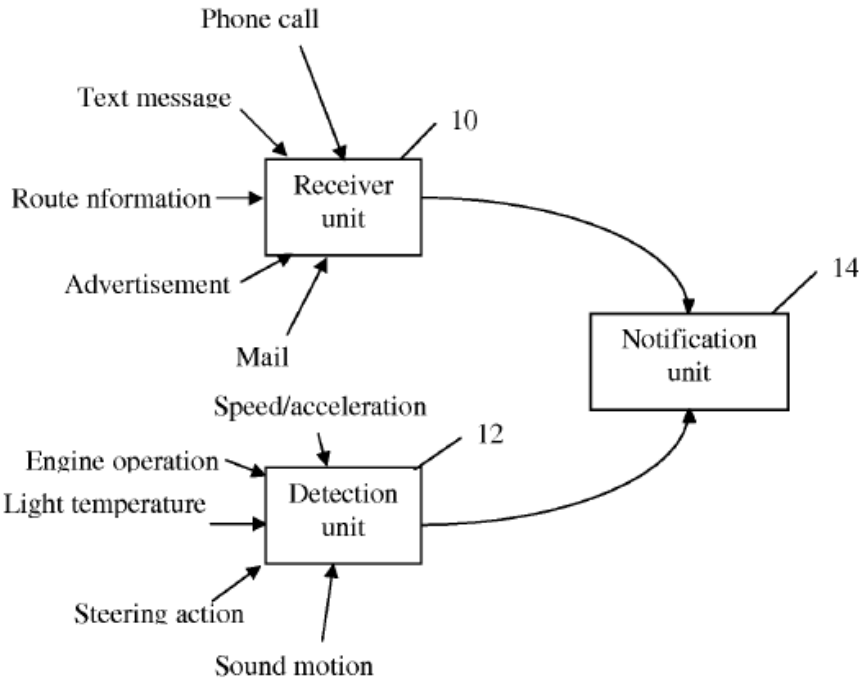
VI. GROUND 1: CLAIMS 1-24 ARE UNPATENTABLE OVER *BOELTER* AND *GRUBER* (GROUND 1A) OR *BOELTER*, *GRUBER*, AND *POLAK* (GROUND 1B)

A. Prior Art Overview

1. *Boelter*

Boelter discloses a system for managing in-vehicle message notifications. *Boelter*, Abstract, ¶¶[0009]-[0010]. This system reduces distractions, especially during critical driving situations. *Boelter*, ¶[0008].

Boelter's system includes a detection unit, a receiver unit, and a notification unit:



Boelter, Fig. 1, ¶¶[0010], [0024]. The receiver unit "receive[s] a number of different types of messages, such as phone calls, text messages, [and] e-mails …,"

and forwards them to the notification unit. *Boelter*, ¶¶[0025], [0028], [0010]-[0011]. The detection and notification units work together to determine when and how to notify the driver of these new messages. *Boelter*, ¶¶[0011], [0010], [0026], [0028]. Notifications may, for example, be delayed until after a critical driving situation ends or prioritized based on pre-set driver preferences. *Boelter*, ¶¶[0029]-[0032].

Boelter's system notifies users of different types of messages at the same time. *Boelter*, ¶¶[0010], [0013], [0017], [0025], [0036]. Figure 4, reproduced below, illustrates a chronological list of missed calls, text messages, and notifications. *Boelter*, ¶¶[0036], [0013], [0017], Fig. 4.

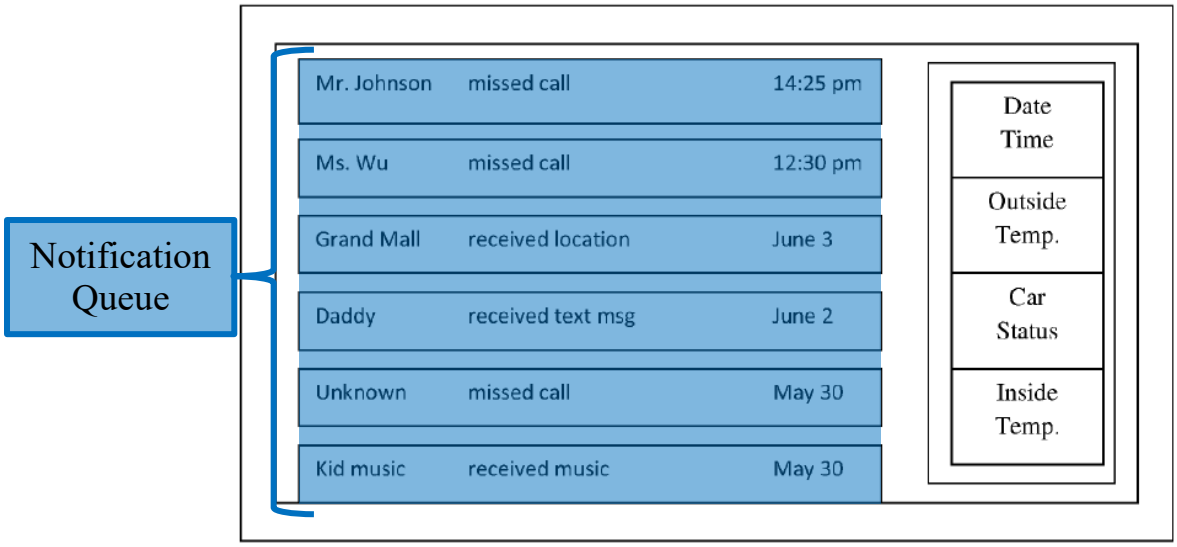


Fig. 4

Boelter, Fig. 4 (annotated). A touchscreen allows for user input using "intuitive finger gesture[s] … without the driver having even to look at the screen." *Boelter*, ¶[0034]; *see also Boelter*, ¶¶[0031]-[0033], Fig. 3.

2. *Gruber*

Gruber discloses a virtual assistant that can read messages to a user. *Gruber*, ¶¶[0009], [0016]. The assistant notifies the user of various message types, including texts, emails, and instant messaging. *Gruber*, ¶¶[0081], [0111]-[0118], [0187], [0213], [0315]-[0322], [0409]-[0496], [0556]. It can be used for receiving and replying to messages when "the user may be unable to interact with device 60, for example, if he or she is driving or engaged in some other activity." *Gruber*, ¶¶[0168]-[0170]; *see also Gruber*, ¶¶[0662]-[0663].

Gruber discloses processes for reading lists of messages that allow the user to respond, delete, or repeat messages. Gruber, ¶ [0686]. *Gruber* provides a pause between the identification of the sender and the reading of the message body, giving the user time to stop the playback of the message. *Gruber*, ¶¶[0682]-[0688]; *see also Gruber*, ¶[0270]. Unless interrupted by the user, *Gruber's* system reads the message body after identifying the sender's name. *Id*.

3. *Polak*

Polak discloses methods and systems that "enable drivers to hear text messages vocally, while driving, and/or to respond to messages in various manners

without touching the phone." *Polak*, ¶[0019]; EX1008, ¶[0018]. *Polak*'s program may be carried out by an application on a mobile device. *Polak*, ¶[0022]; EX1008, ¶[0020]. "FIG. 6 is a schematic illustration of an exemplary application screen 150 displayed on a display of mobile device 100." *Polak*, ¶[0029], Fig. 6; EX1008, ¶[0026], Fig. 5.

Fig. 6

Polak, ¶[0029], Fig. 6; EX1008, ¶[0026], Fig. 5.

B. Combining *Boelter*, *Gruber*, and/or *Polak*

Boelter, *Gruber*, and *Polak* (the "asserted references") are analogous art to the '775 patent because they are in the same field of endeavor: management of incoming notifications. *Compare, e.g.*, *Boelter*, ¶¶[0001], [0009]-[0018], *Polak*, ¶¶[0018]-[0019], [0027-[0035], *and Gruber*, ¶¶[0409]-[0496], [0682]-[0693], *with*

EX1001, 1:19-21, 1:59-62; EX1003, ¶83. Further, *Boelter*, *Gruber*, and *Polak* share a common goal with the '775 patent: improving the delivery of notifications to users engaged in a distracting activity, like driving. *Compare, e.g.*, *Boelter*, ¶¶[0008]-[0010], *Polak*, ¶¶[0001], [0018]-[0019], [0021], *and Gruber*, ¶¶[0008], [0567], *with* EX1001, 1:59-62; EX1003, ¶83.

An ordinary artisan would have been motivated to modify *Boelter* in view of *Gruber* and *Polak*. The asserted references describe notification systems with similar functions and uses. EX1003, ¶¶83-86. All contemplate using their systems in vehicles. *Boelter*, ¶[0010]; *Gruber*, ¶¶[0011], [0182]; *Polak*, ¶[0026]; EX1008, ¶[0023]. They highlight that their systems are configured to detect varied driving conditions. *Boelter*, ¶¶[0010]-[0011], *Gruber*, ¶¶[0205]-[0207];. *Polak*, ¶[0026] EX1008, ¶[0023]. And they describe systems that respond to user input to review, respond to, or delete messages. *Boelter*, ¶¶[0032]-[0034] (touch input); *Gruber*, ¶¶[0210], [0562] (voice or touch); *Polak*, ¶[0032] (voice or touch) EX1008, ¶[0029].

Aiming to minimize distracted driving, *Boelter* suggests limiting the text displayed to a driver. *Boelter*, ¶[0008]. For example, *Boelter* suggests configuring a screen to allow for "intuitive finger gesture[s] … without the driver having even to look at the screen." *Boelter*, ¶[0034], *see also Boelter*, ¶¶[0031]-[0033], Fig. 3. Although *Boelter* discloses providing some notifications on a touchscreen, an

ordinary artisan would have understood *Boelter* is not limited to this embodiment—*Boelter* says its "concepts can be used to notify the driver of an incoming message in one of a number of different ways." *Boelter*, ¶[0010]; EX1003, ¶85. An ordinary artisan would have understood *Boelter* to be compatible with, and ready for improvement by, other user notification schemes (like reading messages aloud). EX1003, ¶85.

An ordinary artisan would have looked to *Gruber* and *Polak* for other ways of notifying a driver because they suggest minimizing drivers' need to read from screens. EX1003, ¶¶85-86. *Gruber* indicates that "looking at and touching the screen" are "impractical to perform in certain contexts." *Gruber*, ¶[0155]. By providing audio output, *Gruber's* system can inform the user of available actions "reduc[ing] the need for a user to interact with a display screen and/or to use a touch interface." *Gruber*, ¶¶[0166]-[0167], [0191]. *Gruber* further contemplates a combination of audio and visual outputs. *Gruber*, ¶[0006]. *Polak* describes a process where incoming messages are announced, allowing users to interact with them "without taking their eyes off the road." *Polak*, ¶[0021]; EX1008, ¶[0019]. After sounding a new message, *Polak* instructs that a user provides "an instruction vocally, or … by touching the screen." *Polak*, ¶[0032]; EX1008, ¶[0029].

An ordinary artisan would have been motivated to incorporate *Gruber's* and *Polak*'s teachings related to reading messages aloud into *Boelter*'s notification

method because doing so would further *Boelter*'s goal of limiting distractions. EX1003, ¶87. Indeed, such a combination would encourage the driver to look at the screen less, reducing the amount of time a driver would take their eyes off the road. EX1003, ¶87. An ordinary artisan would have found reading incoming messages (as *Polak* suggests) and lists of new messages and notifications (as *Gruber* suggests) to be predictable improvements to *Boelter*'s text-based notification system because text-to-speech message reading had already been implemented in vehicles. EX1003, ¶87.

An ordinary artisan would have further been motivated to read *Boelter*'s mixed message queue aloud because it would simplify message review for the user. EX1003, ¶88. For example, reading aloud *Boelter*'s queue would prevent a user from toggling between multiple applications to review different messages. EX1003, ¶¶88, 137-142. A user would not have to decide which messages to select as they are all read aloud, simplifying the process. EX1003, ¶88.

An ordinary artisan would have at least had a reasonable expectation of success in modifying *Boelter* in view of *Gruber* and/or *Polak* because virtual assistants were well known, and implementing text-to-speech in *Boelter* and aspects of the *Gruber* and *Polak* processes discussed herein would not have required physically changing *Boelter*'s system or altering its other features. EX1003, ¶¶89, 142. For example, *Boelter* describes systems "installed [i.e., carried

out] in a vehicle" or integrated "in a mobile communication device such as a smartphone" comprised of computing components similar to those disclose in *Gruber* and *Polak*. *Boelter*, ¶¶[0018], [0024], [0028], [0030]; *Gruber*, ¶¶[0056]-[0066], Figs. 9, 10; *Polak*, ¶[0025]; EX1008, ¶[0038]; EX1003, ¶89. Because *Boelter* already included these physical components, only software modification would be needed for *Boelter*'s notification queue to be read aloud sequentially (as *Gruber* suggests). EX1003, ¶¶89, 95-96, 104-112. An ordinary artisan would have understood implementing text-to-speech would have been a routine programming change for ordinary artisans by the claimed priority date. EX1003, ¶¶89, 112; EX1014, 4:63-5:19; EX1016, 59. Indeed, audio playback of text messages was commercially available in vehicles and other devices well over a year earlier. EX1009; *see also* EX1017. *Gruber* suggests that the programming of hands-free reading is applicable beyond its disclosed embodiments. *Gruber*, ¶¶[0736]-[0745]. *Gruber* even expressly discloses reading a "plurality of data items [] presented on the visual interface" (*Gruber*, ¶[0696]) like *Boelter* discloses. EX1003, ¶¶89, 108. Ordinary artisans would have known "audio has been and will continue to be an integral mode of information display in [in-vehicle technologies]." EX1018, 60; EX1003, ¶89. An ordinary artisan would therefore have had the skills to implement this routine programming. EX1003, ¶¶89, 112 (discussing EX1009; EX1018, 1-7;

EX1016, 79-80; *Polak*, ¶[0019]; EX1008, ¶[0018]; *Gruber*, ¶¶[0056]-[0057], [0736], [00744]).

Combining the references would apply a known technique—*Gruber's* and *Polak*'s audible notifications—to a similar system—*Boelter*'s notification system. EX1003, ¶90. Reading *Boelter*'s notifications aloud, as *Gruber* and *Polak* suggest, would yield predictable results: reducing user reliance on textual display while allowing the user to select which items to hear. EX1003, ¶¶85-86, 90-91, 127. An ordinary artisan would have seen this as an improvement given *Gruber*'s disclosure that reading text to the user "may be beneficial where it may be unsafe for a user to read text from a screen, such as when the user is driving …." *Gruber*, ¶[0282]; EX1003, ¶90.

An ordinary artisan would have been motivated to use *Gruber*'s and *Polak's* read-aloud approaches because there were few other ways to present messages and notifications without requiring users to read from a screen. EX1003, ¶91. Because text-to-speech processing was well understood and commercially available, the combination would not have involved the exploration of new technology. *Gruber*, ¶[0188] (describing commercially available technology for implementing method steps); *Polak*, ¶[0029]; EX1008, ¶[0026]; EX1003, ¶91. Moreover, *Boelter*'s notification unit determines "when and ***how*** to notify a driver." *Boelter*, ¶¶[0025], [0037] (emphasis added). An ordinary artisan would have found *Polak*'s and

Gruber's audible notifications a suitable option for ***how*** to notify a driver in *Boelter*'s method because it would improve user safety. EX1003, ¶91.

An ordinary artisan would have understood that combining *Boelter* with *Gruber* and *Polak* as further detailed below (Sections VI.C-VI.F) would not affect *Boelter*'s operations in other respects because the combinations further *Boelter*'s goal of reducing distractions without removing any features. EX1003, ¶92.

C. Independent Claim 1

1. 1[pre]: "A computer implemented method to be carried out with a processor, a memory, and a touch screen, comprising:"

Boelter discloses 1[pre]. EX1003, ¶¶94-96. *Boelter* discloses its notification method may be implemented as a program "stored on a computer-readable medium" (i.e., memory) that is executed by a vehicle control unit (i.e., processor) and where the notifications/messages are displayed on a touchscreen. *Boelter*, ¶¶[0018], [0024], [0028], [0030].

Boelter's Figure 2 illustrates a notification unit 14 comprising control unit 16, memory 18, and touchscreen 20:

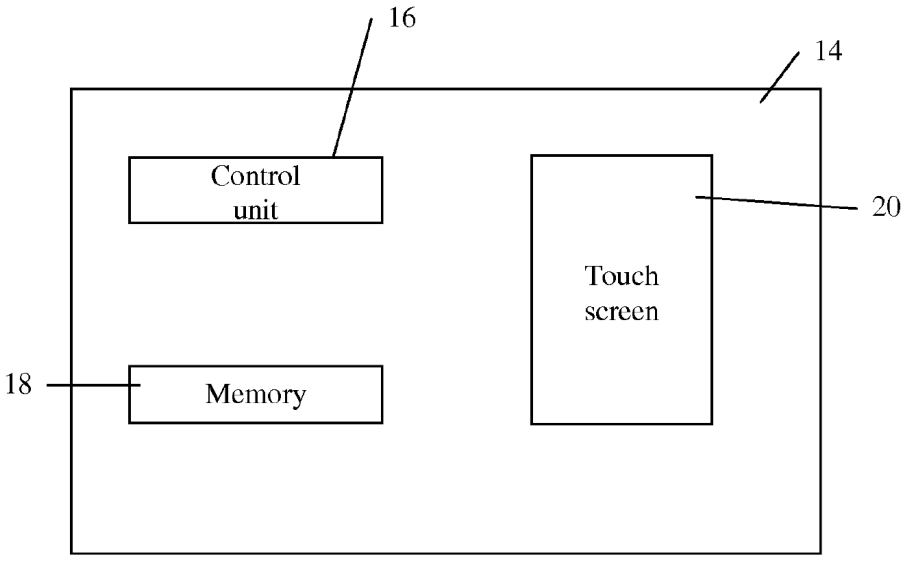


Fig. 2

Boelter, ¶[0028], Fig. 2. Control unit 16 is a processor because it determines whether, when, and how to display different message types based on information stored in memory 18. *Boelter*, ¶[0028]; EX1003, ¶¶95-96. These components carry out *Boelter*'s computer-implemented method.

2. 1[a]: "determining without any input by a user that one or more different types of messages selected from the group consisting of a text message, an email, and a message from a messenger application have been received;"

Boelter discloses 1[a]. EX1003, ¶¶97-99.

Boelter's receiver unit 10 is "preferably configured to receive a number of different types of messages, such as phone calls, text messages, [and] e-mails …." *Boelter*, ¶[0025], *see also Boelter*, ¶[0010]. *Boelter* thus discloses a user that

receives at least two different types of messages (e.g., texts and emails). EX1003, ¶98.

Boelter determines—without any user input—that two or more different types of messages have been received because receiver unit 10 is "configured to receive" various message types. *Boelter*, ¶[0025]. The system "first determines a current driving situation and then decides whether, when, and how to notify the driver," user input to figure out when a new message has arrived. *Id.*; *see also Boelter*, ¶[0028] ("The message received by the receiver unit 10 and the notification control signal determined by the detection unit 12 are forwarded to the notification unit 14 …."); EX1003, ¶99.

3. 1[b]: "analyzing without any input by the user a content of the received text message, message from the messenger application, or email;"

Boelter discloses 1[b]. EX1003, ¶¶100-103.

Boelter's notification unit automatically analyzes the content of received messages to determine sender identity, message type, and date/time received. *Boelter*, ¶¶[0028], [0036], Figs. 2, 4; EX1003, ¶101. This information may be displayed to the user at appropriate times. *Boelter*, ¶¶[0027]-[0029], [0036], [0038]; EX1003, ¶101. *Boelter*'s Figure 4 depicts a summary page listing sender or caller names, message types, and date/time received. *Boelter*, ¶[0036], Fig. 4.

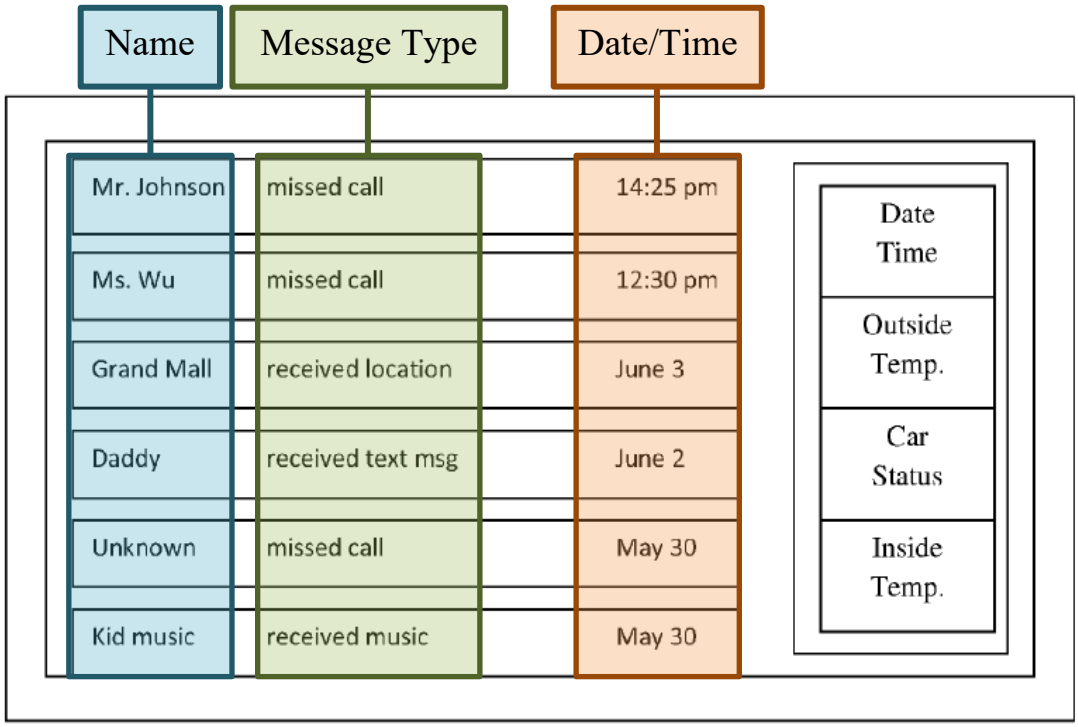


Fig. 4

Boelter, Fig. 4 (annotated). *Boelter*'s separation of message details content into different sender, body, and time parts is consistent with '775 claim 2, which recites that the "the step of analyzing … pars[es] the text message … into the sender and the body." Section VI.D.1; EX1003, ¶¶101, 142.

Boelter's analysis proceeds without user input, before messages are displayed to the user. EX1003, ¶102. Messages are analyzed "to determine which type of message shall be displayed, when the message is to be displayed, and in which format the message is displayed to a driver." *Boelter*, ¶[0028]. Thus, while a user may set prioritization rules, the analysis of incoming messages does not

involve user input. *Boelter*, ¶¶[0013], [0028]-[0031]; EX1003, ¶¶102-103 (distinguishing system configuration from the analysis of incoming messages).

4. 1[c]: "placing the received email, message from the messenger application, or text message in a queue to be read aloud in order of receipt time;"

Boelter and *Gruber* render obvious 1[c]. EX1003, ¶¶104-112.

Boelter discloses notifying users of various received message types—including email and text messages—based on receipt time. *Boelter*, ¶¶[0010], [0013], [0017], [0025], [0036], Fig. 4. During critical driving conditions, emails, text messages, and other notifications are queued for later display. These messages are provided to the driver "at an appropriate selected time or times" after the critical driving situation ends. *Boelter*, ¶¶[0013], [0027], [0029], [0036], Fig. 4

Boelter's Figure 4 illustrates a "summarized notification page" that displays this queue of missed calls, text messages, and notifications, organized by time and date. *Boelter*, ¶¶[0036], [0013], [0017], Fig. 4.

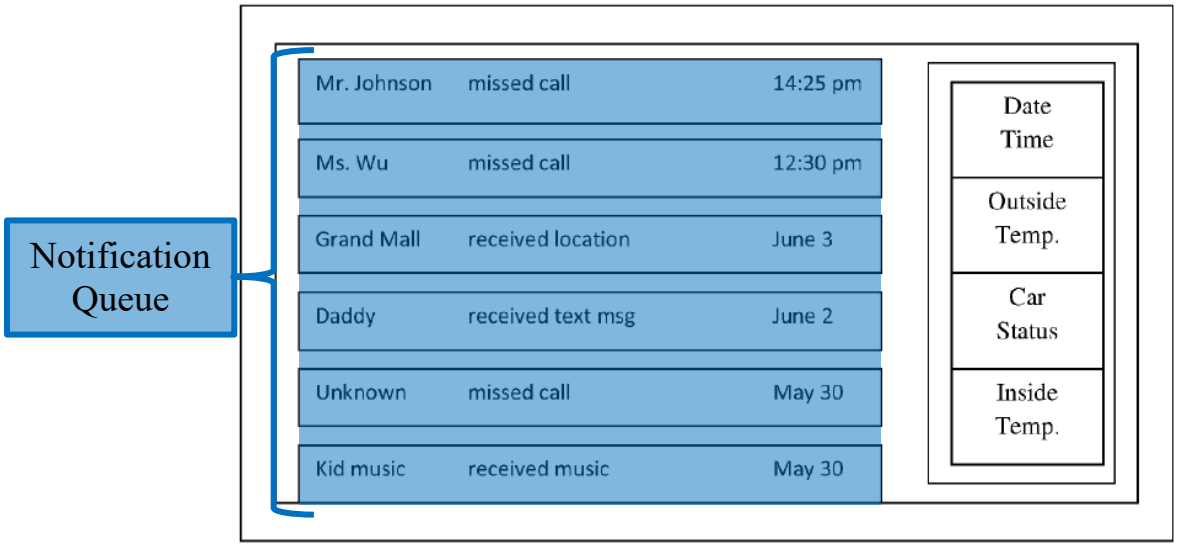


Fig. 4

Boelter, Fig. 4 (annotated). This screen is "used for informing a driver of incoming messages after the critical driving situation has passed." *Boelter*, ¶[0036]. It includes "any missed calls, unread notifications of other types, and messages of other types not previously notified." *Boelter*, ¶[0036], Fig. 4. The notifications in the queue are presented to the user, regardless of whether they are a new text message, new email, new missed call, or other notification. EX1003, ¶107. Because these notifications are organized by time/date, an ordinary artisan "would have understood that *Boelter's* notification page depicts a queue organized in order of receipt time." EX1003, ¶107; EX1014, 10:17-29.

An ordinary artisan would have been motivated to read *Boelter*'s messages aloud because *Gruber* describes reading lists of emails one after another. *Gruber*, ¶¶[0685]-[0686]; EX1003, ¶108. *Gruber* even expressly suggests reading a

"plurality of data items [] presented on the visual interface at the same time, … a subset at a time." *Gruber*, ¶[0696]. Reading *Boelter*'s messages as queued in Figure 4 would have decreased the user's need to click through or select messages on their own, improving message management and furthering *Boelter*'s goal of reducing distractions. EX1003, ¶108; *Boelter*, ¶[0036]. Moreover, because *Boelter*'s queue is a list of messages collected in one place, reading them would have been no different than reading a list of emails from one person as described in *Gruber*. *Gruber*, ¶[0555]; EX1003, ¶108. *Gruber* discloses hands-free list reading is a "core, cross-domain ability for users to be able to navigate results involving more than one item," providing further motivation. *Gruber*, ¶[0555].

Gruber would have made it obvious to read *Boelter*'s messages in order of receipt. EX1003, ¶¶109-112. *Gruber*'s assistant 1002 "read[s] content of text messages, email messages, and the like, and can provide options to the user in spoken form." *Gruber*, ¶[0187]. This includes reading lists of emails or text messages. *Gruber*, ¶¶[0467]-[0472] ("Read all Unread SMS Messages"), [0682]-[0689] (reading emails in a list). *Gruber* suggests reading its messages aloud in "sequential order" and sorting lists "by time." *Gruber*, ¶¶[0555], [0685]-[0689], [0699], [0705].

An ordinary artisan would also have been motivated to read aloud *Boelter*'s entire queue regardless of message type. EX1003, ¶¶110-111. *Boelter* teaches its

queue contains additional notification messages, such as calls, location data, and infotainment messages. *Boelter*, ¶¶[0025], [0036], Fig. 4. As with *Boelter*'s text-based messages, an ordinary artisan would have read these notifications aloud, as *Gruber* suggests. EX1003, ¶111. For example, to announce the first notification from *Boelter* Figure 4 (a missed call), the system would be programmed to say, "You have a missed call from Mr. Johnson, at 14:25 today." EX1003, ¶111 (citing *Gruber*, ¶¶[0013], [0559], [0564], [0598], [0736]). Thus, although message content may differ, an ordinary artisan would have implemented *Gruber*'s reading for all of *Boelter*'s notifications and messages. *Id.*

An ordinary artisan would also have had a reasonable expectation of success in reading aloud *Boelter*'s queue one message after another regardless of the new message's type because *Gruber* suggests reading different types of content as described above. EX1003, ¶112. Because *Gruber* teaches reading messages, an ordinary artisan would have had the skills to program a device to read aloud different types of content. EX1003, ¶¶47-52, 112 . Indeed, such programming was well established and easily implemented. EX1003, ¶¶89, 112; EX1009; EX1018, 1-7, EX1016, 79-80; *Polak*, ¶[0019]; EX1008, ¶[0018]; *Gruber*, ¶¶[0056]-[0057], [0736], [00744]. Additional motivations to combine and reasonable expectations of success appear in Section VI.B.

5. 1[d]: "informing the user without any input by the user that the received text message, message from the messenger application, or email has been received from a sender by reading aloud at least an identity of the sender;"

Boelter and *Gruber* (Ground 1A) or *Boelter*, *Gruber*, and *Polak* (Ground 1B) render obvious 1[d]. EX1003, ¶¶113-121.

Boelter teaches notifying (i.e., informing) users of messages' arrival without user input. *Boelter*, ¶¶[0010], [0013], [0017], [0025], [0036]. *Boelter*'s Figure 4, reproduced below, illustrates a "summarized notification page" including a queue of missed calls, text messages, and notifications, identified by sender name, message type, and time/date received. *Boelter*, ¶¶[0036], [0013], [0017], Fig. 4.

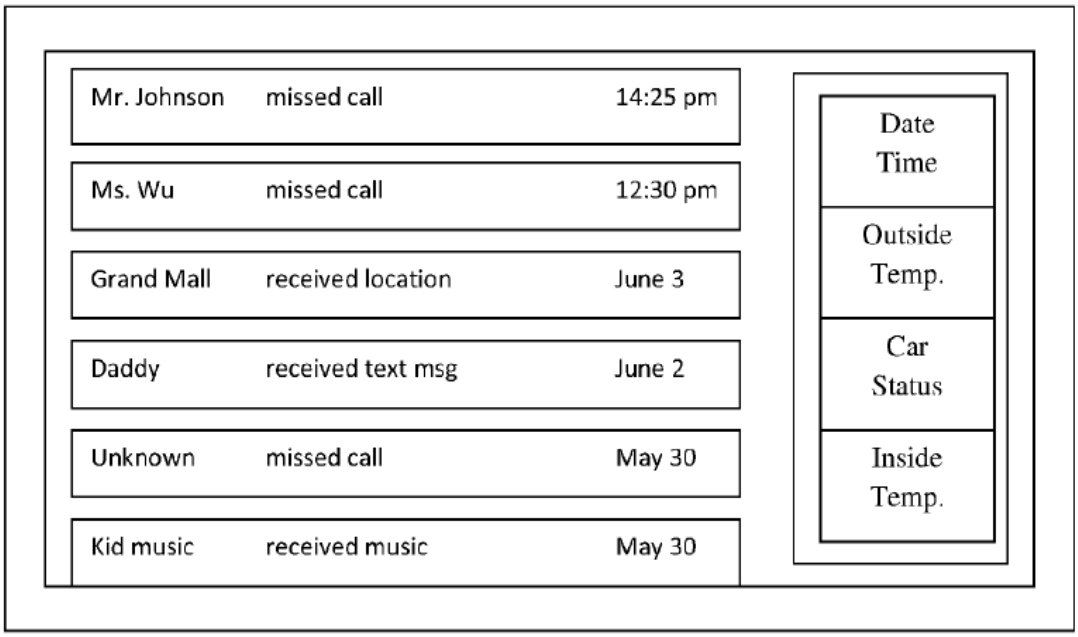


Fig. 4

Boelter, Fig. 4. This screen is "used for informing a driver of incoming messages after the critical driving situation has passed." *Boelter*, ¶[0036]. It includes "any missed calls, unread notifications of other types, and messages of other types not previously notified." *Boelter*, ¶[0036], Fig. 4.

Boelter's summarized notification page may include emails and text messages. Indeed, *Boelter* discloses an embodiment where email and text messages are queued during critical driving conditions, but "after the critical driving situation ended, the driver would be notified of all newly arrived emails and text messages" at an appropriate selected time. EX1003, ¶115; *Boelter*, ¶¶[0027], [0029], [0036].

Driving conditions, not user input, dictate when the user is notified of incoming messages. *Boelter*, ¶[0029]; EX1003, ¶¶114-116.

a. Ground 1A

Gruber would have made it obvious to read at least sender names (identities) aloud to complete *Boelter*'s notification, which occurs without any user input. EX1003, ¶117. *Gruber* teaches reading lists of new text messages or emails. *Gruber*, ¶¶[0270] (task flow may include "reading aloud each text message individually, and pausing after each message to allow the user to provide a spoken command"), [0683] (assistant providing paraphrase of each email in a list followed by each message's unbounded content), [0409]-[0419], [0452]-[0458] (SMS use

cases). For example, *Gruber*'s assistant 1002 may announce the sender of an email message from the list—"Next message, From Harry Saddler, …." *Gruber*, ¶[0688]. After a pause, it then reads the message body. *Id.* Applying a similar approach to *Boelter*'s notifications—identifying sender identity, message type, and time/date; pausing; then reading additional message contents—would have been obvious. EX1003, ¶¶117-118.

An ordinary artisan would have been motivated to combine the asserted references with a reasonable expectation of success. *See* Section VI.B; EX1003, ¶¶117-118.

b. Ground 1B

To the extent Patent Owner disputes Ground 1A, *Polak* would have further made it obvious to read at least sender names (identities) aloud without any user input. EX1003, ¶119. *Polak* teaches that upon arrival of a message, "the sender details such as, for example, name and/or phone number may be sounded to the user first…." *Polak*, ¶[0033]; EX1008, ¶[0030]. An ordinary artisan would have understood this occurs without user input because a user does not need to engage with the application to hear the message notification and sender details. EX1003, ¶119.

An ordinary artisan would have been motivated to implement *Polak*'s automatic reading in *Boelter*'s system, as modified by *Gruber*, because announcing

the arrival of a message (after a critical driving situation has passed) without requiring the user to select the message would remove one decision a user has to make. EX1003, ¶120. An ordinary artisan would also have been motivated to make such a change because *Polak* teaches its process will "enable drivers to drive safely while paying attention to the road, without taking their eyes off the road…." *Polak*, ¶[0021]; EX1008, ¶[0019]; EX1003, ¶120.

An ordinary artisan would have had a reasonable expectation of success in the combination because it would only require simple and routine programming changes. *See* Section VI.B; EX1003, ¶120.

6. 1[e]: "allowing the user a time to take an affirmative action to stop reading aloud to the user a body of the received text message, message from the messenger application, or email;"

Boelter and *Gruber* render obvious 1[e]. EX1003, ¶¶122-128.

Boelter discloses the use of affirmative action limited to a gesture performed on the touchscreen. EX1003, ¶123. *Boelter* teaches a touchscreen that "provides a human machine interaction interface." *Boelter*, ¶[0030]. "[D]efined finger movements" on this touchscreen allow for user-system interaction, including the rejection of incoming messages. *Boelter*, ¶¶[0033]-[0034]. In some embodiments, the user touches a particular portion of *Boelter*'s screen (e.g., a predefined third or quarter of the screen) corresponding to different commands. *Boelter*, ¶¶[0032]-

[0033]. For example, touching the top left portion of the screen in *Boelter*'s Figure 3 would reject a call, touching the top right portion would accept the call, and touching the bottom portion would reject the call and send an automatic reply:

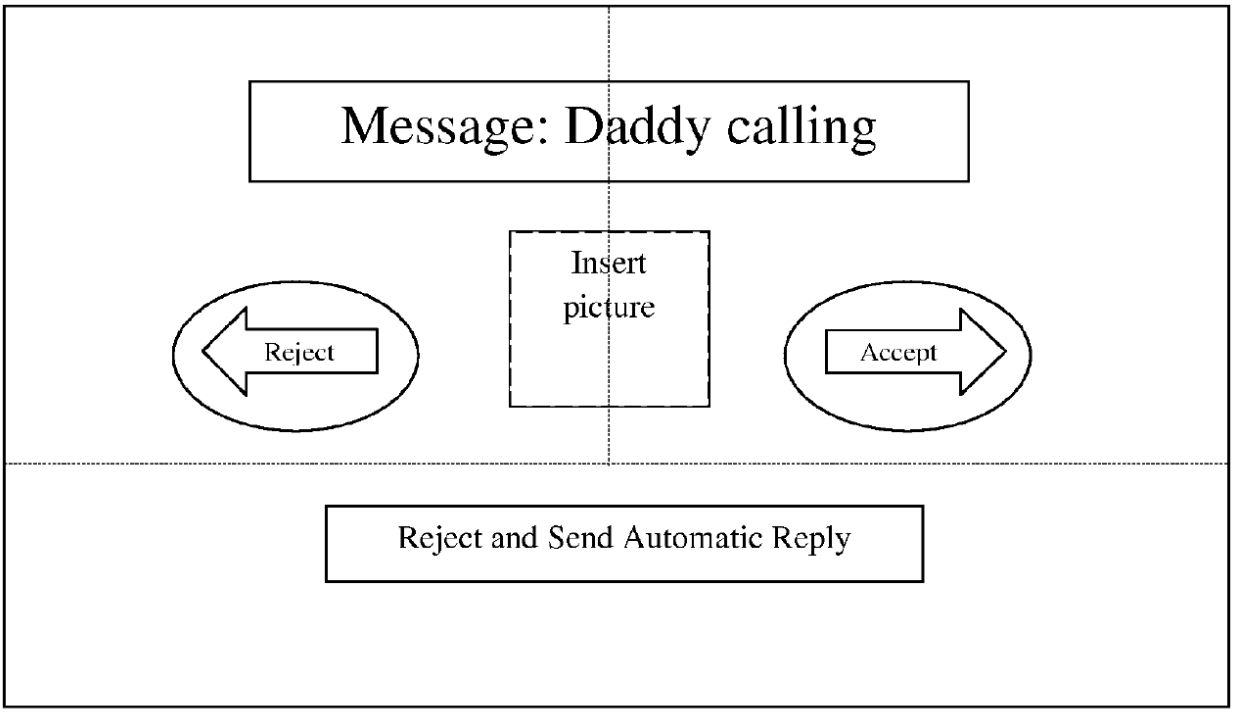


Fig. 3

Boelter, ¶¶[0032]-[0033], Fig. 3.

Boelter also teaches that "defined finger movements, such as right and left sweep, one-finger tap or two-finger tap, anywhere within the screen surface can be associated with certain input control signals." *Boelter*, ¶[0034]. In this way, the system's screen "may be configured such that an intuitive finger gesture allows a driver to interact with the notification unit without the driver having even to look at

the screen, and hence without requiring much attention from the driver." *Id.*; EX1003, ¶124.

Gruber discloses allowing the user time to take an affirmative action to stop reading the body of the messages. EX1003, ¶125. For example, *Gruber* teaches reading an "item-specific paraphrase of the list of emails one by one," briefly pausing, then reading the "unbounded content of the message" (message body). *Gruber*, ¶¶[0683], [0685]-[0686]. *Gruber* explains this pause is "for user interruption"—it allows the user to stop the reading of the message's body. *Gruber*, ¶¶[0684], [0686]; EX1003, ¶125.

An ordinary artisan would have found it obvious to include *Gruber*'s pause between reading aloud the sender identity and body of the message as part of the implementation of reading aloud *Boelter*'s notifications because doing so would give the user an opportunity to decide whether they wanted to hear the message now or later. EX1003, ¶126. An ordinary artisan would have been motivated to pause because not all messages should be read aloud in every scenario. EX1003, ¶125. For example, a user may not wish to have their private messages read aloud if others are in the vehicle with them. EX1003, ¶125. An ordinary artisan would have been motivated with a reasonable expectation of success to include the feature for the reasons discussed above. *See* Sections VI.B, VI.C.4 (1[c]); EX1003, ¶125.

An ordinary artisan would have been motivated to maintain *Boelter*'s predefined tap or swipe as a response feature for interacting with notifications because *Gruber* describes the use of "gestures on a touch-sensitive surface or display" as responsive input to list reading. *Gruber*, ¶[0562]; EX1003, ¶127. This input may be made without the driver even looking at the screen, and with minimal distraction. *Boelter*, ¶[0034]. Following the reading of a message notification with this no-look physical response would provide a way for users to easily work their way through a series of new incoming messages. EX1003, ¶127. It would also decrease the frustrations with audio input, which were understood to be less reliable than tactile responses. EX1003, ¶127.

An ordinary artisan would have had a reasonable expectation of success in maintaining *Boelter*'s predefined tap or swipe because gesture-based responses were common and routine features of touchscreen devices. EX1003, ¶128. Gesture-based responses were compatible with and complementary to read-aloud systems. *See* EX1003, ¶¶84, 128 (citing *Gruber*, ¶[0562]; *Polak*, ¶[0032]; EX1008, ¶[0029]). Pausing between identifying the sender and reading the body of a message, and allowing the user to prevent the message from being presented, had been successfully implemented in similar systems. EX1003, ¶128 (citing *Polak*, ¶¶[0030] [0032]; EX1008, ¶¶[0027], [0029]). Accordingly, such a combination would have yielded predictable results. EX1003, ¶128.

7. 1[f]: "based on not receiving an input corresponding to the affirmative action from the user, reading aloud the body of the received text message, message from the messenger application, or email."

Boelter and *Gruber* render obvious 1[f]. EX1003, ¶¶129-135.

As illustrated in *Boelter*'s Figure 4, *Boelter* teaches notifying users of messages' arrival with a list of messages identified by sender name, message type, and time/date received. *Boelter*, ¶¶[0036], [0013], [0017], Fig. 4. This allows the user to decide whether to "select, view, delete, and/or respond to individual messages." *Boelter*, ¶[0036].

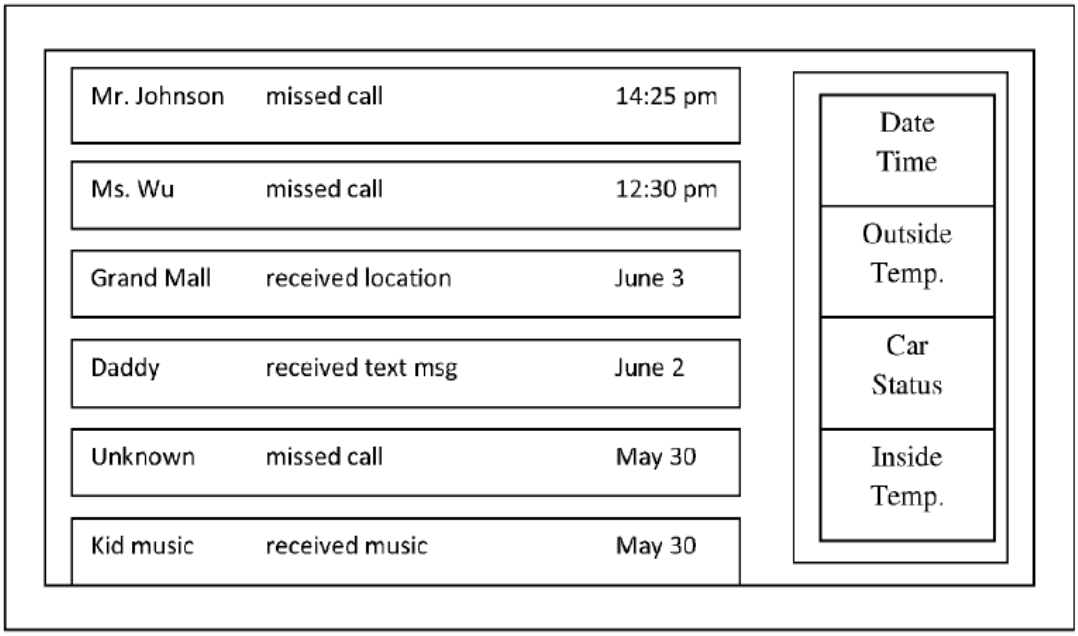


Fig. 4

Boelter, Fig. 4.

Gruber would have made it obvious to read the message body aloud after providing time for user instructions. *Gruber* teaches announcing each email in a list, pausing, then reading the entirety of the message's unbounded content (i.e., the message body). *Gruber*, ¶¶[0683], [0685]-[0686]. For example, *Gruber*'s assistant 1002 may announce an email—"Next message, From Harry Saddler, 30 minutes ago…." *Gruber*, ¶[0688]. If the user takes no action, *Gruber*'s system proceeds to read the message body—"Harry wrote: …." *Id.* Applying a similar approach to *Boelter*'s notifications would notify the user of the message (reading sender's name, message type, and time/date), allowing time for a user action (delete, reply, or view/hear the message (*see Boelter*, ¶[0036], Fig. 4)), then read the full message body ("Harry said: …."). EX1003, ¶131.

Gruber's assistant provides email subjects as part of reading emails aloud. *Gruber*, ¶¶[0686]-[0688]. *Boelter*'s "summarized notification page" includes sender name, message type, and time/date, but allows the user to "view" the full message. *Boelter*, ¶[0036], Fig. 4. An ordinary artisan would have understood email subjects could be read with the user's name (informing the user of the message's arrival), with the message body, or both because *Gruber* teaches that "[t]he domain-specific speakable text templates arrange the different data fields of a domain-specific item type in a suitable order…." *Gruber*, ¶[0559]; EX1003, ¶132.

An ordinary artisan would further have been motivated to program the system to read *Boelter*'s messages aloud as disclosed by *Gruber* because this audible output would have reduced the user's need to read from the touchscreen. EX1003, ¶133. This would further *Boelter*'s goal of providing a "convenient human machine interface" that reduces distractions. *Boelter*, ¶¶[0015], [0034]. An ordinary artisan would have had a reasonable expectation of success in the combination because read-aloud features were already implemented in safe-messaging technology. EX1003, ¶133 (citing *Polak*, ¶[0030]; EX1008, ¶[0027]). Additional motivations to combine and reasonable expectations of success appear in Sections VI.B and VI.C.4-VI.C.6.

D. Dependent Claims

1. Claim 2: "The computer implemented method of claim 1, wherein analyzing the content comprises parsing the email into a subject, the sender, and the body, and parsing the text message or the message from the messenger application into the sender and the body."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 2 for at least the reasons discussed in Section VI.C.3 (1[b]). EX1003, ¶¶101-103, 134-136. *Boelter* divides incoming messages into sender identity, message type, and date/time received (displayed on the "summary screen") and the body (not displayed on the "summary screen"). *Boelter*, ¶¶[0028], [0036], Figs. 2, 4; EX1003, ¶134.

To the extent Patent Owner disputes that *Boelter* discloses claim 2, the asserted references render it obvious. EX1003, ¶¶135-136. *Gruber* discloses dividing messages into parts for reading to a user. *Gruber*, ¶¶[0680], [0682]. For example, when analyzing text messages, *Gruber* teaches separating the notification into announcing the sender's identity and then reading the body of the message. *Gruber*, ¶¶[0409]-[0416] (" 'Message from Mary Richards***<change of voice>*** are you free for dinner tonight?'" (emphasis added)). *Gruber* further teaches dividing emails into subparts including at least sender, subject, and body. *Gruber*, ¶¶[0682]-[0686]. Thus, *Gruber* discloses parsing emails into the subject, the sender, and the body, and other messages into the sender and the body. EX1003, ¶135.

An ordinary artisan would have found it obvious to implement *Gruber*'s parsing into *Boelter*'s notification method. EX1003, ¶136. *Gruber* demonstrates efficiency benefits of separating the sender and subject from the body, such as allowing the user to search or skip messages based on these attributes. *Gruber*, ¶¶[0682]-[0683]. An ordinary artisan would have been motivated to conserve time and resources by providing the user with pieces of information, making it easier for the user to hear and understand the message. EX1003, ¶136. Parsing messages into subparts was one of a few known, predictable options for improving efficiency—messages were already coded with subpart information, others used the approach, and implementing the approach would not require new technology. EX1003, ¶136.

An ordinary artisan would have had a reasonable expectation of success implementing this approach because *Boelter* already suggests its system is configured to separate messages into parts. EX1003, ¶136. Additional motivations to combine and reasonable expectations of success appear in Section VI.B.

2. Claim 3: "The computer implemented method of claim 1, wherein the computer implemented method is carried out with an application that is configured to run in a foreground or in a background, and is further configured to switch back and forth between the foreground and the background, and is further configured to read aloud the identity of the sender, and the body of the received email, text message, or message from the messenger application while the application is in the background."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 3. EX1003, ¶¶137-142.

Boelter's method is carried out on an application "implemented in software or firmware" and displayed on a touchscreen. EX1003, ¶138; Section VI.C.1, VI.C.3, VI.C.4.

Gruber discloses its virtual assistant application switches between a device's foreground and background. *Gruber*, ¶¶[0104], [0154], [0182]; EX1003, ¶139. This system may detect a hands-free context "while a digital assistant application is not being executed in the foreground of an operating system, or is not displaying a graphical user interface on the device." *Gruber*, ¶[0182]. *Gruber* further teaches reading messages aloud when virtual assistant 1002 is not in the foreground as "the

present invention can be implemented without any visual display on the screen of device 60 or elsewhere." *Gruber*, ¶[0189]; *see also Gruber*, ¶[0191] ("[D]isplay of the spoken output is not required."). An ordinary artisan would have concluded that *Gruber*'s application is configured to run in and switch between the foreground/background because it reads the messages in both states. EX1003, ¶139.

Boelter calls for a "convenient human machine interface which … may use a touch screen which provides different formats for notifying a driver." *Boelter*, ¶[0015]. For example, different screens include "a message accept/reject screen, a message blocked screen, and a message summary screen." *Id. Boelter* also suggests formats where "substantially the whole screen can be used for a swiping/touching action so that a driver does not need to locate individual touch-screen buttons." *Id.*; *see also id.*, ¶¶[0016], [0036]. *Boelter* determines which screen to display to the user based on sensor data deduced from different situations (e.g., driving or in a parked car). *Boelter*, ¶¶[0027]-[0028]. *Gruber* teaches that "when a hands-free context is determined (e.g.[,] driving a vehicle), device 60 limits the ability to view visual output presented by the electronic device." *Gruber*, ¶[0272]. This may include "limiting the ability to view visual output presented by the electronic device (for example, deactivating a screen/visual output mode …)." *Gruber*, ¶[0273]; *see also Gruber*, ¶¶[0210], [0219]-[0220] ("preventing [] the display of a

subset of visual outputs"), [0562] ("fully responsive to user input from multiple input channels…"). Consistent with these disclosures, an ordinary artisan would have been motivated to include *Gruber*'s foreground/background switching and background message reading in the asserted combinations' systems because it would have reduced distractions. *See* Section VI.C.7; EX1003, ¶140. An ordinary artisan would have implemented this combination by programming the system to read the messages in the background followed by a display that allows the user to input gestures. EX1003, ¶¶138, 140 (explaining this is consistent with *Boelter*'s screen switching); *Boelter*, ¶¶[0012], [0015], [0027]-[0031], [0036], [0038].

An ordinary artisan would have been motivated to implement *Gruber*'s reading-aloud steps in *Boelter* and a reasonable expectation of success as discussed in 1[d]-1[f]. *See* Sections VI.B, VI.C.5-VI.C.7; EX1003, ¶141. An ordinary artisan would also have been motivated to configure *Boelter*'s system to read messages aloud while the application is in the background because *Boelter* suggests notifications "that require[] minimal attention from the driver" are safer and less distracting. *Boelter*, ¶¶[0015], [0035]; EX1003, ¶141. And an ordinary artisan would have understood providing notifications while an application is in the background would display less text on the screen, providing fewer distractions. EX1003, ¶141; *Boelter*, ¶¶[0035], [0038]. This would also allow users to hear

messages while another application, such as a navigation screen, is being displayed. EX1003, ¶141.

An ordinary artisan would have had a reasonable expectation of success in such a combination because applications that read messages aloud while in the background were common. EX1003, ¶142 (EX1019, 2-3). An ordinary artisan would have expected the combination to be reasonably successful because it would have not required physical changes to *Boelter*'s system, but merely minor tweaks to the programming within an ordinary artisan's skill. EX1003, ¶142; *see also* Section VI.B.

3. Claim 4: "The computer implemented method of claim 1, further comprising prompting the user to call the sender without any input from the user."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 4. EX1003, ¶¶143-148.

Boelter's system is configured to receive, send, and respond to various types of messages, including phone calls, text messages, and emails. *Boelter*, ¶¶[0025], [0028]. *Boelter*'s Figure 3 discloses that drivers may be automatically presented with buttons, allowing the user to accept an incoming call, reject it, or send a responsive message. *Boelter*, ¶[0032], Fig. 3. Through these teachings, *Boelter* suggests treating text-based messages and calls in the same notification queue and

automatically prompting users to respond. EX1003, ¶144; *Boelter*, ¶¶[0032], [0036], Figs. 3-4.

Gruber similarly discloses providing a user with various action options after a message or list is read. *Gruber*, ¶¶[0686] ("a brief pause is provided, so that the user can enter a command for an action (e.g., reply, delete, repeat, etc.) ….") [0597] (prompting the user "would you like to call or get directions?"). It also discloses one-button calling. For example, *Gruber*'s system may prompt the user to make a call when the system senses the user is driving and reminding the user to call someone. *Gruber*, ¶¶[0395]-[0405], [0497]-[0506]. The system may also be configured to allow users the option of calling by contact name or phone number at any time. *Gruber*, ¶[0311]. This suggests prompting telephone calls as an option for responding to a variety of messages. EX1003, ¶145; *Gruber*, ¶¶[0311], [0339]-[0373], [0395]-[0405], [0686], [0597].

An ordinary artisan would have found it obvious to modify *Boelter*'s notification method to prompt a user to call the sender without any input from the user based on *Gruber*'s message reply and calling options. EX1003, ¶146. An ordinary artisan would have been motivated to implement such a feature into *Boelter*'s notification method because *Boelter* already contemplates using its system for calls in addition to messages. EX1003, ¶146. Providing an option to call a sender would have reduced the user's interaction with a screen or the need to

draft a reply message, which would have supported *Boelter*'s goals of creating a safer messaging system. EX1003, ¶146. Adding a prompt to call the sender would not have required new technology because such a prompt already existed in the art. EX1003, ¶146; *see also* Section VI.B.

An ordinary artisan would have also known there are limited next steps to consider after a message is read aloud, including prompting the user to take an action, waiting for the user to input a command, or ending the interaction with the message. EX1003, ¶147 (citing *Polak*, ¶¶[0033]-[0036]; EX1008, ¶¶[0030]-[0033]; EX1015, ¶¶[0024]-[0027], [0040], [0054]-[0058], [0070], [0072]). There are also limited ways for the user to interact with each message, including responding with another message, responding with a call, repeating the message, or taking no further action (including deleting/dismissing the message). *Id.* It would have made sense to prompt the user to select one or more of these options because providing the user with limited options reduces the user's cognitive load while engaged in a distracting activity. *Gruber*, ¶¶[0554], [0561], [0567], [0710]; EX1003, ¶147. In other words, prompting the user to call the sender without input would have been an obvious design choice.

An ordinary artisan would have had a reasonable expectation of success in such a combination because prompting users with response options was routine and well known. EX1003, ¶148. Just as *Boelter* prompts users to respond to a call

with a text message, it would have been just as simple to prompt a phone call in response to an incoming text message or email because the program steps to place a call were not new. EX1003, ¶148.

4. Claim 5: "The computer implemented method of claim 1, wherein a different interface is presented on the touch screen depending on an activity of the user."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 5. EX1003, ¶¶149-155. *Boelter* teaches varying screen configurations for different activities—it provides different interfaces for normal driving conditions and for critical driving conditions. *Boelter*, ¶¶[0013], [0030]-[0035].

Boelter's different driving conditions are different a activities because they require different levels of user attention. *Boelter*, ¶[0029] (limiting certain notifications until "[a]fter a critical [driving] situation has passed"); EX1003, ¶150. *Boelter's* detection unit differentiates between driving situations using vehicle speed and acceleration, traffic conditions, driver fatigue, and other factors. *Boelter*, ¶¶[0011], [0012], [0026]. This accords with the '775 patent, which does not restrict the definition of an "activity." EX1003, ¶150. It merely provides examples, such as driving "a vehicle," "working with a desktop … computer," and "exercise." EX1001, 5:8-18.

To the extent Patent Owner disputes that *Boelter* discloses claim 5, *Gruber* the asserted combinations render claim 5 obvious. *Gruber* discloses a virtual

assistant that provides "hands-free" operation when users are engaged in activities, like driving or walking, that limit their ability to interact with a device. *Gruber*, ¶¶[0007]-[0009], [0168]. *Gruber* also uses device sensors to determine if the user is engaged in activities for which reading aloud text would be useful. *Gruber*, ¶¶[0116], [0170]-[0181], [0662], [0725].

Gruber discloses adjusting the user interface limits distraction, while allowing user access to messages and alerts when the user is driving or engaged in other activities. *Gruber*, ¶¶[0008]-[0009]. The system "adapts various stages of a complex interactive system to modify the user experience to reflect the particular limitations of such a [hands-free] context." *Gruber*, ¶[0009]. *Gruber* suggests mechanisms for accepting and processing speech input that may reduce the need for the user to interact with a display screen and/or to use a touch interface. *Gruber*, ¶[0166]. In some situations, the system may be implemented without displaying content on the device's screen. *Gruber*, ¶¶[0189], [0680], [0682] (preventing display of email objects and attachments in eyes-free mode). These disclosures would have motivated an ordinary artisan to present different interfaces during select activities "to simplify the user experience, limit distractions, and improve safety." during activities (like *Boelter*'s critical driving situations) requiring more user attention than other activities (like sitting in *a* parked car). EX1003, ¶¶152-153.

By November 2014, ordinary artisans would have been familiar with application design and graphical user interfaces for use in vehicles, mobile phones, and a variety of consumer electronic devices. EX1003, ¶154. They would understand methods for using device sensors to determine user activity, as described in *Boelter*, ¶¶[0026]-[0027], and *Gruber*, ¶¶[0169]-[0183]. EX1003, ¶154. Ordinary artisans would also be familiar with the creation of graphical user interfaces for display on devices, allowing them to configure an application to display less information based on user activity. EX1003, ¶154; *see also Boelter*, ¶[0028] (suggesting look-up table for modifying display contents).

An ordinary artisan would have been motivated to combine the asserted references with a reasonable expectation of success. *See* Section VI.B; EX1003, ¶154.

5. Claim 6: "The computer implemented method of claim 5, wherein the activity is driving a vehicle."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 6. *Boelter* discloses that its notification method is "installed [i.e., carried out] in a vehicle" and utilized during driving. *Boelter*, ¶[0024]; *see also* Sections VI.C, VI.D.4; *Boelter*, ¶¶[0009]-[0010], [0018]; EX1003, ¶¶113-121, 156. As discussed in claim 5, *Boelter* discloses or the asserted combinations render obvious displaying a different interface when the user is engaged in different driving

situations. *See* Section VI.D.4 (claim 5). The ordinary artisan would have had motivation to combine *Boelter* and *Gruber*, as well as a reasonable expectation of success. *See* Section VI.B; VI.D.4; EX1003, ¶156.

6. Claim 7: "The computer implemented method of claim 5, wherein the activity is working with a desktop, a laptop, or a tablet computer."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 7. EX1003, ¶¶157-159. *Boelter* discloses its notification method may be integrated "in a mobile communication device such as a smartphone." *Boelter*, ¶[0024]. *Gruber* similarly discloses its assistant may be installed on devices other than a phone or vehicle. *Gruber*, ¶[0245]. For example, "assistant 1002 is installed on device 60 such as a mobile computing device, … , laptop, tablet computer, consumer electronic device, music player, or the like." *Id.*

An ordinary artisan would have found it obvious to implement *Boelter*'s notification system for activities beyond driving because reducing distractions would have been beneficial while working on other devices, like a laptop. EX1003, ¶158 (citing EX1018, 7); *Gruber*, ¶[0011]. Indeed, collecting notifications in a queue and having them read aloud while working would have reduced the amount of effort a user would have needed to exert to manage multiple devices or messaging applications. EX1003, ¶158. And *Boelter* already contemplates the use of its system on a mobile device, motivating its use beyond driving; EX1003, ¶158.

Boelter, ¶[0024]. An ordinary artisan would have had a reasonable expectation of success in such an application because *Gruber* suggests that virtual assistants would be used in various devices and is agnostic as to the activity. EX1003, ¶158; *see* Section VI.B.

An ordinary artisan would have found it obvious to display a different interface on the device when the user is working with a desktop, a laptop, or a tablet computer. EX1003, ¶159; *Gruber*, ¶[0011]. An ordinary artisan would have been motivated to create a different interface for when the user is working with a computer because the activity presents fewer safety concerns. EX1003, ¶159. As such, an ordinary artisan would have implemented an interface that is less restricted compared to when the user is driving. EX1003, ¶159; *Gruber*, ¶[0011] ("the same interface can be made adaptable to both an office environment and a moving vehicle, *with the system dynamically making the necessary changes to user interface behavior as the environment changes*." (emphasis added)); *see also Gruber*, ¶¶[0013]-[0015], [0166]. An ordinary artisan would have had a reasonable expectation of success as described in claim 5. EX1003, ¶159; *see* Section VI.D.4.

7. Claim 8: "The computer implemented method of claim 5, wherein the activity is exercise (Active Mode)."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 8. EX1003, ¶¶160-163; Section VI.D.4.

An ordinary artisan would have found it obvious to utilize the asserted combinations' notification system and method while a user is exercising. EX1003, ¶161. For example, *Gruber* describes its hands-free assistant may be triggered "when the assistant detects that the user is performing an action (e.g., driving, or walking)." *Gruber*, ¶¶[0662], [0725]. *Gruber* further describes that this determination can be made based on how fast a user is moving. *Gruber*, ¶[0183]. Based on these disclosures, an ordinary artisan would have understood that *Gruber* suggests reading notifications and messages aloud during exercise, such as walking. EX1003, ¶161; *Gruber*, ¶¶[0116], [0170]-[0181], [0662], [0725].

An ordinary artisan would have been motivated to use the asserted combinations' notification system and method during exercise because the user would benefit from reduced distractions while performing the activity. EX1003, ¶162; *Gruber*, ¶[0013] (detecting hands-free context and adjusting operation when receiving input and providing output). Looking at a screen is not conducive to these activities where the user is moving. EX1003, ¶162. Indeed, an ordinary artisan would have known that exercise requires heightened attention because users may be on exercise equipment (e.g., a treadmill) or navigating through streets and trails. EX1003, ¶162. An ordinary artisan would have had a reasonable expectation of success in implementing the asserted combinations during exercise because doing so would have been no more than implementing *Boelter*'s assessment of

speed to determine a driving condition as an assessment of speed related to exercise. EX1003, ¶162. Moreover, an ordinary artisan would have understood that this combination would have not required substantial system changes to implement it outside of a vehicle, as *Boelter* discloses it can be implemented on a phone. *Boelter*, ¶[0024]; EX1003, ¶162.

An ordinary artisan would have found it obvious to display a different interface on the device when a user is exercising. EX1003, ¶163. An ordinary artisan would have been motivated to create a different interface for when a user is exercising because the user likely has limited access to their device. EX1003, ¶163. As such, an ordinary artisan would have implemented an interface that 1) requires simple input to prevent the user from having to stop exercise to interact with their device and 2) allows the device to distinguish between intentional input and unintentional input (e.g., a pocket dial). EX1003, ¶163; *Gruber*, ¶¶[0013] (virtual assistant can "adjust its operation accordingly when receiving input, providing output, engaging in dialog with the user, and/or performing (or initiating) actions."), [0166]. An ordinary artisan would have had a reasonable expectation of success as described in claim 5. EX1003, ¶163; *see* Section VI.D.4; VI.B.

8. Claim 9: "The computer implemented method of claim 1, wherein informing the user comprises reading aloud a name of the sender."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 9 for at least the reasons described in Section VI.C.5 (1[d]). EX1003, ¶¶113-121, 164.

9. Claim 10: "The computer implemented method of claim 1, the method further comprising based on receiving the input corresponding to the affirmative action from the user, not reading aloud the body of the received text message, message from the messenger application, or email to the user."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 10 for at least the reasons described in Sections VI.C.6-VI.C.7 (1[e]-1[f]). EX1003, ¶¶122-135, 165. As discussed above, claim limitations 1[e]-1[f] describe providing the user time to stop the reading of the body of a message. An ordinary artisan would have understood that claim 10 merely clarifies that the input of the affirmative action causes the message to not be read aloud. EX1003, ¶165. *Gruber* discloses allowing the user time to take an affirmative action to stop reading the body of the messages. *Gruber*, ¶¶[0683]-[0686]; Sections VI.C.6-VI.C.7; EX1003, ¶165. An ordinary artisan would have been motivated with a reasonable expectation of success to include *Gruber*'s message flow, which includes the

ability to stop or prevent the reading of messages, into *Boelter* for the same reasons described in Sections VI.C.6-VI.C.7 (1[e]-1[f]) and VI.B. EX1003, ¶165.

10. Claim 11: "The computer implemented method of claim 1, further comprising receiving the input from the user without the user looking at the touch screen."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 11. EX1003, ¶166. As discussed in 1[e] above, *Boelter* discloses an "intuitive finger gesture [that] allows a driver to interact with the notification unit without the driver having even to look at the screen, and hence without requiring much attention from the driver." *Boelter*, ¶[0034]; *see also Boelter*, ¶¶[0031]-[0033], [0038]; Section VI.C.6 (1[e]).

11. Claim 12: "The computer implemented method of claim 1, further comprising displaying an interface on the touch screen with no prompts for the user to touch."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 12. EX1003, ¶¶167-170.

Gruber teaches a "hands-free context" that can "limit the tasks or functionalities that are allowed at a given time." *Gruber*, ¶[0272]. "In some embodiments, when a hands-free context is determined (e.g.[,] driving a vehicle), device 60 limits the ability to view visual output presented by the electronic device." *Id.* This feature may include "limiting the ability to interact with a

graphical user interface presented by the electronic device (for example, limiting a device so as to not request touch input from the user ….” *Gruber*, ¶[0274].

As discussed in 1[e] above, *Boelter* discloses an “intuitive finger gesture [that] allows a driver to interact with the notification unit ….” *Boelter*, ¶[0034]; *see also Boelter*, ¶¶[0031]-[0033], [0038]; Section VI.C.6 (1[e]). In some embodiments, the system will understand predetermined gestures made ***anywhere*** on *Boelter*’s touchscreen—the driver may “interact with the notification unit without the driver having even to look at the screen.” *Boelter*, ¶[0034].

These teachings would have suggested providing a display interface without buttons or other prompts for the user to touch because an ordinary artisan would have preferred to provide displays that decreased driver distraction. EX1003, ¶168. They also would have motivated an ordinary artisan to limit the display of requests for touch input in *Boelter*’s notification system because doing so would have decreased distractions. EX1003, ¶169.

An ordinary artisan would have been motivated to combine the asserted references with a reasonable expectation of success. *See* Section VI.B; Section VI.C.6 (1[e]); EX1003, ¶170.

12. Claim 13: "The computer implemented method of claim 1, further comprising receiving the input from the user based on gestures by the user."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 13. EX1003, ¶¶171-172.

As discussed in 1[e] above, *Boelter* discloses an "intuitive finger gesture [that] allows a driver to interact with the notification unit without the driver having even to look at the screen." *Boelter*, ¶[0034]; *see also Boelter*, ¶¶[0031]-[0033], [0038]; Section VI.C.6 (1[e]). These gestures include "defined finger movements, such as right and left sweep, one-finger tap or two-finger tap, anywhere within the screen …." *Boelter*, ¶[0034]. An ordinary artisan would have understood *Boelter*'s gestures were user input, used to interact with the system by deleting or responding to messages. *Boelter*, ¶¶[0033], [0036]; EX1003, ¶172; *see also* Sections VI.B, VI.C.6 (1[e]).

13. Claim 14: "The computer implemented method of claim 13, wherein the input from the user is limited to tapping or swiping the touch screen without a need to look at the touch screen."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 14 for the reasons described in Sections VI.C.6, VI.D.11, and VI.D.12 (1[e], claims 12, 13 describes controls that are limited to "swiping or tapping" that do not require the user to look at the screen. *Boelter*, claim 18, ¶¶[0033]-[0034]; EX1003,

¶173. An ordinary artisan would have been motivated to combine the asserted references with a reasonable expectation of success. *See* Section VI.B; Section VI.C.6 (1[e]); EX1003, ¶173.

14. Claim 15: "The computer implemented method of claim 1, further comprising informing the user by voice of an option to call the sender."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 15 for the reasons discussed in claim 4. *See* Section VI.D.3 (claim 4); EX1003, ¶¶143-148, 174-175.

Gruber's virtual assistant provides a "speech-based prompt" with a question such as "would you like to call or get directions?" which audibly informs the user of an option to a number corresponding to a data item. *Gruber*, ¶[0704]; EX1003, ¶175. As discussed above in claim 4, an ordinary artisan would have found it obvious to modify *Boelter*'s notification method to include a prompt to call the sender without any input from the user based on *Gruber*'s message reply and calling options. *See* Section VI.D.3 (claim 4); EX1003, ¶175. Based on *Gruber*'s disclosures, an ordinary artisan would have further found this option is prompted by voice. EX1003, ¶175; *see also* Section VI.B.

15. Claim 16: "The computer implemented method of claim 15, further comprising, based on receiving an input from the user to call the sender, calling the sender."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 16 for the reasons discussed above in claims 4 and 15. *See* Sections VI.D.3, VI.D.14 (claims 4, 15); EX1003, ¶¶143-148, 174-178. An ordinary artisan would have further found it obvious that, upon being prompted, a user would respond to the prompt, and the system would call the sender when the user responds affirmatively to the prompt because this is the natural flow of a speech-based prompt and was known in the art. EX1003, ¶176.

In addition, *Gruber* suggests a responsive system configured to prompt an action, receive a user's input, and then execute the action. EX1003, ¶177. For example, *Gruber*'s system is configured to prompt the user with an option to take an action (e.g., "Shall I send [your text]?"), receive a user's affirmative command for the system to complete the action (e.g., "yes"), and then complete the action (e.g., send the message). *Gruber*, ¶¶[0473]-[0482], [0685]-[0687]; *see also* ¶¶[0311], [0339]-[0373], [0395]-[0405], [0597].

An ordinary artisan would have been motivated to combine the asserted references with a reasonable expectation of success. *See* Section VI.B; EX1003, ¶178. An ordinary artisan would also have been motivated to implement *Gruber*'s responsive system configuration into *Boelter*'s notification method because doing

so would have enabled a hands-free way to respond to messages with phone calls. EX1003, ¶178. This would have further improved *Boelter*'s objective of decreasing distractions while driving because it would have reduced a user's need to engage with a screen. EX1003, ¶178. An ordinary artisan would have had a reasonable expectation of success in such a combination because *Gruber*'s responsive system configuration was a well-known and routine set of steps for virtual assistants. EX1003, ¶178.

16. Claim 17: "The computer implemented method of claim 2, wherein informing the user comprises informing the user of the identity of the sender and the subject of the received email, or the identity of the sender of the received text message or the message from the messenger application."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 17 for the reasons discussed above in 1[d], 1[f], and claim 2. *See* Sections VI.C.5, VI.C.7, VI.D.1 (1[d], 1[f], claim 2); EX1003, ¶¶101-103, 134-136, 179-180.

As discussed above with respect to 1[d], *Gruber* discloses reading sender names aloud when announcing messages. For emails, *Gruber* discloses announcing both the sender and subject. *Gruber*, ¶[0686] ("'Next message, From Harry Saddler, today at 8:23 am, with the subject: List reading use cases.'"). It would have been obvious to read the subject with the sender and then read the body. An ordinary artisan would have been motivated to read an email subject when informing the user of the message because *Gruber* teaches that "[t]he domain-

specific speakable text templates arrange the different data fields of a domain-specific item type in a suitable order…." and the subject provides context for the message. *Gruber*, ¶[0559]; EX1003, ¶180.

An ordinary artisan would have been motivated to combine the asserted references with a reasonable expectation of success. *See* Section VI.B; EX1003, ¶180.

17. Claim 18: "The computer implemented method of claim 1, wherein after reading aloud the body of the received email, text message or message from the messenger application, the received email, text message or message from the messenger application is automatically dismissed and a status of the received email, text message or message from the messenger application is changed to read."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 18. EX1003, ¶¶181-184.

Boelter and *Gruber* describe messages as having read and unread states. *Boelter*, ¶[0036] ("[T]he summarized notification page may include any missed calls[and] unread notifications of other types …."); *Gruber*, ¶¶[0467]-[0472], [0682], [0685], [0734]. *Boelter* notifies users of incoming messages—missed calls, unread notifications, and "messages of other types not previously notified." *Boelter*, ¶[0036]. These newly received, previously unread messages and

notifications appear, for example, in *Boelter*'s Figure 4:

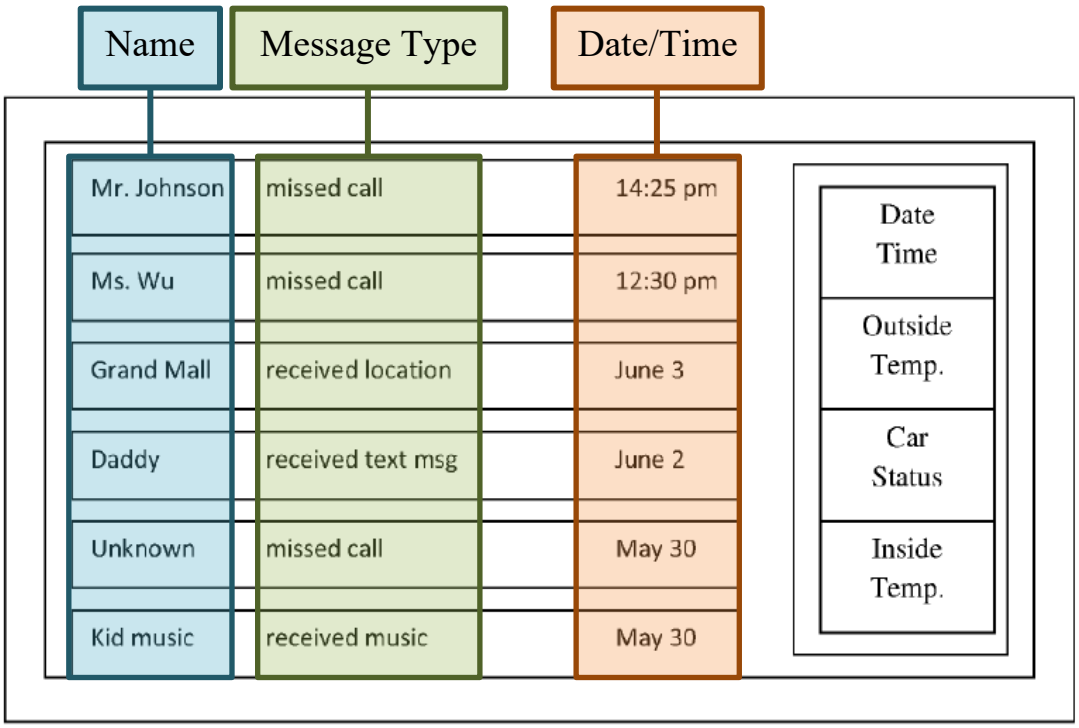


Fig. 4

Boelter, Fig. 4 (annotated), ¶[0036]. *Gruber* similarly discloses presenting "NEW" or "unread" messages to the user. *Gruber*, ¶¶[0467]-[0472], [0682]-[0683], [0685]. *Gruber*'s system tracks a "read/unread status" that indicates "NEW" messages, allowing it to present previously unreviewed messages to the user. *Gruber*, ¶¶[0682], [0685].

An ordinary artisan would have found it obvious to automatically dismiss the notification associated with a message as part of *Boelter*'s status change. EX1003, ¶183. This would have prevented the user from reviewing a notification

multiple times. EX1003, ¶183. In addition, an ordinary artisan would have found dismissing the notification furthered *Boelter*'s goal of reducing distractions as removing a notification from the queue would decrease the amount of information presented to the user. EX1003, ¶183. An ordinary artisan would have seen the dismissal feature as one of several limited options for addressing text messages after they are read: dismiss automatically, dismiss manually, or keep as "new." EX1003, ¶183 (citing *Barat*, ¶¶[0024]-[0027], [0040], [0054]-[0058], [0070], [0072]). Incorporating an automatic dismissal alongside a status change would avoid repeating notifications while limiting the need for user input. EX1003, ¶183.

An ordinary artisan would have had a reasonable expectation of success in such a combination. EX1003, ¶184. For example, automatic dismissal of messages/notifications was well known and commonplace. EX1003, ¶184. An ordinary artisan would further have reasonably expected a successful combination because adding the dismissal would not have required any significant changes to the system. EX1003, ¶184. Indeed, an ordinary artisan would have known or found it routine to include a dismissal step in a notification method because maintaining (or displaying) every notification in a queue indefinitely would have been unnecessary. EX1003, ¶184; *see also* Section VI.B.

18. Claim 19: "The computer implemented method of claim 18, wherein after the received email, text message or message from the messenger application is read aloud and the status changed to read, the user is notified that a new email, text message, or message from the messenger application has been received."

Boelter and *Gruber* or Boelter, *Gruber*, and *Polak* render obvious claim 19 at least for the reasons discussed above in claim 18. *See* Section VI.D.17 (claim 18) (discussing status changes). EX1003, ¶¶181-187.

After the status change discussed in claim 16, an ordinary artisan would have found it obvious to notify the user that a new message has arrived because *Gruber* discloses a notification before announcing the name of the sender. *See Gruber*, ¶¶[0185], [0409]-[0416]; EX1003, ¶186. For example, *Gruber* describes that "upon receiving text message … virtual assistant 1002 causes device 60 to output an audio indication, such as a beep or tone, indicating receipt of a text message." *Id.* Indeed, an ordinary artisan would have found this notification consistent with a cellphone's default notification as the use of sounds to announce incoming messages was common for all types of messages (unless the phone is silenced). EX1003, ¶186.

An ordinary artisan would have been motivated to combine the asserted references with a reasonable expectation of success. *See* Section VI.B; EX1003, ¶187. An ordinary artisan would have been motivated to include *Gruber*'s beep

upon receipt of a new message because this would align with standard notification methods. EX1003, ¶187. It would have been obvious to delay this beep until after the system finished reading the current message because doing so would avoid interrupting the message reading and distracting the user. EX1003, ¶187. An ordinary artisan would have had a reasonable expectation of success in including *Gruber*'s beep after a message was read because it merely requires completing one action before doing the next. EX1003, ¶187. Indeed, adding the beep would not require any changes that were not already known or within an ordinary artisan's skill set. EX1003, ¶187.

19. Claim 20: "The computer implemented method of claim 1, wherein the reading aloud is carried out by an audio system in a vehicle."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 20. EX1003, ¶¶188-189.

As discussed above, *Boelter* discloses its notification method may be "installed [i.e., carried out] in a vehicle." *Boelter*, ¶[0024], *see* Section VI.C (claim 1); *see also Boelter*, ¶¶[0009]-[0010], [0018]. When *Gruber*'s virtual assistant is used within a vehicle, the "assistant 1002 provides spoken output, which may be output via speakers (in device 60 or installed in the vehicle) …." *Gruber*, ¶[0187]. An ordinary artisan would have been motivated to read aloud *Boelter*'s notification queue/messages on a vehicle's audio system because *Gruber* suggests using a

vehicle's audio for playback. EX1003, ¶189. An ordinary artisan would have had a reasonable expectation of success in such a combination because it would not require any changes to a vehicle's system and had already been implemented in cars. EX1003, ¶189. Reading messages using a vehicle's audio system would be no different than sending music or a call to the system. EX1003, ¶189; *see also* Section VI.B.

20. Claim 21: "The computer implemented method of claim 1, wherein the input from the user is limited to a) pausing or rereading the reading of the received email, text message or message from the messenger application; b) dismissing the received email, text message, or message from the messenger application; c) deleting the received email; and d) calling the sender."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 21. EX1003, ¶¶190-193.

Gruber discloses "[a]fter reading the first message, a brief pause is provided, so that the user can enter a command for an action (e.g., reply, delete, repeat, etc.) …." *Gruber*, ¶[0686]. *Gruber* further suggests that message playback can be paused. *Gruber*, ¶¶[0562], [0706]. And as discussed above, *Gruber* also suggests receiving an input to call a sender. *Gruber*, ¶¶[0311]-[0405]; Sections VI.D.3, VI.D.12-VI.C.13 (claims 4, 13-14). *Gruber* suggests that providing users with limited options reduces the user's cognitive load while engaged in a distracting activity. *Gruber*, ¶¶[0554], [0561], [0567], [0710]; EX1003, ¶191.

It would have been obvious to limit the options presented to the user so that the system does not distract the user from driving. EX1003, ¶192. A user could replay, pause, dismiss, delete, or call back without looking at the touchscreen. *Boelter*, ¶[0032] (no free-form option). This closed set of limited options would also require less attention than drafting a responsive message, limiting distractions and reducing the user's cognitive load. EX1003, ¶192. Limiting the number of options would also present the user with fewer buttons (or portions of the touchscreen) to touch, reducing distractions from a more complicated menu structure. *Boelter*, ¶¶[0031]-[0033] (dividing touchscreen into quarters or thirds to avoid distracting driver); Fig. 3 (same).

An ordinary artisan would have been motivated to combine the asserted references with a reasonable expectation of success. *See* Section VI.B; EX1003, ¶193.

21. Claim 22: "The computer implemented method of claim 1, wherein the method is carried out on a device selected from the group consisting of: a smart phone, a mobile phone, a phablet, a tablet, a wearable device, a watch, and a vehicle operating device."

Boelter and *Gruber* or *Boelter*, *Gruber*, and *Polak* render obvious claim 22 because *Boelter* discloses that its notification method is "installed [i.e., carried out] in a vehicle" or integrated "in a mobile communication device such as a

smartphone." *Boelter*, ¶[0024]; *see* Sections VI.C, VI.D.17, VI.D.19 (claims 1, 18, 20); *see also Boelter*, ¶¶[0009]-[0010], [0018]; EX1003, ¶¶181-184, 188-189, 194.

E. Independent Claim 23

1. 23[pre]: "A mobile phone, comprising: one or more processors; a memory, a touch screen; and one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for steps comprising:"

Boelter discloses or renders obvious 23[pre]. As discussed above in 1[pre], *Boelter* teaches a computer-implemented method carried out on a control unit (i.e., processor), a memory, and a touchscreen. *Boelter*, ¶¶[0018], [0024], [0028]; EX1003, ¶¶94-96, 195-197; Section VI.C.1 (1[pre]).

Boelter discloses that its "method can be implemented in software or firmware to run in a messaging control unit of a vehicle. The program can also be stored on a computer-readable medium for execution by a control device of a vehicle." *Boelter*, ¶[0018]. *Boelter* further discloses that its method "could be implemented … in a mobile communication device such as a smartphone." *Boelter*, ¶[0024]. An ordinary artisan would have found it obvious to implement *Boelter*'s method on a smartphone having the same components (i.e., a processor, a program stored on a memory, and a touchscreen) and would have had a reasonable

expectation of success in doing so because *Boelter* suggests that a smartphone is a suitable option for its method. EX1003, ¶196.

Boelter and *Gruber* (Ground 1A) or *Boelter*, *Gruber*, and *Polak* (Ground 1B) render obvious the program including instructions for the following steps.

2. 23[a]: "determining without any input, by a user that one or more different types of messages selected from the group consisting of a text message, an email, and a message from a messenger application have been received;"

Boelter discloses 23[a] for the reasons discussed above in 1[a]. *See* Section VI.C.2; EX1003, ¶¶97-99, 198.

3. 23[b]: "analyzing without any input by the user a content of the received text message, message from the messenger application, or email;"

Boelter discloses 23[b] for the reasons discussed above in 1[b]. *See* Section VI.C.3; EX1003, ¶¶100-103, 199.

4. 23[c]: "placing the received email, message from the messenger application, or text message in a queue to be read aloud in order of receipt time;"

Boelter and *Gruber* render obvious 23[c] for the reasons discussed above in 1[c]. *See* Section VI.C.4; EX1003, ¶¶104-112, 200.

5. 23[d]: "informing the user without any input by the user that the received text message, email, or message from the messenger application has been received from a sender by reading aloud at least an identity of the sender;"

Boelter and *Gruber* (Ground 1A) or *Boelter*, *Gruber*, and *Polak* (Ground 1B) render obvious 23[d] for the reasons discussed above in 1[d]. *See* Section VI.C.5; EX1003, ¶¶113-121, 201.

6. 23[e]: "allowing the user a time to take an affirmative action to stop reading aloud to the user a body of the received text message, message from the messenger application, or email;"

Boelter and *Gruber* render obvious 23[e] for the reasons discussed above in 1[e]. *See* Section VI.C.6; EX1003, ¶¶122-128, 202.

7. 23[f]: "based on not receiving an input corresponding to the affirmative action from the user, reading aloud the body of the received text message, message from the messenger application or email, and for the received email, further reading aloud a subject of the received email."

Boelter and *Gruber* render obvious 23[f] for the reasons discussed above in 1[f]. *See* Section VI.C.7 (1[f]); EX1003, ¶¶129-135, 203-207. *Boelter* and *Gruber* further render obvious 23[f]'s additional language ("and for the received email, further reading aloud a subject of the received email") because *Gruber* discloses reading aloud the subject of an email. *Gruber*, ¶¶[0682]-[0688]; *see* Sections VI.C.7, VI.D.16 (1[f], claim 17).

Gruber's assistant provides email subjects as part of reading emails aloud. *Gruber*, ¶¶[0686]-[0688]. *Boelter*'s "summarized notification page" includes sender name, message type, and time/date, but allows the user to "view" the full message. *Boelter*, ¶[0036], Fig. 4. An ordinary artisan would have understood email subjects could be read with the user's name (informing the user of the message's arrival), with the message body, or both because *Gruber* teaches that "[t]he domain-specific speakable text templates arrange the different data fields of a domain-specific item type in a suitable order…." *Gruber*, ¶[0559]; EX1003, ¶204.

Gruber would have motivated an ordinary artisan to read an email's subject with the message body because it provides context for the message. EX1003, ¶205. In one example, *Gruber*'s assistant announces: "First message, From Harry Saddler, 3 days ago. With the subject: Short meeting today 3 pm." *Gruber*, ¶[0685]. After the pause, the unbounded content of the message is read: "'Harry wrote: Next meeting to scope a project X—is Monday October 17, 2 pm. I can't find a room, so let's meet in the IL2/4th elevator lobby, and go from there.'" *Id.* In this example, an ordinary artisan would have found reading the subject beneficial because it provides more context as to when the meeting in the lobby will take place—3pm today. EX1003, ¶205. An ordinary artisan would have found it helpful to read the subject with the body because it would clarify the message. EX1003,

¶205. An ordinary artisan would have found this beneficial when applied to *Boelter*, because it provides the user with additional information while they are engaged in a distracting activity, such as driving. EX1003, ¶205. Further, reading aloud the sender identity during the informing (step 23[c]) without the subject would enable the user to skip messages without divulging additional information about the message. An ordinary artisan would have had a reasonable expectation of success in the combination because *Gruber* already reads the subject aloud as part of its message flow. *Gruber*, ¶¶[0685]-[0688]; EX1003, ¶204. Reading the subject after the pause instead of before it would have involved similar programming that was within the skill of the ordinary artisan. EX1003, ¶206. Additional motivations to combine and reasonable expectations of success appear in Section VI.B.

F. Independent Claim 24

1. 24[pre]: "A system for delivering one or more of a text message and an email and a message from the messenger application to a user engaged in an activity, the system comprising a processor, a memory, and a touch screen, the system configured to deliver the text message, the message from the messenger application, and the email to the user by:"

Boelter discloses or renders obvious 24[pre] for the reasons discussed above in 1[pre] and 23[pre]. EX1003, ¶¶94-96, 195-197, 208; Sections VI.C.1, VI.D.4-V.D.7, VI.E.1; *see also*, *Boelter*, ¶[0029] (using the term "delivery embodiments" to describe its invention).

2. 24[a]: "running an application configured to deliver one or more of a text message, an email, or a message from the messenger application to a user while the user is engaged in an activity;"

Boelter discloses or *Boelter* and *Gruber* render obvious claim 24[a] for the reasons discussed above in 23[pre] and claim 3. EX1003, ¶¶94-96, 195-197, 208-209; *see* Sections VI.E.1, VI.D.2, VI.D.4 (24[pre], claim 3, claim 5). *Boelter* discloses that its user is engaged in driving, which is an activity. *Boelter*, ¶[0036]; EX1003, ¶209.

3. 24[b]: "determining without any input by a user that one or more different types of messages selected from the group consisting the text message, the email, and the message from the messenger application have been received;"

Boelter discloses 24[b] for the reasons discussed above in 1[a]. *See* Section VI.C.2; EX1003, ¶¶97-99, 210.

4. 24[c]: "placing the received email, message from the messenger application, or text message in a queue to be read aloud in order of receipt time;"

Boelter and *Gruber* render obvious 24[c] for the reasons discussed above in 1[c]. *See* Section VI.C.4; EX1003, ¶¶104-112, 211-212.

Claim 1 and claim 24 both recite analyzing content of a message (1[b]/24[d]) and placing the message in a queue (1[c]/24[c]) but list these claim limitations in a different order. An ordinary artisan would understand that analyzing a message content (e.g. identifying a sender name) could be done before

or after placing the message in a queue. EX1003, ¶212. *Boelter* does not limit these steps to a particular order. And swapping the order of the analyzing and queueing steps does not change how the queue is formed, how the user is informed of message, or how a user interacts with a message. EX1003, ¶212.

5. 24[d]: "analyzing without any input by the user a content of the received text message, message from the messenger application, or email;"

Boelter discloses 24[d] for the reasons discussed above in 1[b] and 24[c]. *See* Sections VI.C.3, VI.F.4; EX1003, ¶¶100-112, 211-213.

6. 24[e]: "informing the user without any input by the user that the received text message, message from the messenger application, or email has been received from a sender by reading aloud at least an identity of the sender;"

Boelter and *Gruber* (Ground 1A) or *Boelter*, *Gruber*, and *Polak* (Ground 1B) render obvious 24[e] for the reasons discussed above in 1[d]. *See* Section VI.C.5 (1[d]); EX1003, ¶113-121, 214.

7. 24[f]: "allowing the user a time to take an affirmative action to stop reading aloud to the user a body of the received text message, message from the messenger application, or email;"

Boelter and *Gruber* render obvious 24[f] for the reasons discussed above in 1[e]. *See* Section VI.C.6 (1[e]); EX1003, ¶¶122-128, 215.

8. 24[g]: "based on not receiving an input corresponding to the affirmative action from the user, reading aloud the body of the received text message, message from the messenger application, or email, and for the received email, further reading aloud a subject of the received email."

Boelter and *Gruber* (Ground 1A) or *Boelter*, *Gruber*, and *Polak* (Ground 1B) render obvious 24[g] as discussed above in 1[f] and 23[f]. *See* Sections VI.C.7, VI.E.7; EX1003, ¶¶129-135, 203-207, 216.

VII. MANDATORY NOTICES

A. Real Parties-in-Interest

Apple Inc. is the real party-in-interest.

B. Related Matters

This Petition is related to the petitions concurrently filed in IPR2025-01426 (U.S. 9,591,117), IPR2025-01427 (U.S. 10,110,725), IPR2025-01428 (U.S. 10,277,728), and IPR2025-01430 (11,316,964).

The '775 patent was asserted in *MessageLoud, Inc. v. Apple Inc.*, Case No. 6:25-cv-00185 (W.D. Tex.), and *MessageLoud, Inc. v. Samsung Electronics Co., Ltd.*, Case No. 2:25-cv-00486 (E.D. Tex.). Petitioner is not aware of any other judicial or administrative matter that would affect, or be affected by, a decision in this proceeding.

C. Lead and Backup Counsel, and Service Information

Lead Counsel	Backup Counsel
Joshua L. Goldberg, Reg. No. 59,369 joshua.goldberg@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-6092 Fax: 202-408-4400	Erika H. Arner, Reg. No. 57,540 erika.arner@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street Suite 800 Reston, VA 20190-6023 Phone: 571-203-2754 Fax: 202-408-4400 Jeffrey C. Totten, Reg. No. 65,229 jeffrey.totten@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4232 Fax: 202-408-4400 William C. Neer, Reg. No. 78,874 william.neer@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue NW Washington, DC 20001-4413 Phone: 202-408-4054 Fax: 202-408-4400

Please address all correspondence to counsel at the addresses above.

Petitioner consents to electronic service by email at the above email addresses.

VIII. GROUNDS FOR STANDING

Petitioner certifies the ’775 patent is available for IPR, and that Petitioner is not barred or estopped from requesting IPR challenging the claims on the grounds identified herein.

IX. CONCLUSION

For the reasons above, Petitioner requests institution and cancellation of each challenged claim.

The Office may charge any required fees to Deposit Account No. 06-0916.

Date: August 29, 2025

By: /Joshua L. Goldberg/
Joshua L. Goldberg, Lead Counsel
Reg. No. 59,369

CERTIFICATION OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(a)(1)(i), the undersigned hereby certifies that the foregoing PETITION FOR *INTER PARTES* REVIEW contains 13,311 words, excluding the parts of this Petition that are exempted under 37 C.F.R. § 42.24(a), as measured by the word-processing system used to prepare this paper.

Date: August 29, 2025

By: /Joshua L. Goldberg/
Joshua L. Goldberg, Lead Counsel
Reg. No. 59,369

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§42.6(e) and 42.105(a), the undersigned certifies that on August 29, 2025, a copy of the foregoing **Petition for *Inter Partes* Review of U.S. Patent No. 10,516,775**, the **associated power of attorney**, and **Exhibits 1001-1019** were served by FedEx Priority Overnight® on the correspondence address of record indicated in the Patent Office's public Patent Center system for U.S. Patent No. 10,516,775:

Payam Moradian
Moradian Law
10680 W Pico Blvd
Suite 300
Los Angeles, CA 90064

A courtesy copy of the foregoing was also served by FedEx Priority Overnight® on litigation counsel:

Craig L. Uhrich, ESQ.
222 Center Ave
Oakley, KS 67748

Dated: August 29, 2025

Respectfully submitted,

/Daniel E. Doku/
Daniel E. Doku
Senior Litigation Paralegal
Finnegan, Henderson, Farabow,
Garret & Dunner LLP