

Petition for *Inter Partes* Review of
U.S. Patent No. 8,447,607 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

DIALECT, LLC,
Patent Owner

Case IPR2024-00751
U.S. Patent No. 8,447,607 B2
Issue Date: May 21, 2013

Title: MOBILE SYSTEMS AND METHODS OF SUPPORTING
NATURAL LANGUAGE HUMAN-MACHINE INTERACTIONS

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,447,607 B2**

Table of Contents

	Page
I. MANDATORY NOTICES UNDER §42.8(A)(1)	7
A. Real Party-In-Interest under §42.8(b)(1).....	7
B. Related Matters under §42.8(b)(2)	7
C. Lead and Back-Up Counsel under §42.8(b)(3)	8
D. Service Information	9
II. FEE PAYMENT	9
III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D).....	9
A. Grounds for Standing	9
B. Identification of Challenge and Statement of Precise Relief Requested	10
C. Considerations Under §§ 314(a) and 325(d).....	10
IV. OVERVIEW OF THE CHALLENGED PATENT.....	12
A. Person of Ordinary Skill in the Art (POSITA)	12
B. Specification Overview	12
V. CLAIM CONSTRUCTION	15
VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	15
A. Overview of Grounds	15
B. Ground 1: Claims 1-15 Are Obvious Over Coffman In View Of Kennewick and Ross.	16
1. Claim 1: “A device for processing natural language inputs, comprising one or more processors configured to:” (1[pre])	16
(a) “receive a multi-modal natural language input from a user, the multi-modal natural language input including a natural language utterance and a non-speech input;” (1[a]).....	19

Table of Contents (continued)

	Page
(b) “generate a non-speech transcription from the non-speech input;” (1[b])	23
(c) “identify the user who provided the multi-modal natural language input;” (1[c])	26
(d) “generate a speech-based transcription based on a cognitive model associated with the user, wherein the cognitive model includes information on one or more prior interactions between the user and the device;” (1[d]).....	28
(e) “generate a merged transcription from the speech-based transcription and the non-speech transcription;” (1[e])	34
(f) “identify, from among a plurality of entries that are in a context stack and that are each indicative of context, an entry in the context stack that matches information in the merged transcription;” (1[f])	36
(g) “identify a domain agent associated with the entry in the context stack;” (1[g])	47
(h) “determine a request based on the merged transcription; and” (1[h])	50
(i) “communicate the request to the domain agent, wherein the domain agent is configured to generate a response to the user.” (1[i]).....	51
2. Claim 2: “The device of claim 1, wherein the response includes an aggregation of content gathered when the identified domain agent processes the request.”	53
3. Claim 3: “The device of claim 1, wherein the multi-modal natural language input is a first multi-modal natural language input and is received during a first session, the one or more processors further configured to receive a second multi-modal natural language input from another user during the first session.”	56

Table of Contents
(continued)

Page

4.	Claim 4: “The device of claim 1, wherein the one or more processors are configured to support interactions with a plurality of users during an interleaved session.”	57
5.	Claim 5: “The device of claim 1, wherein the cognitive model is a first cognitive model, wherein the one or more processors are configured to generate the speech-based transcription based on a second cognitive model that includes information on one or more interactions between the device and a plurality of users of the device.”	59
6.	Claim 6: “The device of claim 5, wherein the one or more processors are configured to generate the speech-based transcription further based on an environmental model that has information indicative of an environment of the user.”	62
7.	Claim 7: “The device of claim 6, wherein the environmental model has information indicative of how noisy the environment of the user is.”	65
8.	Claim 8: “The device of claim 1, wherein the one or more processors are configured to receive one or more subsequent multi-modal inputs, the subsequent multi-modal inputs including at least one of a subsequent natural language utterance or a subsequent non-speech input.”	65
9.	Claim 9: “The device of claim 1, wherein the identified domain agent is configured to update the context stack in response to processing the request.”	66
10.	Claim 10: “The device of claim 1, wherein the one or more processors are configured to synchronize the context stack with a context stack of another device.”	67
11.	Claim 11: “The device of claim 10, wherein the one or more processors are configured to synchronize the cognitive model with a cognitive model of another device.”	70

Table of Contents
(continued)

	Page
12. Independent Claim 12	72
13. Claim 13: “The method of claim 12, wherein the cognitive model is a first cognitive model, and wherein the generating the speech-based transcription is based on a second cognitive model that includes information on one or more interactions between the device and a plurality of users of the device.”	72
14. Independent Claim 14	72
15. Claim 15: “The device of claim 14, wherein the one or more processors are configured to generate the speech-based transcription further based on an environmental model, wherein the environmental model includes information indicative of how noisy the environment of the user is.”	74
C. Ground 2: Claims 5-7, 13-15 Are Obvious Over Ground 1 Prior Art In Further View Of Lee.....	74
VII. CONCLUSION.....	78
CERTIFICATE OF SERVICE	80

Petition for *Inter Partes* Review of
U.S. Patent No. 8,447,607 B2

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 8,447,607 B2 to Chris Weider et al. (filed June 4, 2012, issued May 21, 2013) (“’607” or “’607 patent”)
1002	Declaration of Christopher M. Schmandt (“ Schmandt ”)
1003	International Patent Application Publication No. WO 00/20962 to Daniel Coffman, Stephane H. Maes et al. (filed October 1, 1999, published April 13, 2000) (“ Coffman ”)
1004	U.S. Patent App. Pub. No. 2004/0044516 A1 to Robert A. Kennewick et al. (filed June 3, 2003, published March 4, 2004) (“ Kennewick ”)
1005	U.S. Patent App. Pub. No. 2002/0133354 A1 to Steven I. Ross (filed August 16, 2001, published September 19, 2002) (“ Ross ”)
1006	U.S. Patent App. Pub. No. 2002/0087315 A1 to Victor Wai Leung Lee et al. (filed May 23, 2001, published July 4, 2002) (“ Lee ”)
1007	Redline comparison between Kennewick and the ’607 patent
1008	Excerpts from Dafydd Gibbon et al., <i>Handbook of Multimodal and Spoken Dialogue Systems : Resources, Terminology, and Product Evaluation</i> (2000) (“ Gibbon ”)
1009	Excerpts from <i>Microsoft Computer Dictionary</i> (5th ed. 2002)
1010	Excerpts from <i>Merriam-Webster’s Collegiate Dictionary</i> (10th ed. 1994)
1011	Declaration of Ingrid Hsieh-Yee, Ph.D.
1012	Prosecution history of U.S. Patent No. 8,447,607 B2
1013	Proof of Service of Complaint, <i>Dialect, LLC v. Google LLC</i> , Case No. 1:23-cv-00378 (D. Del.)

I. MANDATORY NOTICES UNDER §42.8(A)(1)

A. Real Party-In-Interest under §42.8(b)(1)

Google LLC (“Petitioner”) is the real party-in-interest to this IPR petition.¹

B. Related Matters under §42.8(b)(2)

The ’607 patent is the subject of pending litigation in which Petitioner is named as a defendant, *Dialect, LLC v. Google LLC*, Case No. 1:23-cv-00378 (D. Del.) (filed April 3, 2023), and *Dialect, LLC v. Bank of America, NA*, Case No. 2:24-cv-00207 (E.D. Tex.) (filed March 22, 2024). The ’607 was also the subject of *Dialect, LLC v. Samsung Electronics Co.*, Case No. 2:23-cv-00061 (E.D. Tex.), which was terminated on September 11, 2023. Petitioner was first served with a complaint alleging infringement of the ’607 patent on April 6, 2023. (EX1013, p.002.) On March 28, 2024, Petitioner’s motion to transfer to the Northern District of California was granted. *See* Dkt. 32, Case No. 1:23-cv-00378 (D. Del.).

Petitioner is filing IPR petitions against eight patents asserted by Patent Owner in the pending litigation involving Petitioner:

U.S. Patent No.	IPR Case
7,398,209 B2	IPR2024-00746

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties in interest to this proceeding.

Petition for *Inter Partes* Review of
U.S. Patent No. 8,447,607 B2

U.S. Patent No.	IPR Case
7,502,738 B2	IPR2024-00747
8,015,006 B2	IPR2024-00748
7,693,720 B2	IPR2024-00749
9,031,845 B2	IPR2024-00750
8,447,607 B2	IPR2024-00751
8,849,652 B2	IPR2024-00752
7,640,160 B2	IPR2024-00753

C. Lead and Back-Up Counsel under §42.8(b)(3)

Petitioner provides the following designation of counsel.

LEAD COUNSEL	BACK-UP COUNSEL
Heidi L. Keefe (Reg. No. 40,673) hkeefe@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington, DC 20004 Tel: (650) 843-5001 Fax: (650) 849-7400	Andrew C. Mace (Reg. No. 63,342) amace@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004 Tel: (650) 843-5287 Fax: (650) 849-7400
	Mark R. Weinstein (Admission <i>pro hac vice</i> to be requested) mweinstein@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (650) 843-5007 Fax: (650) 849-7400

LEAD COUNSEL	BACK-UP COUNSEL
	Reuben Chen (Admission <i>pro hac vice</i> to be requested) rchen@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (650) 843-5007 Fax: (650) 849-7400

D. Service Information

This Petition is being served by Federal Express to the correspondence address of record for the '607 patent, 174899 – The Law Offices of David A. Gerasimow, P.C., 211 W. Wacker Dr., Ste. 1717, Chicago, IL 60606. Petitioner consents to electronic service at the addresses provided above for lead and back-up counsel.

II. FEE PAYMENT

Petitioner requests review of 15 claims, with a \$41,500 payment.

**III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND
CONSIDERATIONS UNDER §§ 314(A) AND 325(D)**

A. Grounds for Standing

Petitioner certifies that the '607 patent is available for IPR and that Petitioner is not barred or otherwise estopped.

B. Identification of Challenge and Statement of Precise Relief Requested

Petitioner requests IPR institution based on:

Ground	Claims	Basis for Challenge under §103
1	1-15	Coffman (EX1003), Kennewick (EX1004) and Ross (EX1005)
2	5-7, 13-15	Ground 1 Prior Art and Lee (EX1006)

C. Considerations Under §§ 314(a) and 325(d)

§314(a): Petitioner is unaware of another post-grant challenge against the '607 patent. The pending litigation against Petitioner is still in early stages, with no discovery or claim construction having occurred and no trial date set, and as noted, was recently transferred to the Northern District of California. If IPR is instituted, Petitioner intends to bring a motion to stay the litigation.

§325(d): The prosecution of the '607 patent was relatively short. In an initial office action, the pending claims were deemed allowable based on an examiner's amendment. (EX1012, pp.00148-00155.) The Examiner identified both Kennewick (**EX1004**) and U.S. 7,137,126, the U.S. counterpart to Coffman (**EX1003**), finding that they taught most, but not all, of the claim limitations as amended:

Kennewick et al. (US 2004/0044516) and Coffman et al. (US 7,137,126) teach natural language systems which perform transcriptions (where Coffman teaches merged transcriptions from non-speech and speech input and where Kennewick teaches where

transcriptions can be based on a cognitive model [i.e. where speech recognition may use the user profile]) and which include user identification, context stacks, domain agents, and determining requests as claimed, but does not teach where an entry in a context stack (which includes a plurality of entries that are each indicative of context) that matches information in the merged transcription is identified (merged from the speech and non-speech-based transcription where the speech-based transcription is generated based on the cognitive model including information on one or more prior interactions)

(EX1012, p.00152.) As discussed in the analysis below, Petitioner respectfully submits the Examiner materially erred in not recognizing that the claim 1[f] step of identifying an entry in a context stack that matches information in a merged transcription would have been obvious in view of Coffman. Just like claim 1[f], Coffman describes a context stack whose entries are compared to a merged transcription of multi-modal input to find “an acceptable match” that “fits the utterance at the level of the NLU symbolic language.” (Coffman, e.g., 6:6-12, 38:8-16 (merged transcription), 29:18-30:3 (context stack), 30:4-10, 30:13-16, 41:8-9 (context stack entries), 40:25-27, 41:2-5, 42:1-5 (matching entries to transcribed user input).) In addition, as explained below, claim 1[f] would have also been obvious over **Ross** (EX1005), which was not identified during prosecution. Ross makes clear, with specific examples, that it would have been obvious for Coffman’s system to identify a context entry that matches information in a transcription of user input.

(Ross, e.g., ¶¶0034-0053 (describing testing transcribed user utterance against information in context list by matching text).) The combination would have predictably resulted in **“identify[ing], from among a plurality of entries that are in a context stack and that are each indicative of context, an entry in the context stack that matches information in the merged transcription”** as recited by claim 1[f]. In addition, with respect to claims 5-7 and 13-15, the Petition points to Lee (EX1006) in Ground 2 in connection with generating a speech-based transcription based on a **“second cognitive model”** or **“general cognitive model.”** Lee was also not identified during prosecution.

IV. OVERVIEW OF THE CHALLENGED PATENT

A. Person of Ordinary Skill in the Art (POSITA)

A POSITA as of August 29, 2005 would have possessed a bachelor’s degree in electrical engineering or computer science, and two years of work experience in the field of speech- and natural language-interfaces for interacting with and directing computer systems and software, including speech recognition and processing (and more experience could substitute for less formal education, and vice versa). (EX1002 (“Schmandt”), ¶¶11-15.)

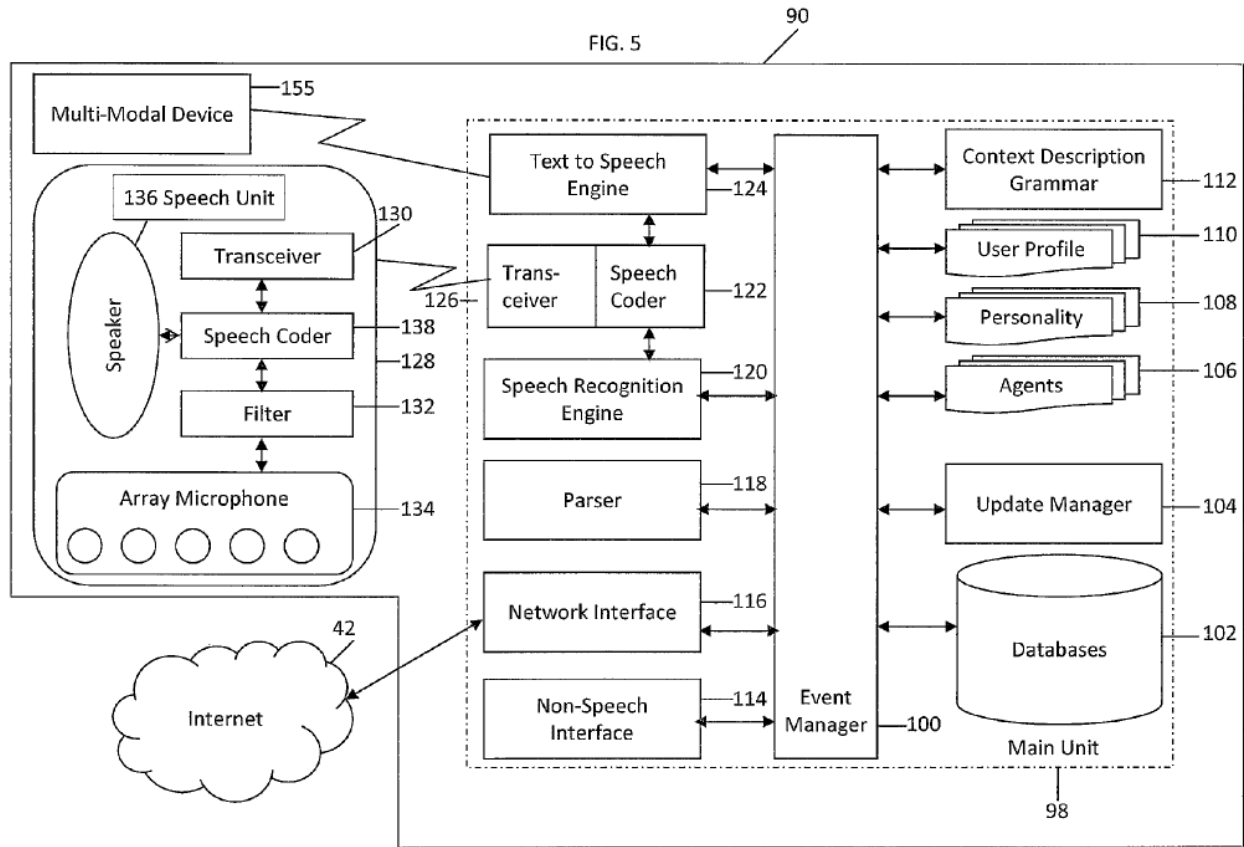
B. Specification Overview

The ’607 patent states that it “provides mobile devices having a speech interface and/or a combination of a speech interface and a non-speech interface to

enable natural language human-machine interactions” and that “[m]ore specifically, the invention enables mobile users to submit natural language speech and/or non-speech questions or commands in a wide range of domains.” (’607, 1:20-25.)

The ’607 patent admits that “[s]peech recognition has steadily improved in accuracy and today is successfully used in a wide range of applications,” and that “[n]atural language processing has been applied to the parsing of speech queries.” (’607, 1:65-2:2.) According to the ’607 patent, however, “[t]here remain a number of significant barrier to creation of a complete natural language verbal and/or textual-based query and response environment.” (’607, 2:6-8.) The ’607 patent purports to “overcome...drawbacks” of prior systems. (’607, 2:57-58.)

Figure 5 of the ’607 patent, reproduced below, “is an overall diagrammatic view of the interactive natural language processing system according to an embodiment of the invention”:



(’607, Fig. 5, 15:38-40.) As shown in the figure above, the system includes, among other things, speech unit 128, speech recognition engine 120, context description grammar module 112, parser 118, agents 106, multi-modal device 155 and non-speech interface 114. Speech unit 128 includes a microphone to receive a spoken utterance from a user. (’607, 25:50-52.) After initial processing, the speech is passed to speech recognition engine 120 for processing using the context description grammar module 112. (’607, 26:18-20.) Then, “[a]ny recognized information may be processed by the parser 118, which transforms information into complete algorithms and questions using data supplied by knowledge agents.” (’607, 26:20-23.) “The knowledge agents may then process the commands or questions.” (’607,

26:27-31.) Non-speech interface **114** and multi-modal devices **155** may be used as a substitute or complement to the speech interface. ('607, 26:51-53.)

V. CLAIM CONSTRUCTION

Express claim construction is unnecessary because the prior art renders the challenged claims obvious under any reasonable construction. Petitioner reserves its right to respond to Patent Owner claim construction arguments.

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Overview of Grounds

The challenged claim attempts to lay claim to well-known techniques for speech-enabled interfaces for interacting with and directing computer systems. The claims employ standard techniques that were known long before the '607 patent, such as using speech recognition, non-speech input, and context information to select an application for carrying out a command or question determined from the user's input.

The Petition relies primarily on **Coffman** (**EX1003**), which despite being published more than five years before the '607 patent's earliest claimed priority date, discloses a remarkably similar multi-modal conversational computing system. The Petition combines Coffman with **Kennewick** (**EX1004**) and **Ross** (**EX1005**). Kennewick shares a number of named inventors with the '607 patent, and the '607 patent incorporates a substantial amount of Kennewick's disclosures. (*See* EX1007.)

The Petition cites Kennewick for, among other things, its disclosures regarding “**cognitive models**” recited in, e.g., claim 1[d] and claim 5. The Petition cites Ross in connection with claim 1[f] for its teachings about identifying a context entry in a context list by matching text in a context entry with transcribed user input. The Petition further combines with Lee (EX1006) with respect to claims 5-7 and 13-15 in Ground 2 for its teachings related to generating a speech-based transcription based on a “**second cognitive model**” or “**general cognitive model**.”

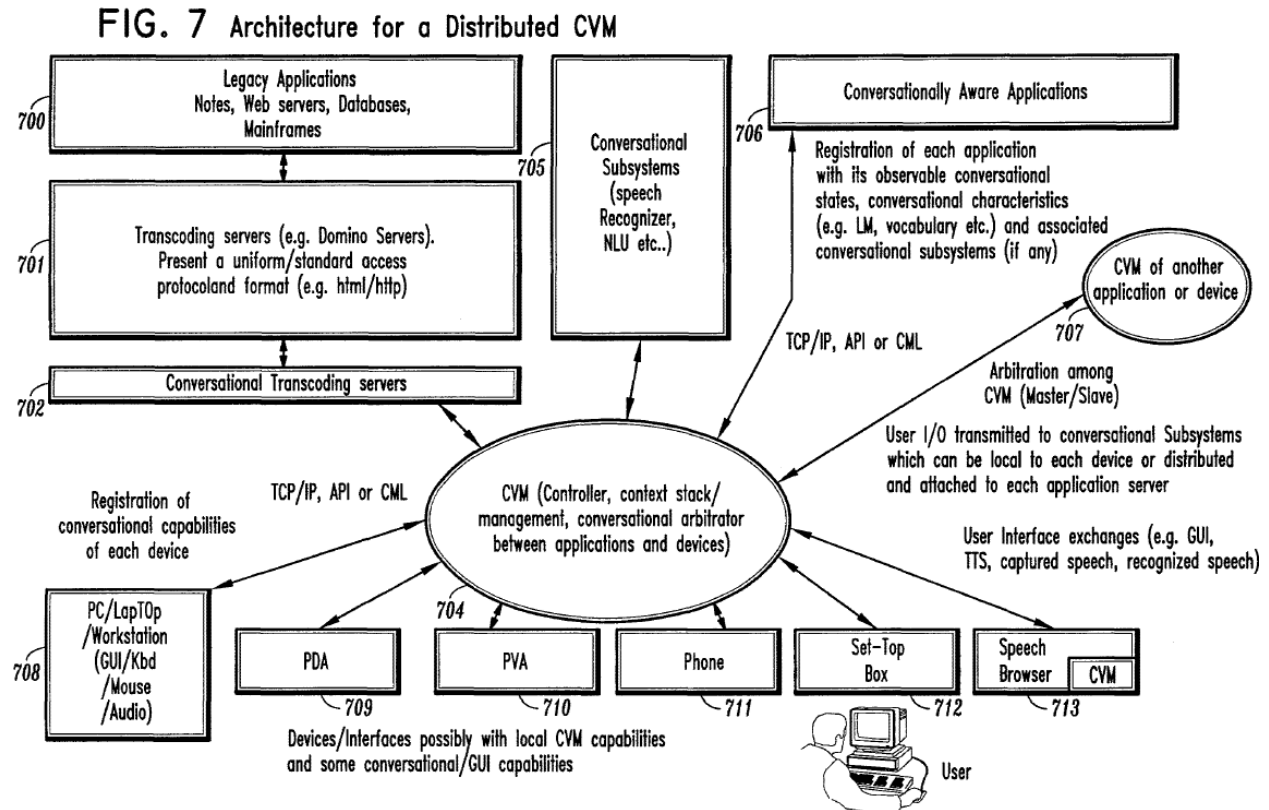
Coffman, Kennewick, Ross, and Lee are prior art under at least pre-AIA §102(b).

B. Ground 1: Claims 1-15 Are Obvious Over Coffman In View Of Kennewick and Ross.

1. Claim 1: “A device for processing natural language inputs, comprising one or more processors configured to:” (1[pre])

Coffman discloses a conversational computing system implemented on a “**device**,” such as a PC, PDA or “conversational car PC (PVA, Personal Vehicle Assistant).” (Coffman, 13:1-3 (“The CVM system may be built on top of a conventional OS and APIs 15 and conventional device hardware 16 and located on a server or any client device (PC, PDA, PvC).”), 56:22-25 (“conversational PDA,” “conversational car PC (PVA, Personal Vehicle Assistant),” and “conversational smart phone”), 55:27-30 (“When a user interacts with the collection of devices, the interaction may always be via a central unit such as a PVA (personal vehicle

assistant) 710 in a car, or a speech browser 713.”); Schmandt, ¶62.) Figure 7 of Coffman illustrates various devices, including PC 708, PDA 709 and PVA 710:



(Coffman, Fig. 7; *see also id.* 55:12-19 (“Referring now to Fig. 7, a diagram illustrates an architecture for a distributed CVM according to one aspect of the present invention. The heart of the distributed system is a CVM 704 (which may be located on a server, a PC, etc) which acts as the conversational arbitrator between a plurality of applications 706, devices 708-713, other CVM applications or devices 707 and conversational resources 705.”).)² As explained below, the conversational

² All underlining added to quoted material unless noted.

computing device, such as a PC or PVA, is “**for processing natural language inputs**,” such as spoken natural language utterances from a user.

A POSITA would have understood and found it obvious for the device to “**compris[e] one or more processors**”—this describes nothing more than routine and conventional components and operation of a generic computer system. (Schmandt, ¶63.) Such a system operates by using physical processor(s) (e.g., a microprocessor) to execute computer program instructions. (*Id.* (citing EX1009, pp.0024-0025 (“microprocessor”), p.0033 (“software”), p.0028 (“program”), p.0016 (“execute”)).) Coffman explains that its “CVM system may be built on top of a conventional OS and APIs 15 and conventional device hardware 16 and located on a server or any client device (PC, PDA, PvC).” (Coffman, 13:1-3.) A POSITA, therefore, would have understood and found it obvious that the device for processing natural language inputs in Coffman’s conversational computing system would comprise “**one or more processors**.” (Schmandt, ¶63.) A POSITA would have further understood and found it obvious that the one or more processors would be “**configured**” to perform the claimed steps, which describe operations performed by a combination of computer hardware and/or software. (*Id.*)

As shown below, the claimed device is disclosed and rendered obvious by the prior art.

- (a) “receive a multi-modal natural language input from a user, the multi-modal natural language input including a natural language utterance and a non-speech input;”
(1[a])

To start, Coffman discloses “receiv[ing] a multi-modal...input from a user.” Coffman explains that its conversational computing system “provides a universal coordinated multi-modal conversational user interface (CUI) 10.” (Coffman, 12:7-8.) With Coffman’s multi-modal conversational system, “speech is used in parallel with mouse, keyboard, and other input devices such as a pen,” as illustrated in Figure 6 below:

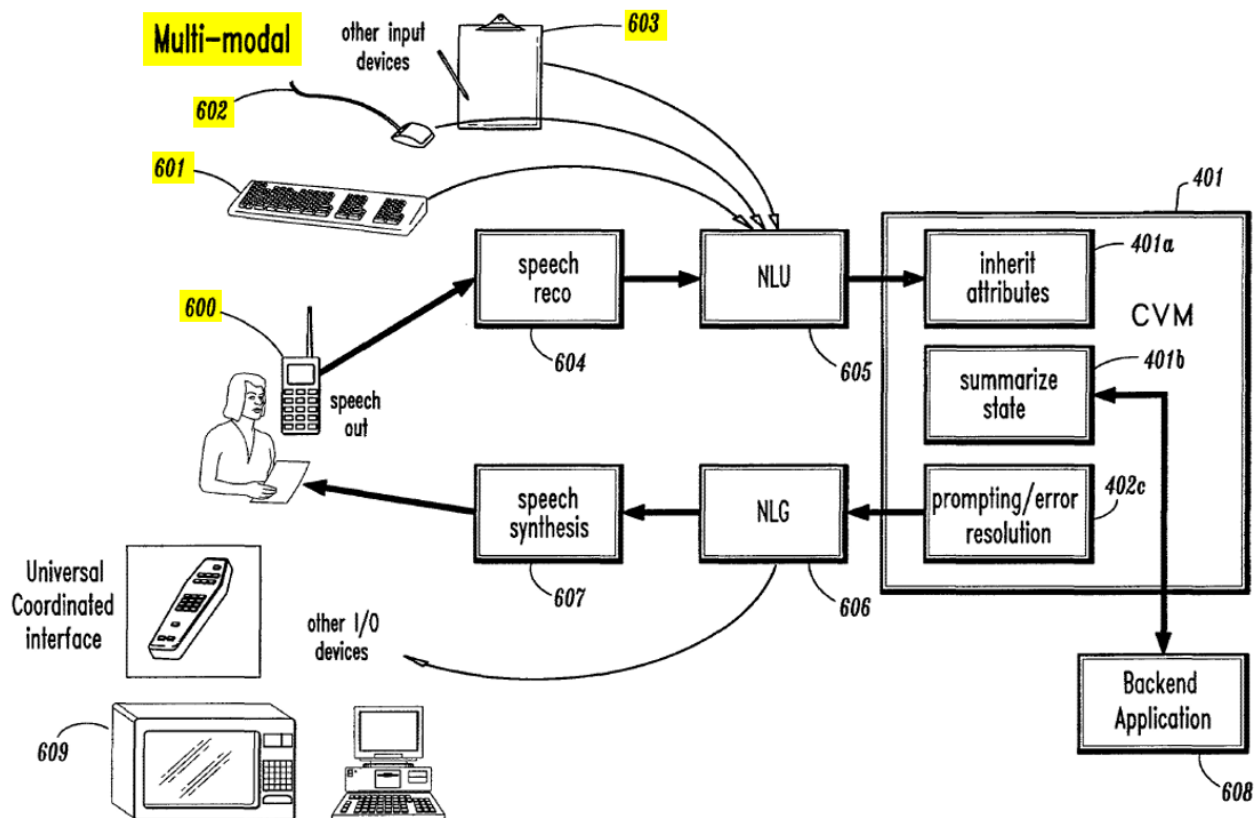


FIG. 6

(Coffman, Fig. 6 (highlighting added, showing multi-modal input); *see also id.*, 11:22-24 (“This is the notion of ‘multi-modality’ whereby speech is used in parallel with mouse, keyboard, and other input devices such as a pen.”), 12:8-10 (“The ‘multi- modality’ [sic] aspect of the CUI implies that various I/O resources such as voice, keyboard, pen, and pointing device (mouse), keypads, touch screens, etc can be used in conjunction with the CVM platform.”), 19:20-23 (describing I/O API layer 210), 53:8-10 (“Voice input from a microphone (e.g. the standard voice input”), 53:22-26 (“speech or any other input stream”).) Referring to Figure 6, Coffman describes receiving and processing multi-modal input from a user, including speech (from phone 600)—an **“utterance”**—and **“non-speech input”** (from, e.g., keyboard 601):

Referring now to Fig. 6, a diagram illustrates a conversational input/output interface in accordance with one embodiment of the present invention. As illustrated, a conversational input interface according to an embodiment of the present invention can process multi-modal input, that is, files/streams/resources, speech via a phone 600, keyboard 601, pointing devices 602, handwriting devices 603, including natural interfaces. This means that all the input and output events across all the modalities are caught and transferred to the dialog manager (that also stores it appropriately in the context stack). Spoken input from a speech client (e.g., telephone 600) is subject to a speech recognition process 604 and other input (e.g., keyboard, mouse clicks etc) are subject to NLU processing 605. Each input is subject to

attribute acquisition (401a) whereby the attribute value n-uples are acquired from the input.

(Coffman, 37:14-23.) Coffman in fact provides a specific example of receiving multi-modal input that includes both an “**utterance**” and “**non-speech input**”:

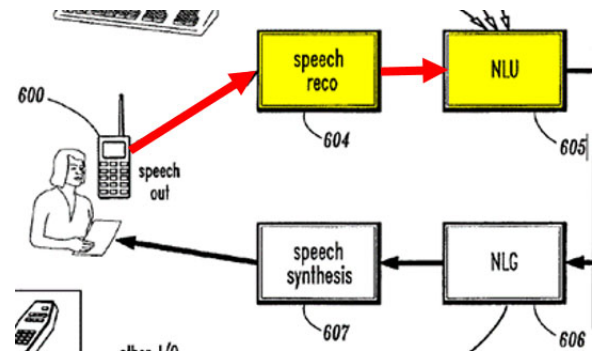
Indeed, any input in any modality is captured by the dialog manager of the CVM kernel layer as an event that is added to the associated context or context stack. For example, a mouse click or pointer/stylus pointing action followed by the command “I would like to open this” is disambiguated into a set of attribute value pairs: Command: open, Object: Windows or task selected by the last mouse click.

(Coffman, 6:30-7:3; Schmandt, ¶65.)

Coffman further discloses receiving “**multi-modal natural language input**” from a user because the input includes “**a natural language utterance.**” In the context of the ’607 patent, and speech recognition and processing generally, “**natural language**” generally speaking refers to the forms of language used by a person for communication, as opposed to artificial or formal languages such as logics, algebras, and programming languages—for example, English is a natural language, while C++ and Java are programming languages. (Schmandt, ¶66 (citing Gibbon (EX1008), p.0035 (“natural language” definition)).) Consistent with the ’607 patent, Coffman discloses several examples of *natural language* speech utterances, such as “I would like to open this,” “it should be synchronized with my PDA,” and “e-mail from my mother.” (Coffman, 6:32-7:3 (“For example, a mouse

click or pointer/stylus pointing action followed by the command ‘I would like to open this’ is disambiguated into a set of attribute value pairs: Command: open, Object: Windows or task selected by the last mouse click.”), 48:1-3 (“Since user identity can be associated in the context of each discourse, command ambiguity can be immediately and transparently resolved (e-mail from my mother is correctly understood, independently of the user).”), 56:25-57:2 (“grocery list,” “it should be synchronized with my PDA,” and “Do it!”); Schmandt, ¶66.)

Coffman’s receipt (and use) of spoken “**natural language**” input is also confirmed by Coffman explaining that the spoken input, once recognized, is subjected to natural language parsing and understanding (“NLU”) processes, as illustrated by Figure 6 (excerpted and annotated at right). (Coffman, e.g., Fig. 6,



53:22-26 (“User inputs can be represented by their transcriptions or their mappings into a symbolic language after parsing and NLU.”), 53:8-10 (“Voice input from a microphone (e.g. the standard voice input) can be arguments of function calls with the sequence of words, phones, or user identity or queries (symbolic language representation provided by NLU).”), 37:14-22 (“Spoken input from a speech client (e.g., telephone 600) is subject to a speech recognition process 604 and other input (e.g., keyboard, mouse clicks etc) are subject to NLU processing 605.”), 6:6-8 (“In

another aspect, the CVM provides the capability to have natural dialog with NLU, NLG, contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains).”), 5:1-4 (“In one aspect of the present invention, the CVM utilizes conversational subsystems (which may be local or distributed) including speech recognition, speaker recognition, text-to-speech, natural language understanding and natural dialog generation engines to understand and generate dialog between and user and machine.”), 37:29-32 (“The key conversational aspect is in the input procedure wherein commands can be entered in NLU (to provide natural language understanding of input queries)).”); Schmandt, ¶67.)

With respect to “**non-speech input**,” Petitioner further notes that Coffman describes how non-speech input is also “subject to NLU processing 605” (Coffman, 37:14-22, Fig. 6) thereby disclosing another way that Coffman teaches receiving “**multi-modal natural language input**” from a user.

(b) “generate a non-speech transcription from the non-speech input;” (1[b])

The ’607 patent written description mentions transcribing speech input a number of times, but it does not appear to mention transcribing non-speech input. (’607, e.g., 19:54, 22:16-17.) Transcribing (i.e. generating a “transcription”) is broadly understood as some form of recording, such as making a copy, whether verbatim or not, or converting information or data from one form to another. (Schmandt, ¶69 (citing EX1010, p.1253 (“transcribe”)).)

Coffman discloses “**generat[ing] a non-speech transcription from the non-speech input.**” Consistent with this ordinary understanding of “transcription,”

Coffman describes how multi-modal input is transcribed into ASCII or binary input:

[T]he CVM provides the capability to have a natural dialog with NLU, NLG, contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains). A conversational input interface is provided whereby a set of multi-mode input streams are each transcribed into an ASCII command or query (i.e., lists of attribute-value pairs or n-uples). Each input entity (command, NLU query field or argument unit (isolated letter, word, etc.) is associated with time-marks and appended accordingly to a compounded input stream.

(Coffman, 6:6-12; *see also id.*, 6:27-7:3 (“Using the conversational sub-systems, each input is converted into a binary or ASCII input (lists of attribute-value pairs or n-uples), which can be directly processed by the programming language as built-in objects.”), 38:26-29 (“In summary, the input procedure provides a set of multi-mode input streams, each transcribed into an ASCII command, query, or list of attribute value n-uples. Each input entity (command, NLU query field or argument unit (isolated letter, word etc.) are associated to time-marks and appended accordingly to

a compounded input stream.”); Schmandt, ¶70.)³ As shown in the excerpt above, Coffman describes transcribing “a set of multi-mode input streams.” Because, as discussed for 1[a], Coffman teaches that the “multi-mode input streams” include non-speech input, such as keyboard and mouse input, a POSITA would have understood Coffman’s description of transcribing “a set of multi-mode input streams” to include transcribing non-speech input. (Schmandt, ¶70.) In fact, Coffman provides a specific example of how speech input and non-speech input (a mouse click) are transcribed (“converted into a binary or ASCII input”) and ultimately “disambiguated into a set of attribute value pairs”:

Using the conversational sub-systems, each input is converted into a binary or ASCII input (lists of attribute-value pairs or n- uples), which can be directly processed by the programming language as built-in objects. Calls, flags and tags are automatically included to transmit between object and processes the conversational meta-information required to correctly interface with the different objects. Indeed, any input in any modality is captured by the dialog manager of the CVM kernel layer as an event that is added to the associated context or context stack. For example, a mouse click or pointer/stylus pointing action followed by the command “I would like to open this” is disambiguated into a set of attribute value pairs: Command: open, Object: Windows or

³ “ASCII” refers to text formatted using the American Standard Code for Information Interchange (“ASCII”). (Schmandt, ¶70 n.4 (citing EX1009, p.006 (“ASCII”)).)

task selected by the last mouse click.

(Coffman, 6:27-7:3; Schmandt, ¶70.)

(c) **“identify the user who provided the multi-modal natural language input;” (1[c])**

Coffman discloses **“identify[ing] the user who provided the multi-modal natural language input”** because its conversational system includes functionality for “speaker recognition” and “speaker identification,” such as “text-independent speaker identification” using a user’s command or request. (Schmandt, ¶71.) For example, Coffman explains that “the CVM utilizes conversational subsystems (which may be local or distributed) including speech recognition, speaker recognition, text-to-speech, natural language understanding and natural dialog generation engines to understand and generate dialog between and user and machine. (Coffman, 5:1-4.) Figure 4 of Coffman depicts conversational system 400 with speaker recognition system 409:

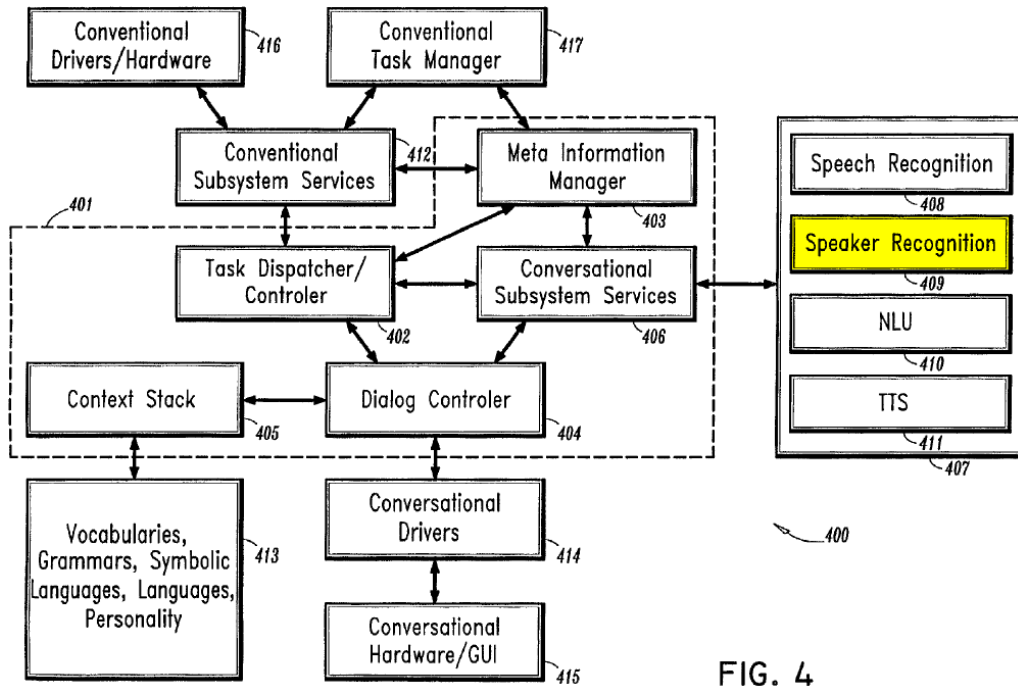


FIG. 4

(Coffman, Fig. 4 (highlighting added); *see also id.*, 29:1-4, 15:13-16:24 (listing various speaker-identification-related “conversational foundation classes” including “Acoustic Identify speaker,” “Acoustic Verify speaker” and “Speech biometrics identify.”).) Coffman further discloses that “[i]n a speech-based system, each command can be used to perform text-independent speaker identification.” (Coffman, 47:25-27; *see also id.*, 8:16-18 (“A user is automatically identified whenever a query to a resource is made.”), 48:7-9 (“[T]he user identity could be obtained through alternative procedures such as manual selection or input by the user of his or her identity.”), 46:28-29 (“[T]he stack of context should contain the identity of the user as the most recently authenticated identity.”); Schmandt, ¶71

(citing EX1008, pp.0047 (defining “speaker recognition”), 0051 (defining “text-independent speaker recognition system”)).)

- (d) **“generate a speech-based transcription based on a cognitive model associated with the user, wherein the cognitive model includes information on one or more prior interactions between the user and the device;”**
(1[d])

Coffman discloses **“generat[ing] a speech-based transcription.”** As discussed for 1[b], Coffman describes generating transcriptions of multi-modal input, including transcriptions of both speech-based input and non-speech input. (Coffman, 6:6-12; *see also id.*, 38:26-29.) For example, Coffman states, “[b]ased on the modality (pointer, keyboard, file, speech), the task dispatcher 402 redirects the stream to the appropriate conversational subsystems or conventional subsystem with speech inputs being transcribed/understood.” (Coffman, 41:2-4.) And as noted, Coffman provides an example of how both speech input and non-speech input (a mouse click) are transcribed and ultimately “disambiguated into a set of attribute value pairs.” (Coffman, 6:27-7:3.) As noted for 1[c], Coffman further teaches that the transcription can comprise information associating the recognized text with an identified user. (*See also* Coffman, 48:1-4, 8:22-23.)

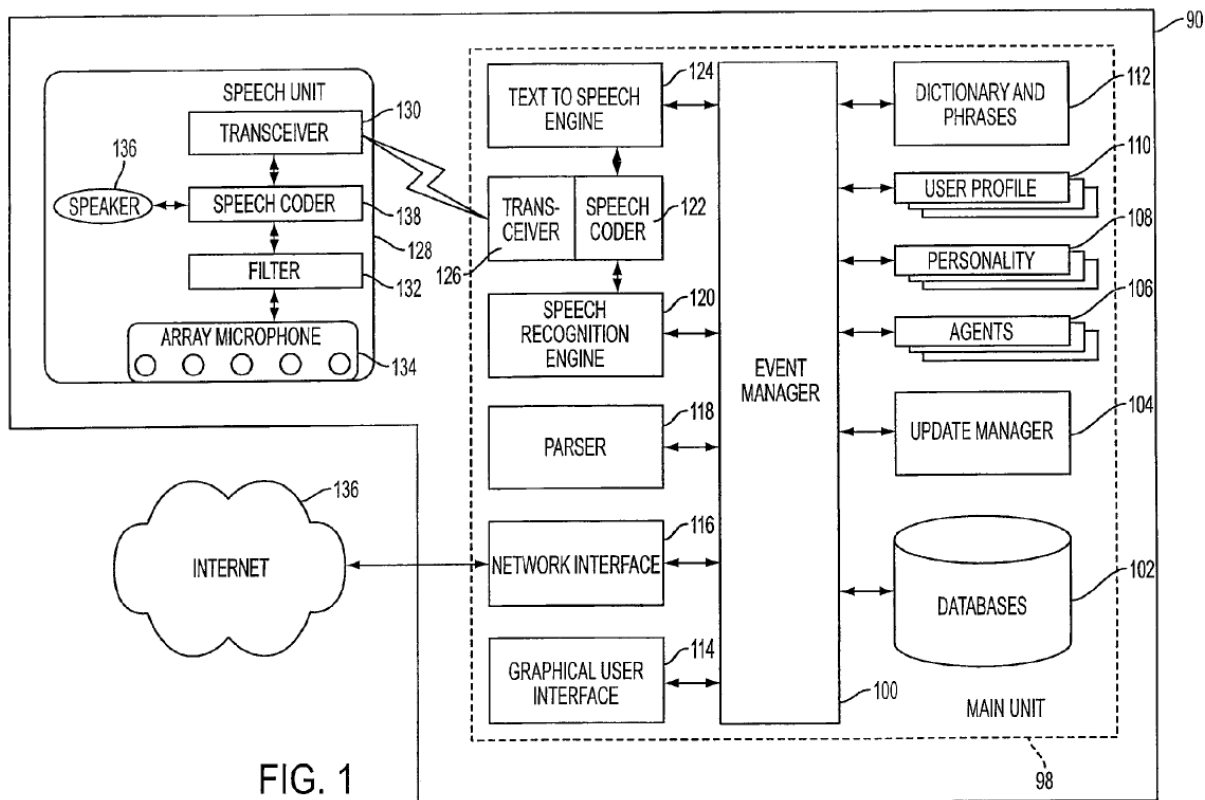
As explained below, it would have further been obvious to generate a speech-based transcription **“based on a cognitive model associated with the user, wherein**

the cognitive model includes information on one or more prior interactions between the user and the device.” (Schmandt, ¶¶73-82.)

With respect to a “**cognitive model**,” the ’607 patent describes it as information derived from a user’s interactions with the system that may be used in determining a user’s future actions by “assisting with speech recognition and/or question or command recognition.” (’607, 24:1-11; Schmandt, ¶74.)

Turning to Coffman, it describes collecting information about the user, such as preferences, security habits, biometrics and behavior. (Coffman, 33:1-5 (“The CVM 401 further comprises a meta information manager 403 which manages elements such as...information about the user (preferences, security habits, biometrics, behavior, etc.)”).) Coffman teaches that the collected information can be used “to perform more robust recognition” and to “recognize the user.” (Coffman, 37:5-10.) Coffman further teaches that the collected information can include “**information on one or more prior interactions between the user and the device**,” such as “the usual phrasing of a query, the type of query phrased, command frequency (often used, not often used), preferred applications, time or usage, etc.” (Coffman, 37:8-10.) Coffman therefore discloses a “**cognitive model**” comprising information collected about a user during their past interactions with the device that the system may use in connection with creating a transcription associated with a user of the user’s input. (Schmandt, ¶75.)

Kennewick (EX1004) further confirms and renders obvious the claimed “**cognitive model.**” Kennewick, U.S. Patent Application Publication No. 2004/0044516 A1, is entitled “Systems and Methods for Responding to Natural Language Speech Utterance.” The ’607 patent shares a number of named inventors with Kennewick (e.g., Robert A. Kennewick and Richard Kennewick) and incorporates a substantial amount of Kennewick’s disclosures. (Schmandt, ¶76 (citing EX1007).) Figure 1 of Kennewick, reproduced below, is nearly identical to Figure 5 of the ’607 patent. It depicts “an overall diagrammatic view according to one embodiment of the invention” (Kennewick, ¶0073):



(Kennewick, Fig. 1.) As shown in the figure above, system **90** includes, among other things, speech unit **128**, speech recognition engine **120**, dictionary and phrases **112**, parser **118**, and agents **106**. Kennewick provides substantially the same description of system operation as the '607, describing performing speech recognition on spoken input, processing recognized speech using a parser to identify commands and/or questions, and then selecting an agent to process the identified commands/questions. (Kennewick, ¶¶0086, 0088-0089, 0152-0154.)

Like Coffman, Kennewick discloses that its “system may create, store and use extensive personal profile information for each user.” (Kennewick, ¶0029.) The personal profile information includes “**information on one or more prior interactions between the user and the device,**” such as “the history of [the user’s] interactions with the system,” “their special use of terminology,” “history of questions asked,” and “session histories.” (Kennewick, ¶¶0114 (“Personal profiles may include information specific to the individual, their interests, their special use of terminology, the history of their interactions with the system, and domains of interest.”), 0029 (“Information in the profile may be added and updated automatically as the user uses the system or may be manually added or updated by the user or by others....Examples of information in a user profile includes, history of questions asked, session histories, formatting and presentation preferences, special word spelling, terms of interest, special data sources of interest, age, sex,

education, location or address, place of business, type of business, investments, hobbies, sports interests, news interests and other profile data.”); Schmandt, ¶77.)

Kennewick further discloses that the collected user profile information may be used in connection with speech recognition—i.e., “**generat[ing] a speech-based transcription**”—stating that “[t]he personal profile data may be used by the agents 106, the speech recognition engine 120, the text to speech engine 124, and the parser 118.” (Kennewick, ¶0114; Schmandt, ¶78.) Kennewick explains, for example, that “the system 90 may employ a speech recognition engine 124 seeding for improved word recognition accuracy, using data from the dictionary and phrase tables 112, user profiles 110, and the agents 106.” (Kennewick, ¶0149.) To “maximize the probability of correct recognition,” word probabilities “may be dynamically updated based on a number of criteria including the application domain, the questions or commands, contexts, the user profile and preferences, user dialog history, the recognizer dictionary and phrase tables, and word spellings.” (Kennewick, ¶0149.) For example, Kennewick states that “when a user uses a word that is not recognized at all or is not correctly recognized by the speech recognition engine 120 then the user may be asked to spell the word,” and the pronunciation for the word can then be added to the user’s profile. (Kennewick, ¶0151.) “Through this process the speech recognition engines learns with time and improves accuracy.” (Kennewick, ¶0151.)

Rationale and Motivation to Combine (Coffman and Kennewick). It would have been obvious to combine Coffman’s conversational system with Kennewick’s teachings providing additional detail about collected profile information and using the profile information to provide improved speech recognition. (Schmandt, ¶79.) Because the user profile information includes information associated with a user’s prior interactions with the system—such as dialog history and word spellings—the combination would have predictably resulted in **“generat[ing] a speech-based transcription based on a cognitive model associated with the user, wherein the cognitive model includes information on one or more prior interactions between the user and the device”** as claimed. (*Id.*)

Coffman and Kennewick are analogous references in the same field as the ’607 patent, and they would have been reasonably pertinent to problems facing the ’607 patent inventors, including collecting and using user profile information in connection with speech recognition and processing. (Schmandt, ¶80 (citing ’607, 1:20-25; Coffman, e.g., Abstract; Kennewick, ¶0003).)

A POSITA would have been motivated to combine. (Schmandt, ¶81.) Coffman and Kennewick describe collecting similar information about the user, such as dialog history. (Kennewick, ¶¶0149-0151; Coffman, 37:8-10 (“the usual phrasing of a query, the type of query phrased, command frequency (often used, not often used), preferred applications, time or usage, etc.”).) And as noted, Coffman already

discloses using collected user information to “perform more robust recognition” and to “recognize the user.” Kennewick provides express motivation to combine, explaining that the user profile information can be used to improve speech recognition accuracy. (Kennewick, ¶¶0149 (“[T]he system 90 may employ a speech recognition engine 124 seeding for improved word recognition accuracy, using data from the dictionary and phrase tables 112, user profiles 110, and the agents 106.”), 0151 (“Through this process the speech recognition engines learns with time and improves accuracy.”); Schmandt, ¶81.) A POSITA therefore would have been motivated to incorporate Kennewick’s techniques into Coffman’s similar system to obtain the same benefits and advantages. (Schmandt, ¶81.)

A POSITA would have had a reasonable expectation of success, as the combination would have involved the application of known technologies and techniques without undue experimentation. (Schmandt, ¶82 (citing Kennewick, ¶0006; EX1008, p.0024 (“adaptive language model” entry)).)

(e) “generate a merged transcription from the speech-based transcription and the non-speech transcription;” (1[e])

Coffman further discloses “**generat[ing] a merged transcription from the speech-based transcription and the non-speech transcription,**” describing how transcribed inputs from multi-mode input streams are “appended accordingly to a compounded input stream”:

In another aspect, the CVM provides the capability to have natural

dialog with NLU, NLG, contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains). A conversational input interface is provided whereby a set of multi-mode input streams are each transcribed into an ASCII command or query (i.e., lists of attribute-value pairs or n-uples). Each input entity (command, NLU query field or argument unit (isolated letter, word, etc.) is associated with time-marks and appended accordingly to a compounded input stream.

(Coffman, 6:6-12; *see also id.*, 38:26-29 (“In summary, the input procedure provides a set of multi-mode input streams, each transcribed into an ASCII command, query, or list of attribute value n-uples. Each input entity (command, NLU query field or argument unit (isolated letter, word etc.) are associated to time-marks and appended accordingly to a compounded input stream.”); Schmandt, ¶83.) Coffman further describes an example of how “transcripts are merged”:

In accordance with the present invention, the input problem is treated as a merge of the output of multiple decoders, ASCII transcription or a list of attribute value n-uples.. Each input stream is converted into its ASCII transcription and aligned with input time- marks via speech recognition processing 604. When different input stream are associated to the same task, the transcripts are merged as follows. First, commands and queries are sorted based on the time marks and appended to a single data stream. Command formulation can be checked against FSG rules and re-sorted to satisfy the grammar rules. NLU queries do not necessarily require re-sorting. For NLU queries, the symbolic fields are

filled for each stream, then compounded at the level of the final input stream.

(Coffman, 38:8-16.)

- (f) **“identify, from among a plurality of entries that are in a context stack and that are each indicative of context, an entry in the context stack that matches information in the merged transcription;” (1[f])**

As explained below, 1[f] would have been obvious in further view of **Ross** (**EX1005**).

To start, Coffman’s conversational system includes a **“context stack”** where the state and history of each active task/process, such as an application process, is tracked in its own “context”:

The core of the CVM 401 is the context stack 405 which operates and is managed under the control of the dialog controller 404 (it is to be understood that the context stack 405 id [sic; is] directly related to the global history and meta information repository discussed above). In general, the context stack 405 accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 (or at least identifiers of such conversational arguments) for the different engines that may be needed for input understanding (e.g., files or arguments that the engines use for performing their respective tasks such as active FSG, topic, vocabulary, HMM (hidden markov models), voiceprints, language models or possible queries for a speech input). In

other words, the term “context” refers to the state of each discourse (whether active or normative), which keeps track of the past history of the discourse, its current state, and the specific characteristics and full query arguments of the corresponding task (e.g., vocabulary file, language model, parsing, tags, voiceprint, TTS rules, grammar, NLU etc. of each active task/process) along with any data needed for input understanding. It is to be appreciated that, in a distributed CVM (as explained below), the context stack (as well as other CVM components) may be directly associated with networked services (i.e., distributed over the network) (as described above with respect to the context and global history).

(Coffman, 29:18-30:3; *see also id.*, 30:4-10.)

Coffman further makes clear that its “context stack” includes **“a plurality of entries...that are each indicative of context.”** (Schmandt, ¶86.) Coffman explains, for example, that “each new task, process, or thread creates a new stack entry and is associated with a discourse.” (Coffman, 30:4-10; *see also id.*, 30:13-16 (“The dialog controller 404 manages the context stack 405 by creating a new stack entry in the context stack 405 for each new task/process/thread that is spawned either local or remotely from a networked device....”), 41:8-9 (“Each new (active) task/process/thread creates a new stack entry in the context stack 405, with or without activated discourse.”).) Coffman further explains how context stack entries are searched when determining a context matching a user’s input. (Coffman, 40:25-

27 (“Queries that are rejected or too improbable cause the dialog controller 404 to peruse down the context stack 405 to look for the next available context.”), 42:1-2 (“Completion/search for the active context is performed from context to context down the stack.”).)

Coffman further teaches “**identify[ing]...an entry in the context stack that matches information in the merged transcription.**” (Schmandt, ¶87.) As noted above, Coffman explains that context stack entries are searched to identify a context matching the user’s input. (Coffman, 40:25-27, 42:1-5.) The search involves comparing the transcribed input against information in the context stack entries:

Based on the modality (pointer, keyboard, file, speech), the task dispatcher 402 redirects the stream to the appropriate conversational subsystems or conventional subsystem with speech inputs being transcribed/understood. The output of these subsystems is run down the context stack 405 to extract the active query and complete it.

(Coffman, 41:2-5.) A context stack entry is identified when an acceptable match between the user’s input and context stack is found:

Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon [sic] a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the

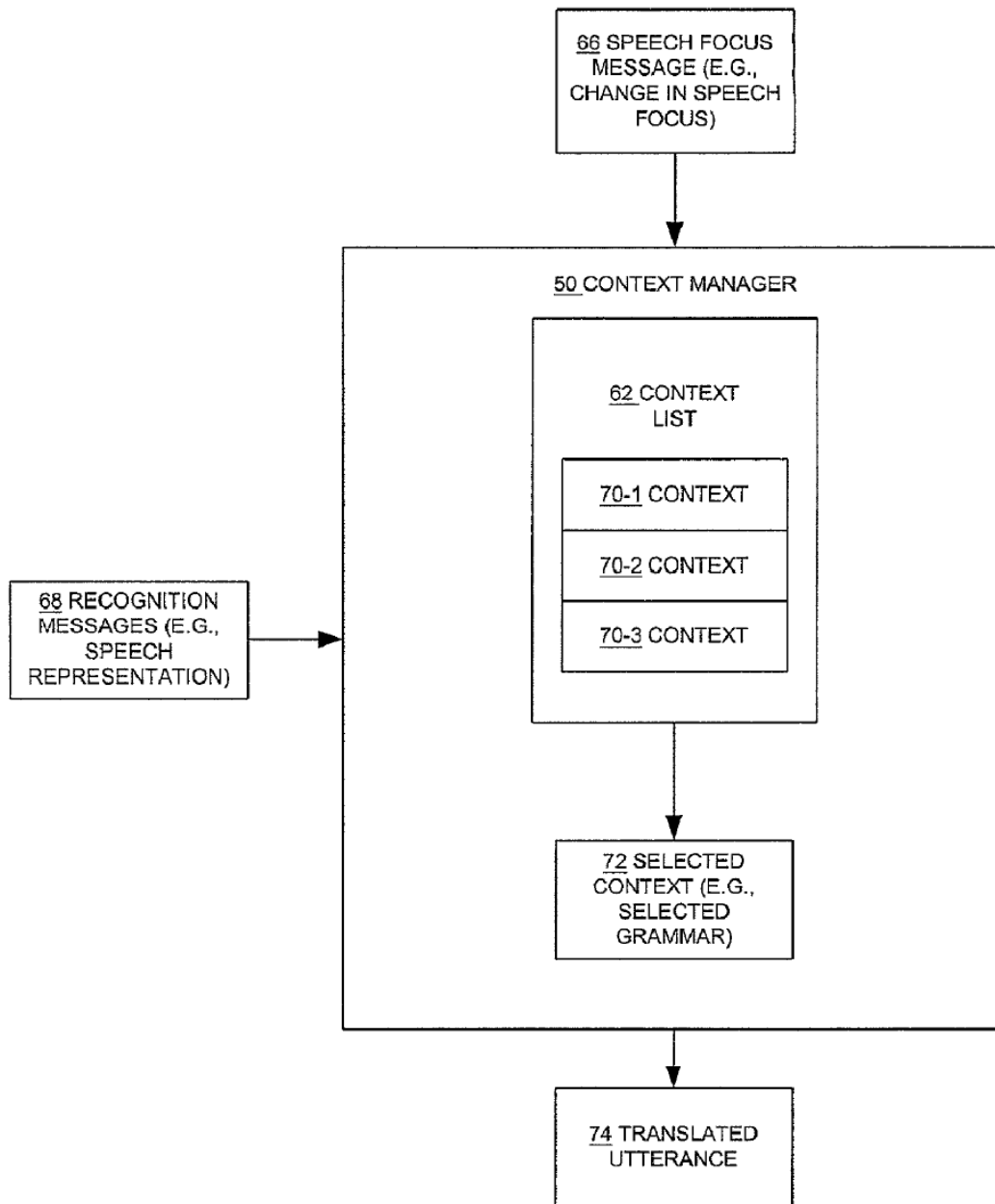
corresponding process becomes active.

(Coffman, 42:1-5.) For instance, taking Coffman’s example involving the utterance “I would like to open this” and mouse click input, Coffman teaches that the system matches the recognized word “open” in the user’s utterance and the mouse click information to a context entry for an application’s “open” command. (Coffman, 6:32-7:3; Schmandt, ¶83.) This is therefore an example of “**match[ing] information in the merged transcription,**” because as discussed for 1[e], Coffman teaches generating a merged transcription that would include transcriptions of the utterance and the mouse click input. (Coffman, e.g., 6:6-12, 38:8-16, 38:26-29; Schmandt, ¶87.)

To the extent there is any question whether Coffman alone discloses or renders obvious 1[f], it would have been obvious in view of **Ross**, as explained next. (Schmandt, ¶¶88-97.)

Ross, U.S. Patent Application Publication No. 2002/0133354 A1, is entitled “System and Method for Determining Utterance Context in a Multi-Context Speech Application.” (Ross, face page.) Ross describes techniques for determining a context associated with a user’s spoken command or question in order to determine an application to invoke to process the command or question, just like Coffman, Kennewick and the ’607 patent. (Ross, e.g., ¶¶0010, 0013; Schmandt, ¶89.)

Ross describes a technique where determining a context involves evaluating grammars associated with potential contexts against a user's recognized speech input—Figure 4 of Ross illustrates a general overview:



(Ross, Fig. 4.) As shown above, Ross includes a context manager **50** that includes a context list **62**. (Ross, ¶0033.) Context list **62**, in turn, “includes contexts **70** (e.g.,

70-1, 70-2, 70-3, etc.) for speech-enabled applications 26, which represent grammars for the applications 26.” (Ross, ¶0033.) A user’s transcribed utterance (a “recognition message”) is tested against the grammars in the context list in order to determine an application to invoke:

The context manager 50 maintains the priority and state of the various grammars in the context list 62 in the system....Recognition messages 68 from the speech engine interface 30 are tested by the context manager 50 against the active grammars in the context list 62 in priority order. When a successful match is found, the corresponding translation 74 is dispatched to the script engine 38 for execution, and the priority of the matching grammar (i.e., the selected context 72) is raised.

(Ross, ¶0034.)

Ross further makes clear that “test[ing]...against the active grammars” and finding a successful “match” involves matching information in a user’s transcribed utterance to grammar information of the context entries in the context list. (Schmandt, ¶87.) Ross explains that “a grammar is defined for each application 26” and describes two example grammars, one for an electronic mail application:

```
<mail> = do I have any messages |  
    open <message> |  
    create a message |  
    send <message>|  
    print <message>.  
<message> = the? <nth> message | it | this.  
<nth> = first | second | third | fourth | fifth | . . .
```

(Ross, ¶¶0035, 0040) and one for a calendar application:

```
<appointment> = do I have any appointments |  
    open <appointment> |  
    create an appointment |  
    print <appointment>.  
<appointment> = the? <nth> appointment | it | this.
```

(Ross, ¶0046). A grammar expression can generally comprise word tokens, rule references, and grammar operators. (Schmandt, ¶91.) For example, in the grammar for the electronic mail application shown above, the line “<message> = the? <nth> message | it | this,” defines a rule named “message.” (*Id.*) The rule specifies a grammar expression that includes word tokens (e.g., “the,” “message”), a reference to another rule (“<nth>”) and a grammar operator (“?”). (*Id.*) As shown above, each grammar includes a number of grammar expressions. Therefore, as Ross explains, the grammar for the electronic mail application allows spoken phrases such as “open the first message,” “create a message,” “send this,” and “print it” to be matched, while the grammar for the calendar application matches phrases such as “open the

first appointment,” “create an appointment,” “print the fourth appointment,” and “print it.” (*Id.*; Ross, ¶¶0041-0051.) The phrases that can be matched can be easily recognized by substituting the term in angle brackets with text (or an expression) to which it can correspond. For example, in the grammar for the electronic mail application, “<message>” can correspond to “the? <nth> message,” “it,” or “this.” Therefore, the grammar expression “print <message>” can match “print it” (as well as “print this” or “print the first message” and so on). (Schmandt, ¶91.) Ross discloses, therefore, that text in a user’s transcribed utterance, such as “print it,” matches the “print <message>” expression defined by the “mail” rule as well as the “it” expression defined by the “message” rule in the grammar for the electronic mail application. (*Id.*) With respect to the calendar application grammar, “print it” would also match the “print <appointment>” and “it” expressions in the “appointment” rules. (*Id.*)

Using the example electronic mail and calendar application grammars shown above, Ross further describes how identifying context(s) involves matching transcribed text to the grammar expressions in the grammars:

If both of these grammars were loaded into the context manager **50**, (ignoring foreground/background issues in this example, assuming that the grammars are the same for both foreground and background), the speech center system **20** is listening for any of the phrases accepted by either grammar. If the speech center **20** hears a phrase such as “print

the first message” or “print the first appointment,” the context manager 50 can readily figure out the intended target application 26 for the uttered sentence. Only one grammar will accept the phrase, which thus indicates the selected context 72 for that phrase and that associated application 26 is the one that should be targeted to receive the corresponding command.

If the sentence is “print it” however, both grammars are capable of accepting the utterance. The context manager 50 therefore has to make a choice by referring to the context list 62 of applications in order of recency of access. The context manager 50 tests the utterance against these grammars (indicated by the contexts 70 in the context list 62) in priority order, and passes the commands on to the first application 26 that has a grammar that will accept the phrase.

(Ross, ¶¶0052-0053; Schmandt, ¶92.)

In addition, with respect to the claimed “**context stack**,” Ross as noted describes a context manager 50 that includes a context list 62. (Ross, e.g., Fig. 4, ¶0033.) The context list 62, in turn, includes contexts 70. The contexts in the context list 62 are sorted in priority order, such that the most-recently-accessed context will be the first context evaluated when processing the next user utterance. (Ross, ¶¶0035 (“[W]hen an application 26 which is not the top priority application 26 is chosen as the target for a speech command, the application 26 indicated by the selected context 72 will move to the head of the list 62. In this way, the last application 26 that the user touched or talked to will get the first opportunity at interpreting the next

user utterance, and the other applications **26** will be ordered in a most-recently-accessed way.”), 0037 (“[T]he context manager **50** receives the recognition message **68** and selects a context **70** for a speech enabled application **26** by checking the speech representation against contexts **70**, starting at top of context list **62** to determine the highest priority context **70** that can process the speech representation”).) Ross’s priority ordering description of a context list is consistent with the well-known concept of a “stack,” in which the most recent item added to the stack will be the first item considered from the stack in a later operation. (Schmandt, ¶93 (citing EX1009, p.0017 (“last in, first out”))).) This “last in, first out” (“LIFO”) behavior of a stack contrasts with, for example, the “first in, first out” (“FIFO”) behavior of a queue. (*Id.* (citing EX1009, pp.0017 & 0022 (definitions of “first in, first out” and “last in, first out”))).)

Rationale and Motivation to Combine (Coffman and Kennewick with Ross). It would have been obvious to further combine Coffman and Kennewick with Ross’s teachings regarding matching transcribed user utterances with grammar information in context entries in a context stack. (Schmandt, ¶94.) The combination would have predictably resulted in Coffman’s conversational system identifying a context stack entry that matches information in a transcribed user’s utterance. (*Id.*) The combination would have therefore disclosed “**identify[ing], from among a plurality of entries that are in a context stack and that are each indicative of**

context, an entry in the context stack that matches information in the merged transcription” as claimed. (*Id.*)

Ross is an analogous reference in the same field as the ’607 patent and would have been pertinent to a number of problems facing the ’607 patent inventors, including techniques for using a grammar to identify a context for processing a user command or question and techniques for choosing an application to invoke to process a user’s a command or question in a user’s transcribed utterance. (Schmandt, ¶95 (citing ’607, 1:20-25; Ross, e.g., ¶¶0010, 0013).)

A POSITA would have been motivated to combine. (Schmandt, ¶96.) The combination would have involved the application of known techniques to improve a similar system in the same way. (*Id.*) Coffman, for its part, explains that context stack entries include information “needed for input understanding,” including “possible queries for a speech input” and “grammar[s].” (Coffman, 29:18-30:3.) Coffman does not detail how transcribed user input is matched against information in context stack entries, such as “possible queries” and “grammars.” But as shown above, Ross provides a detailed explanation and examples of such a matching process. Ross’s matching process similarly involves using grammar information in context entries stored in a context list that, as explained, is organized as a “stack.” Therefore, a POSITA, looking to implement or adapt Coffman’s conversational

system, would have been motivated to look to Ross's more detailed disclosures for further context and understanding of Coffman's teachings. (Schmandt, ¶96.)

A POSITA would have had a reasonable expectation of success, as the combination would have involved the straightforward application of conventional techniques and technologies without undue experimentation. (Schmandt, ¶97 (citing Ross, ¶0060; EX1009, p.008 ("Backus Naur Form")).)

(g) "identify a domain agent associated with the entry in the context stack;" (1[g])

1[g] recites a "**domain agent**." Generally speaking, in the software context, an "agent" broadly refers to a program that performs actions on behalf of another entity, such as a user or another software program or computing system. (Schmandt, ¶98.) Consistent with this general understanding, the '607 patent describes a "**domain agent**" as an agent that organizes "domain specific behavior and information" and "may be data-driven, scripted or created with compiled code." ('607, 25:34-36, 28:11-12; Schmandt, ¶98.)

A "**domain agent**" in Coffman corresponds to the executable software of a conversationally-aware application (and associated data, such as vocabulary, grammar, language models and NLU symbolic mapping). (Coffman, e.g., 13:26-29 ("The code implementing the applications, API calls and protocol calls includes interpreted and compiled scripts and programs, with library links, conversational logic (as described below) engine call and conversational foundation classes.");

Schmandt, ¶99.) Conversationally-aware applications “are applications that are specifically programmed for operating with a CVM core layer (or kernel) 14 via conversational application APIs 13.” (Coffman, 12:28-30.) A conversationally aware application operates with a CVM by providing its data files to the CVM for use by the conversational subsystems—an application hooks into the CVM API by “registering all its conversational capabilities and requirements” including “[a]ctive vocabulary, finite state grammar and language models to control the application” and “[s]ymbolic mapping if NLU is supported or at list [sic; least] a context state list.” (Coffman, 57:23-27; *see also id.*, 36:31-32 (“each application will provide the CVM 401 its own parsing and translation arguments”).)

A conversationally-aware application, consistent with the understanding of “agent,” receives, processes, and responds to user questions, queries and commands. (Coffman, e.g., 13:8-13 (queries transformed into “application calls” and application output returned to user), 37:2-4 (task executes query), 40:9-19 (“execute completed queries” and “output of executed or running processes”); Schmandt, ¶100.) A conversationally-aware application, consistent with the ’607 patent’s description of “domain agent,” also usually provides functionality related to a specific domain, such as a particular application/subject matter (such as email or spreadsheet) or device (such as a PDA or smart phone). (Coffman, 16:35-17:7 (listing “application specific” foundation classes, including “Manage_e-mail,” “Manage_calendar,” and

“Manage_addressbook/director”), 56:11-14 (e-mail), 58:30-32 (spreadsheet), 56:22-25 (“conversational PDA,” “conversational car PC (PVA, Personal Vehicle Assistant),” and “conversational smart phone”), 36:29-31 (“More specifically, the present invention provides the capability to have a natural dialog with NLU, NLG and mixed initiative across multiple applications, with multiple domains.”); Schmandt, ¶100.)

A conversationally-aware application in Coffman is also “**associated with [an] entry in the context stack.**” As discussed for 1[f], Coffman explains that context stack entries are created for and associated with applications. (Schmandt, ¶101; Coffman, 30:4-10 (“More specifically, each new task, process, or thread creates a new stack entry and is associated with a discourse. Each application may be associated with multiple discourses (e.g. the application management discourse and the application content navigation discourses).”); *see also* Coffman, 29:18-30:3.)

Coffman therefore teaches “**identify[ing] a domain agent associated with the entry in the context stack**” because, by identifying a context stack entry matching the user’s input, the associated conversationally-aware application is selected to process a command/query based on the user’s input. (Schmandt, ¶102; Coffman, 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other

application is stored to activate if the recognized query is rejected by the user.”); *see also* Coffman, 42:1-5, 41:2-5, 40:12-15 (“By way of example, the task dispatcher/controller 402 will decide on the basis of the context stack 405 the different statistical parsers that must operate on a query for the dialog controller 404 to identify the active context and complete the query.”).)

(h) “determine a request based on the merged transcription; and” (1[h])

1[h] was addressed in the analysis of 1[f]. As explained, both Coffman and Ross describe determining a command/query (“**determin[ing] a request**”) for an application to perform based on transcribed user input—and in Coffman, the user input is included in a “**merged transcription.**” (Schmandt, ¶103; Coffman, e.g., 6:32-7:3 (describing “I would like to open this” example), 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed”), 41:2-5 (“The output of these subsystems is run down the context stack 405 to extract the active query and complete it.”), 42:1-5 (“As soon [sic] a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active.”); Ross, e.g., ¶¶0052-0053 (e.g., “associated application **26** is the one that should be targeted to receive the corresponding command,” “passes the commands on to the first application **26** that has a grammar that will accept the phrase”), 0021 (“The output **16** is a command or other input based on the spoken utterance **14** directed to the speech enabled external

application **26** (see. FIG. 2) selected by the conversation manager **28** based on the context **70.**”), 0034 (“When a successful match is found, the corresponding translation **74** is dispatched to the script engine **38** for execution”).)

- (i) **“communicate the request to the domain agent, wherein the domain agent is configured to generate a response to the user.” (1[i])**

As discussed for 1[g] and 1[h], Coffman and Ross both describe how an identified application is invoked to process a command/query based on the user’s input. A POSITA would have understood that, as part of invoking the identified application, the command/query (**“request”**) would be communicated to the application. (Schmandt, ¶104.) For example, as noted, Coffman describes how “[t]he first task to have its dialog completed by producing a non-ambiguous query is executed” (Coffman, 37:2-4) and Ross similarly teaches that the identified application “receive[s] the corresponding command” (Ross, ¶0052). A POSITA therefore would have understood and found it obvious that a command/query (**“request”**) would be communicated to a conversationally-aware application in Coffman (**“the domain agent”**). (Schmandt, ¶104.)

Coffman further makes clear that a conversationally-aware application (**“domain agent”**) **“is configured to generate a response to the user.”** (Schmandt, ¶105.) Coffman discloses that a conversationally-aware application returns output as a result of processing the request, which is used to generate a response to provide

to the user. For example, Coffman describes executing completed queries and “return[ing] output of executed or running processes” and presenting them to the user using, for example, text-to-speech, as Coffman illustrates in Figure 6:

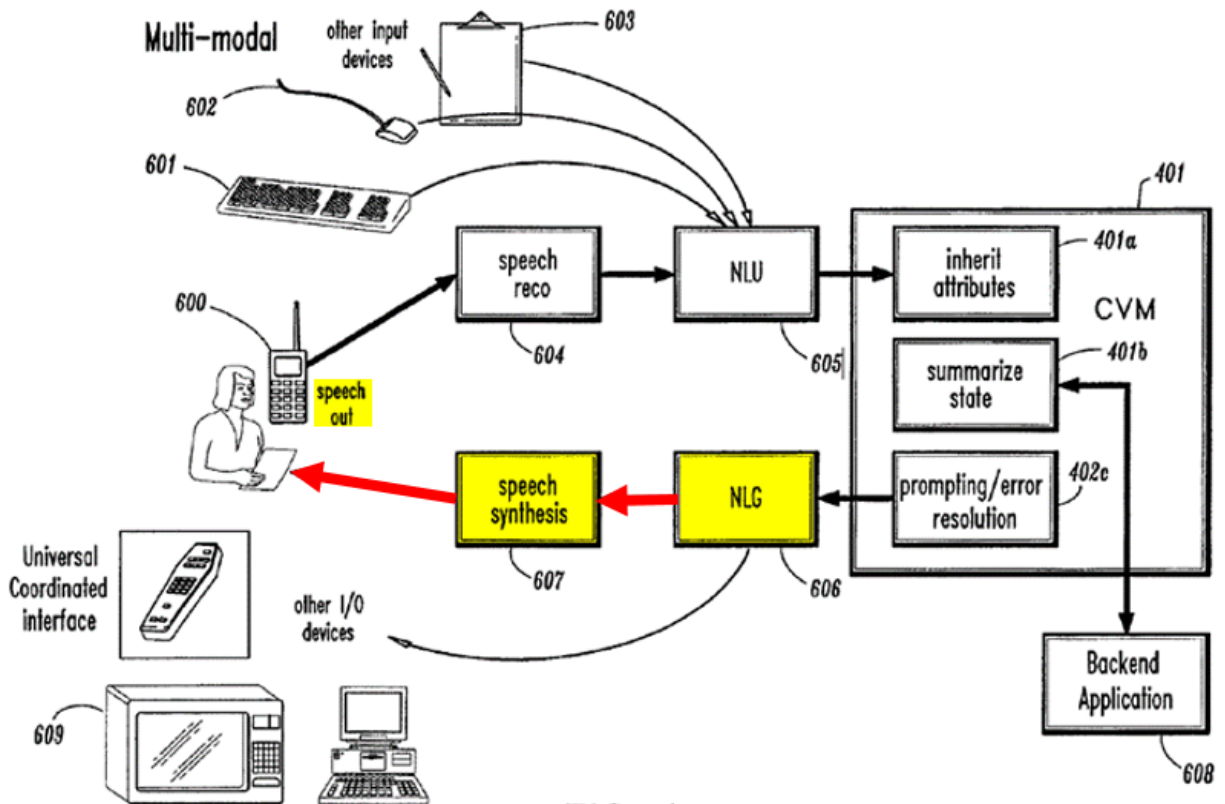


FIG. 6

(Coffman, Fig. 6 (annotations added), 40:9-19; *see also id.*, 13:8-13 & 58:9-11 (CVM converts application “output into speech...provided to the user”), 41:23-24 (“[C]omplete and non-ambiguous commands will result in certain results (e.g., outputs or actions).”) 39:3-5 (“With regard to conversational output dispatches and interface, the CVM 401 produces output to files/streams/resources, display (single or multi-windows, GUI, color, images, movies), audio. Individually, each output stream behaves conventionally.”).)

2. Claim 2: “The device of claim 1, wherein the response includes an aggregation of content gathered when the identified domain agent processes the request.”

The additional limitation of claim 2—“**wherein the response includes an aggregation of content gathered when the identified domain agent processes the request.**”—would have been obvious over Coffman alone or in view of Kennewick.

As discussed for 1[i], Coffman describes how a conversationally-aware application receives and processes a request and provides a response. Coffman also describes, in connection with processing a request, collecting different types of information or gathering information from multiple sources. (Schmandt, ¶107.) For example, Coffman describes a name dialer application on a smartphone that can use both information stored locally on the smartphone as well as information stored on a network machine. (Coffman, 45:2-6 (“For example, a name dialer application can be added to a conversational smart phone. The names that are often used are stored locally and recognized. On the other hand, unknown names or names that were never used before are sent to a more powerful networked machine for recognition and then download the updated information (phone number to dial etc.).”).) As another example, Coffman describes an in-vehicle navigation functionality where location information can be retrieved from local or remote sources and combined with traffic information in order to provide turn by turn directions to the user. (Coffman, 57:5-7 (“You now instruct the PVA to guide you turn by turn to the store. Your position

is computed, the location of the store is fetched, locally or from a server, and an itinerary is computed to take into account the latest traffic information.”).) Coffman’s in-vehicle navigation functionality, where different types of information (e.g., location and traffic information) are gathered in order to provide a response in the form of turn by turn directions, is an example of a “**response [that] includes an aggregation of content gathered when the identified domain agent processes the request.**” (Schmandt, ¶107.)

The additional limitation of claim 2 would have also been obvious over Coffman in view of Kennewick. As discussed above, Kennewick similarly discloses domain agents and that “queries are...sent to the designated agent[s].” (Kennewick, ¶0186.) Kennewick further explains that “[t]he agents may then, in turn, generate their own queries and/or commands to be sent to, for example, local or remote information sources to retrieve needed data.” (*Id.*; *see also id.* (describing example of compiling stock portfolio information involving multiple queries).) The queries are executed, and the domain agents return the results, which may have been based on information obtained from several sources. (*Id.* (“As a result of the execution of the queries, results may be returned by the domain agents and/or sources. The system **90** may then extract or scrape the desired information from the one or more results, which may be returned in any one of a number of different formats at **414**. That is, the results sought by a user may be the summary or the results of further processing

of information obtained from several sources for example.”).) Kennewick further describes how the results can be “gather[ed] and combin[ed] into a single best result” and “the compound results [] presented through the text to speech engine **124**, to the user in a useful and expected manner at **420**.” (*Id.*) Kennewick therefore discloses a generating a response to a user that “**includes an aggregation of content gathered when [designated agent[s]] process[] [a] request.**” (Schmandt, ¶108.)

The rationale and motivation to combine Coffman and Kennewick was discussed in the analysis of 1[d], which applies here as well. (Schmandt, ¶109.) Moreover, with respect to claim 2 in particular, a POSITA would have been motivated to incorporate Kennewick’s teachings so that the response generated for the user would have included an aggregation of content gathered when Coffman’s conversationally-aware application processed the request. (*Id.*) A POSITA would have readily appreciated the benefits of generating such a response, and that those benefits would have been applicable to a wide range of applications. (*Id.*) For example, a POSITA would have recognized that responding to a query often requires gathering different pieces of information from different sources—such as the in-vehicle navigation scenario that Coffman describes. As such, a POSITA would have recognized that aggregating content would allow the system to provide more complete and useful responses to the user. (*Id.*) Incorporating Kennewick’s teachings into Coffman’s conversational system would have involved the

straightforward application of Kennewick's technique to improve Coffman's similar conversational computing system in the same way. (*Id.*)

3. **Claim 3: “The device of claim 1, wherein the multi-modal natural language input is a first multi-modal natural language input and is received during a first session, the one or more processors further configured to receive a second multi-modal natural language input from another user during the first session.”**

Coffman explains that its conversational computing system “allows multiple users while actively running” and describes “multi-user” discourses, which Coffman describes as “conversational interaction[s] associated with a given task/process/thread.” (Coffman, 47:25-26 (“Conversational VM can adapt dynamically to the preferences of multiple users and to the active context. It allows multiple users while actively running.”), 47:27-48:7 (“Any change of user automatically implies the creation of a new active context which pushes the previous context down the context stack, unless the new active context is waived explicitly by the new user or the active application....Exceptions to this rule while traversing the context stack may automatically imply that the discourse becomes flagged as multi-users.”), 29:22-23 (“activated discourse (i.e., conversational interaction associated with a given task/process/thread)”).) A POSITA would have understood a discourse, as a conversational interaction associated with a given task/process/thread, would correspond to a “**session.**” (Schmandt, ¶110 (citing EX1009, p.0031 (“session”))).) By describing multi-user discourses—i.e., multiple

users participating in a single discourse—a POSITA would have understood and found it obvious for Coffman’s conversational system and user device (such as a PVA or PC (and its processors)) to be “**configured to receive a second multi-modal natural language input from another user during the first session.**” (*Id.*) And, as explained in the analysis of claim 4 below, to the extent there is any question whether Coffman discloses multiple users interacting with the same during the same session, it would have been obvious in view of Kennewick, which expressly discloses sessions in which multiple users interact with the system. (Kennewick, ¶0045.) Claim 3 therefore would have also been obvious over Coffman in view of Kennewick. (Schmandt, ¶110.)

4. Claim 4: “The device of claim 1, wherein the one or more processors are configured to support interactions with a plurality of users during an interleaved session.”

The ’607 patent describes an “**interleaved session**” as a session in which multiple users access the system in an “interleaved” manner during a session. (’607, 8:20-23 (“According to another embodiment of the invention, the system may support multiple users that access the system during a same session, in an interleaved or overlapping manner.”); Schmandt, ¶111.)

As with claim 3, the additional limitations of claim 4 would have been obvious over Coffman. Claim 4 is similar to claim 3, but broader because it recites “supporting interactions” instead of “receiving multi-modal natural language input.”

As discussed above for claim 3, Coffman discloses “multi-user discourses”—i.e., sessions in which multiple users may provide system input. A POSITA would have understood and found it obvious that, in the course of taking part in a multi-user discourse, users would naturally interact with the system in an “**interleaved**” manner. (Schmandt, ¶112.)

Moreover, it would have been obvious in view of Kennewick for Coffman’s conversational system to support “**interleaved sessions.**” (Schmandt, ¶113.) Kennewick expressly discloses “interleaved sessions,” providing a description that appears in substantially similar form in the ’607 patent. (*Id.* (side-by-side table comparing Kennewick, ¶0045, with ’607, 8:20-34).) Because both Kennewick and the ’607 patent contrast an “interleaved session” with multiple users using the system at different times, and provide substantially the same descriptions of how the system operates during an interleaved session, Kennewick makes clear that an “interleaved session,” just as in the ’607 patent, is a session where users interact with the system in an interleaved manner during the same session. (Schmandt, ¶114.)

The rationale and motivation to combine Coffman and Kennewick discussed in the analysis for 1[d] applies here as well. (Schmandt, ¶115.) With respect to claim 4, it would have further been obvious to combine Kennewick’s teachings regarding an “interleaved session” with Coffman, predictably resulting in “[t]he **device of claim 1, wherein the one or more processors are configured to support**

interactions with a plurality of users during an interleaved session,” as claimed.

(*Id.*) As discussed above, Coffman already teaches that multiple users can interact with the conversational system during the same discourse. Kennewick simply confirms that the interaction can be “interleaved.” Moreover, to the extent there is any question whether Coffman discloses multiple users interacting with the same during the same session, it would have been obvious in view of Kennewick. A POSITA would have readily appreciated that enabling multiple users to interact during the same session would beneficially allow for more flexible and natural interactions with the system. (*Id.*) A POSITA would have therefore been motivated to incorporate Kennewick’s teachings regarding interleaved sessions into Coffman’s conversational system. (*Id.*)

5. Claim 5: “The device of claim 1, wherein the cognitive model is a first cognitive model, wherein the one or more processors are configured to generate the speech-based transcription based on a second cognitive model that includes information on one or more interactions between the device and a plurality of users of the device.”

As discussed for 1[d], the written description of the ’607 patent describes a “personalized cognitive model **810**.” (’607, 24:1-11.) In addition to that model, the ’607 patent also describes a second, “general cognitive model **806**” that “corresponds to interaction patterns with the system for multiple users.” (’607, 24:12-21; Schmandt, ¶116.)

Kennewick discloses the claimed “**second cognitive model**” in the form of profile information about other users. (Schmandt, ¶117.) This is because Kennewick describes how its system “may employ a speech recognition engine **124** seeding for improved word recognition accuracy, using data from...user profiles 110.” (Kennewick, ¶0149; Schmandt, ¶117.) As discussed above for claim 1[d], the collected user profile information includes prior interactions with the system, such as “the history of [the user’s] interactions with the system,” “their special use of terminology,” “history of questions asked,” “session histories” and spellings and pronunciations provided by users. (Kennewick, ¶¶0114, 0029, 0151.) A POSITA would have understood and found it obvious that these interactions would have been “**between the device and a plurality of users of the device**” because Kennewick explains that its system supports multiple users in a given session—i.e., multiple users of the same device. (Schmandt, ¶117; Kennewick, ¶0045 (“Multiple users may use the system at different times or during interleaved or overlapping sessions.”), 0147 (“multiple users engage in overlapping sessions”), 0139 (describing microphone “steered to point in the direction of the one or more users speaking”).) Coffman describes multi-user sessions as well. (Coffman, 47:25-26, 47:27-48:7, 29:22-23.)

The rationale and motivation to combine Coffman and Kennewick was addressed in the analysis for 1[d] and applies here as well. (Schmandt, ¶118.) With

respect to claim 5, it would have further been obvious to incorporate Kennewick's teachings regarding using other users' profile information ("**second cognitive model**") in connection with speech recognition. (*Id.*) Kennewick provides express motivation to combine, explaining that user profile information can be used to improve word recognition accuracy. (*Id.* (citing Kennewick, ¶0149).) For example, if a user added a new word and pronunciation to their profile, including that new word for other users could improve the speech recognition accuracy of the other users' spoken input as well. (*Id.* (citing Kennewick, ¶0151 (speech recognition engine learns with time and improves accuracy based on user-provided spellings)).) Similarly, past questions input by other users could help the system recognize likely input from the user. (*Id.* (citing Kennewick, ¶0149 (describing updating word probabilities to maximize the probability of correct recognition based on, among other things, questions or commands, the user profile and preferences, word spellings, and user dialog history)).) A POSITA would have appreciated that applying Kennewick's teachings to Coffman would improve a similar system in the same way. (*Id.*)

Ground 2 explains how generating a speech-based transcription based on a "**second cognitive model**" would have also been obvious in further view of **Lee** (**EX1006**).

6. **Claim 6: “The device of claim 5, wherein the one or more processors are configured to generate the speech-based transcription further based on an environmental model that has information indicative of an environment of the user.”**

The written description of the '607 patent states that an “**environmental model**” may include “information associated with the user’s environment and surroundings” and makes clear that it encompasses a broad range of information:

According to one embodiment of the invention, the environmental model **808** may include information associated with the user’s environment and surroundings. The information may include the type of environment that a user is in (e.g., quiet or noisy); details of a microphone and/or speaker system; the user’s current global position and movement, as may be determined by GPS; current system status, such as what song/movie is playing, is the system in the midst of retrieving something, or other system status; details on all voice-enabled devices in the immediate vicinity, such as a presence of a voice-enabled TV, stereo, and DVD player in the same room; a user’s credit card information, such as numbers and current balances, wherein the user may ask a mobile telephone to download and pay for a video and the system may respond that there are insufficient funds; or other information. The information may be accessed to invoke a context, domain knowledge, preferences, and/or other cognitive qualities that enhance the interpretation of questions and/or commands.

(’607, 24:22-39; Schmandt, ¶120.)

An “**environmental model**” as claimed would have been obvious in view of Kennewick. (Schmandt, ¶121.) Kennewick explains “[t]he accurate recognition and parsing of the user’s speech requires the best possible signal to noise ratio at the input to the speech recognition engine **120**.” (Kennewick, ¶0138.) Kennewick describes using an “adaptive filter” in order “to optimize the signal to noise ratio at the input to the speech recognition engine **120**.” (Kennewick, ¶0141.) Kennewick explains that the adaptive filter has “band-shaping parameters” that are optimized using “[a]lgorithms comparing the background noise to the signal received from the users speech.” (Kennewick, ¶0141.)

Kennewick therefore discloses an “**environmental model**” in the form of an adaptive filter and its band-shaping parameters. (Schmandt, ¶¶121-122.) Consistent with the written description and claim 6, Kennewick’s environmental model “**has information indicative of an environment of the user**” because, as Kennewick explains, the band-shaping parameters are optimized by using the user’s environment—i.e., by comparing the background noise of the signal received from the user’s speech. (Schmandt, ¶122.) Kennewick further teaches that an adaptive filter and its band-shaping parameters are used in generating a speech-based transcription, by explaining that the adaptive filter is used to provide the best possible speech input for the speech recognition engine to process. (Kennewick, ¶¶0138, 0141.) Therefore, the output of the speech recognition engine—i.e., the

speech-based transcription—is “**generated...based on an environmental model.**”
(Schmandt, ¶122.)

The rationale and motivation to combine Coffman and Kennewick was discussed in the analysis of 1[d] and applies here as well. (Schmandt, ¶123.) With respect to claim 6, it would have further been obvious to incorporate Kennewick’s teachings about using an adaptive filter into Coffman’s conversational system. (*Id.*) The combination would have predictably resulted in Coffman’s conversational system processing spoken input from the user using an adaptive filter to optimize the input for speech recognition and generating a speech-based transcription. (*Id.*) The combination therefore would have disclosed “**the one or more processors are configured to generate the speech-based transcription further based on an environmental model that has information indicative of an environment of the user**” as claimed. (*Id.*)

Kennewick provides express motivation to combine. (Schmandt, ¶124.) As noted, Kennewick explains that “[f]iltering and noise elimination may be important in facilitating the various functionalities of the system **90**” and that [t]he accurate recognition and parsing of the user’s speech requires the best possible signal to noise ratio at the input to the speech recognition engine **120**.” (Kennewick, ¶0138.) Kennewick further explains that one way to optimize signal to noise ratio is to apply an adaptive filter as explained above. (Kennewick, ¶0141.) With respect to

Coffman, a POSITA would have recognized that the devices in the system would operate in a variety of different and dynamic sonic environments and therefore would have appreciated the benefits of using an adaptive filter. A POSITA, therefore, would have been motivated to incorporate Kennewick's adaptive filter teachings into Coffman's conversational system (or at least found it obvious to try) to improve a similar system in the same way. (Schmandt, ¶124.)

7. Claim 7: “The device of claim 6, wherein the environmental model has information indicative of how noisy the environment of the user is.”

The analysis of claim 6 addressed claim 7. A POSITA would have understood that Kennewick's adaptive filter and band-shaping parameters (“**environmental model**”) would have “**information indicative of how noisy the environment of the user is.**” (Schmandt, ¶125.) This is because, as explained, the parameters would be optimized using “[a]lgorithms comparing the background noise to the signal received from the users speech.” (*Id.* (quoting Kennewick, ¶0141).)

8. Claim 8: “The device of claim 1, wherein the one or more processors are configured to receive one or more subsequent multi-modal inputs, the subsequent multi-modal inputs including at least one of a subsequent natural language utterance or a subsequent non-speech input.”

The limitations of claim 8 describe nothing more than the normal, expected use of a multi-modal conversational system, and are disclosed and obvious over Coffman. (Schmandt, ¶126.)

A POSITA would have understood and found it obvious that Coffman's multi-modal conversational system, as a system for user(s) to carry out back-and-forth dialog with a computer system, would receive one or more subsequent multi-modal inputs, including both natural language utterances and non-speech input. (Schmandt, ¶127; Coffman, 4:6-8 ("The conversational computing paradigm prescribes that systems dialog with a user to complete, disambiguate, summarize or correct queries and the result of their executions."), 6:6-8 ("[T]he CVM provides the capability to have natural dialog with NLU, NLG, contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains).").) For example, Coffman explains that "[m]ixed initiative allows a user to naturally complete, modify, or correct a request via dialog with the system." (Coffman, 11:17-18.) In fact, Coffman provides a lengthy example of an ongoing interaction between a user and the conversational system that includes at least "subsequent natural language utterances," such as "grocery list" and "it should be synchronized with my PDA!" (Schmandt, ¶127 (citing Coffman, 56:26-57:20).)

9. Claim 9: "The device of claim 1, wherein the identified domain agent is configured to update the context stack in response to processing the request."

Claim 9 is obvious over Coffman. (Schmandt, ¶128.) Coffman discloses that a conversationally-aware application can update a context stack in response to processing a request, explaining for example, that "[e]ach action, completed query

and conversational argument of an active task as well as each returned value/result are stored on the context stack 405.” (Coffman, 40:27-29; *see also id.*, 30:6-8 (“Each context associated with a given discourse comprises the latest requests made to the corresponding process/task/thread as well as the latest output.”), 21:23-26 (“A context stack 217 is managed by the dialog manager 219. The context stack 217 comprises all the information associated with an application. Such information includes all the variable, states, input, output and queries to the backend that are performed in the context of the dialog and any extraneous event that occurs during the dialog.”); Schmandt, ¶128.)

10. Claim 10: “The device of claim 1, wherein the one or more processors are configured to synchronize the context stack with a context stack of another device.”

Claim 10 is obvious over Coffman. (Schmandt, ¶129.) Coffman explains that its conversational computing system can take the form of a distributed system involving multiple devices. For example, referring to Figure 3 (reproduced below), Coffman explains that “distributed conversational protocols 300 allow network conversational applications 200, 200a and network-connected devices (local client and other networked devices such as a server) to exchange information register their current conversational state, arguments (data files 209) and context with each other.” (Coffman, 22:32-23:3.)

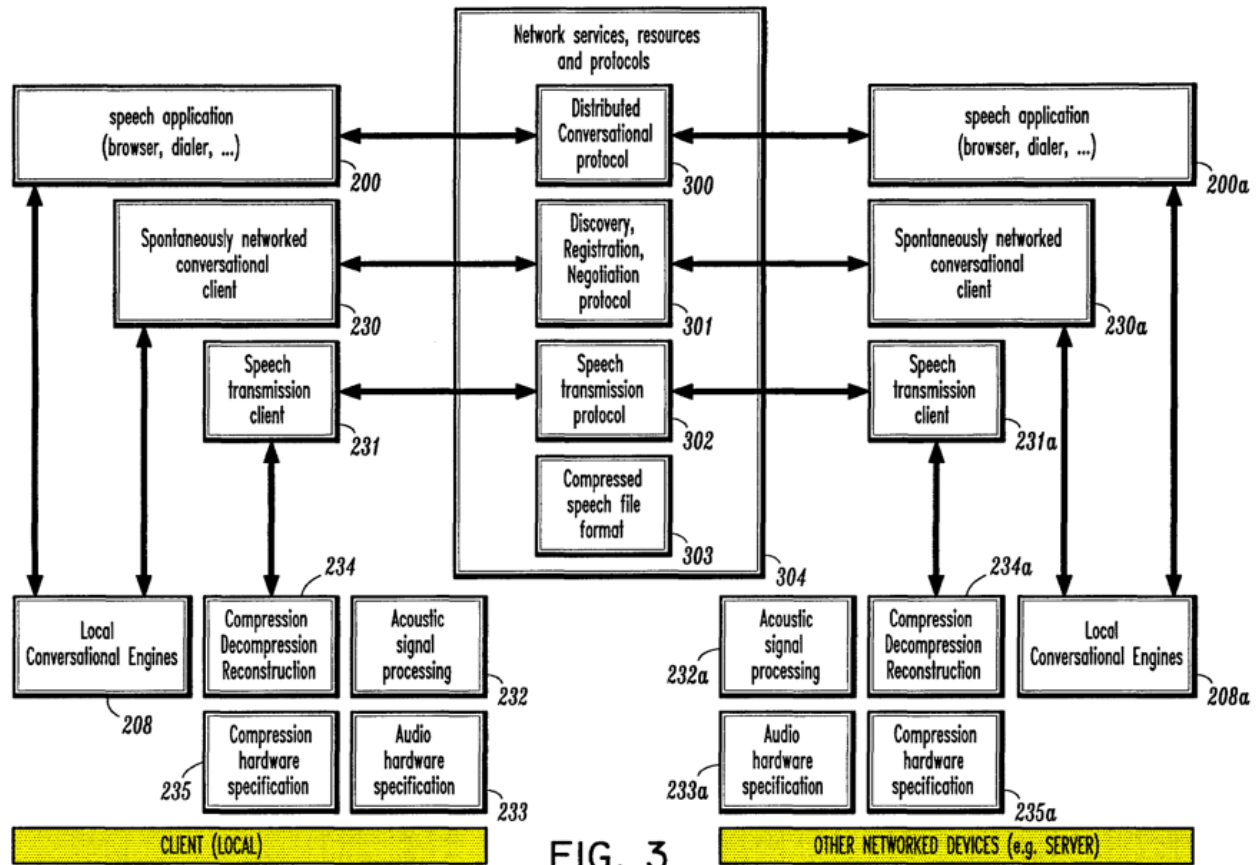


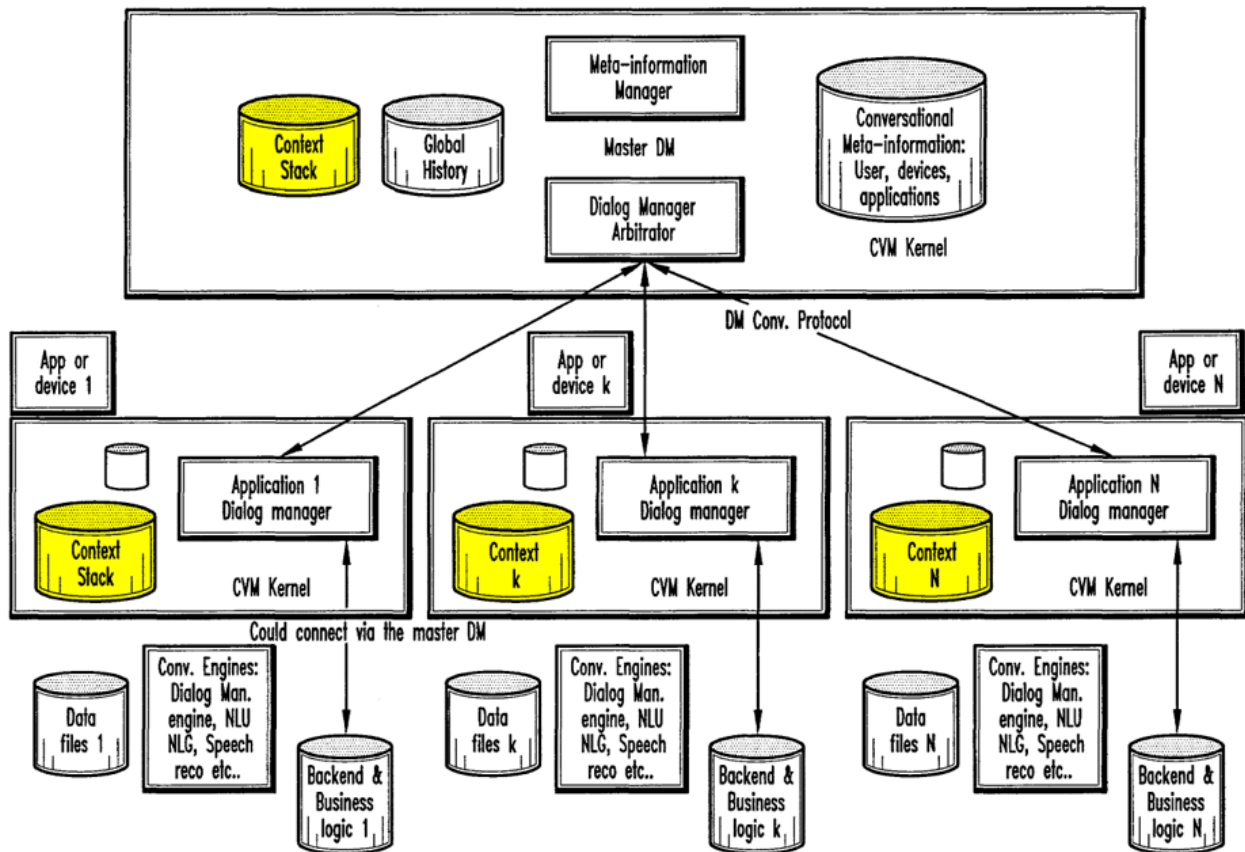
FIG. 3

(Coffman, Fig. 3 (highlighting added to indicate multiple network-connected devices).) Coffman further explains that the devices in its conversational system “typically...will have their own...context stack 217, 217a”:

Typically, different devices, CVMs or different applications will have their own dialog manager, context stack 217, 217a and global history 218, 218a. Through the dialog manager DM protocols (which are part of the distributed protocols 300 (Fig. 3), the different dialog managers will negotiate a topology with a master dialog manager and slave or peer dialog managers. The active master dialog manager (illustrated as dialog manger 219 in Fig. 9) will be responsible for managing the flow of I/O to the different managers to decide the active dialog and

appropriately execute a query and update the context and history.

(Coffman, 27:1-7.) Figure 10 of Coffman, for example, depicts a conversational system with multiple devices, each device having its own context stack (indicated in yellow highlighting):



(Coffman, Fig. 10 (highlighting added); *see also id.*, 27:24-26 (“Fig. 10 illustrates a hierarchical dialog between multiple dialog managers (i.e, the master arbitrator, and the slave dialog managers 1, k, and N) of various devices/applications (1, k and N).”).)

Coffman further makes clear that a device’s context stack can be “synchronized” with a context stack of another device, explaining:

With a master/slave network configuration, the master dialog manager can compile all the lists and match the functions and needs with conversational capabilities. In addition, context information may be transmitted by indicating passing or pointing to the context stack/history of the device or application that the controller can access and add to its context stack.

(Coffman, 25:10-14; Schmandt, ¶130.) Coffman further explains that, using dialog manager protocols, devices can exchange various kinds of information, including notification of “addition to contexts,” “context update” and “notification of context changes.” (Coffman, 27:7-20.)

A POSITA would have therefore further understood and found it obvious for a user’s device (such as PC or PVA) in Coffman’s conversational system to be **“configured to synchronize the context stack with a context stack of another device.”** (Schmandt, ¶131.)

11. Claim 11: “The device of claim 10, wherein the one or more processors are configured to synchronize the cognitive model with a cognitive model of another device.”

As discussed above for claim 1, Coffman and Kennewick disclose the claimed **“cognitive model”** in the form of user profile information. With respect to claim 11, Kennewick further discloses that user profile information can be synchronized between devices:

The system may be used in conjunction with a wide range of platform environments. These environments include, as a peripheral to a PC, a

portable or wearable computer or other computer, embedded in a PC or other computer, on a personal digital assistant (e.g., a PDA with wireless networking capability, in a specialized device such as a bar code scanner or payment terminal, on a wireless telephone, or other platforms....A user may use the system on one or more devices. In this case history and profile information for the user may be synchronized between the multiple devices on a periodic basis or other basis.

(Kennewick, ¶0043; Schmandt, ¶132.)

The rationale and motivation to combine Coffman and Kennewick was discussed in the analysis of 1[d] and applies here as well. (Schmandt, ¶133.) With respect to claim 11, it would have obvious to combine Kennewick’s teachings about synchronizing profile information with the conversational system of Coffman and Kennewick discussed above for claim 1. (*Id.*) The combination would have predictably resulted in the conversational system as described, further configured to synchronize a user’s profile information (“**the cognitive model**”) with a user profile (“**a cognitive model**”) of another device. (*Id.*) Kennewick provides express motivation to combine, explaining that “[a] user may use the system on one or more devices.” (*Id.* (quoting Kennewick, ¶0043).) A POSITA would have understood that the ability to synchronize user profile information across devices would beneficially enable the system to provide a more consistent and predictable experience for the user. (*Id.*)

12. Independent Claim 12

Claim 12 recites the method counterpart to claim 1. (Schmandt, ¶134 (side-by-side chart comparing claims 1 and 12).) The preamble of claim 1 recites “one or more processors,” while claim 12 recites “one or more processors” in each step. This is not a meaningful difference because, as explained for claim 1, it would have been obvious for the “one or more processors” (such as processors in a PC or PVA) to perform each of the recited steps. (Schmandt, ¶135.) Therefore, claim 12 is disclosed and obvious for the same reasons as claim 1. (*Id.*)

13. Claim 13: “The method of claim 12, wherein the cognitive model is a first cognitive model, and wherein the generating the speech-based transcription is based on a second cognitive model that includes information on one or more interactions between the device and a plurality of users of the device.”

Claim 13 is substantially the same as claim 5. (Schmandt, ¶136.) Therefore, for the reasons discussed for claim 5, claim 13 is also disclosed by and obvious over the prior art. (*Id.*) **Ground 2** explains how generating a speech-based transcription based on a “**second cognitive model**” would have also been obvious in further view of **Lee (EX1006)**.

14. Independent Claim 14

Claim 14 is similar to claim 1. (Schmandt, ¶137 (side-by-side chart comparing claims 1 and 14).) Claim 14, however, does not recite limitations related to “non-speech.” For example, claim 14 recites only “receiv[ing] a natural language

utterance” while claim 1 recites “receiv[ing] a multi-modal natural language input” that includes “a non-speech input.” Similarly, claim 14 recites a “speech-based transcription” rather than a “merged transcription” generated from a speech-based transcription and a non-speech transcription.

While claim 14 omits claim 1’s limitations related to “non-speech,” it adds limitations related to “general cognitive model” that are substantially similar to limitations related to a “second cognitive model” recited in claim 5. (Schmandt, ¶139.) Like the “second cognitive model” of claim 5, which “includes information on one or more interactions between the device and a plurality of users of the device,” the “general cognitive model” of claim 14 “includes information on one or more prior interactions between the device and a plurality of users.”

As discussed for claim 5, the “second cognitive model” corresponds to the collection of other users’ profile information, and the claimed “prior interactions” include activities such as “the history of [the user’s] interactions with the system,” “their special use of terminology,” “history of questions asked,” “session histories” and spellings and pronunciations provided by users. (Kennewick, ¶¶0114, 0029, 0151.) With respect to claim 5, therefore, the claimed “**personal cognitive model**” corresponds to the specific user’s profile information, and the “**general cognitive model**” corresponds to the collection of other users’ profile information. (Schmandt, ¶140.)

Therefore, for the reasons discussed above for claim 1 and claim 5, claim 14 is disclosed by and obvious over the prior art. (Schmandt, ¶141.) **Ground 2** explains how generating a speech-based transcription based on a “**general cognitive model**” would have also been obvious in further view of **Lee** (EX1006).

15. Claim 15: “The device of claim 14, wherein the one or more processors are configured to generate the speech-based transcription further based on an environmental model, wherein the environmental model includes information indicative of how noisy the environment of the user is.”

Claim 15 is an amalgam of claims 6 and 7, which together recite “generat[ing] the speech-based transcription...based on an environmental model” (claim 6) that “has information indicative of how noisy the environment of the user is” (claim 7). (Schmandt, ¶142.) Therefore, for the reasons discussed for claims 6 and 7, claim 15 is disclosed by and obvious over the prior art. (*Id.*)

C. Ground 2: Claims 5-7, 13-15 Are Obvious Over Ground 1 Prior Art In Further View Of Lee.

As discussed in the Ground 1 analysis, claims 5, 13 and 14 recite using a “**second cognitive model**” or “**general cognitive model**” to generate a speech-based transcription. The “**second cognitive model**” as recited in claims 5 and 13 “**includes information on one or more interactions between the device and a plurality of users of the device,**” and the “**general cognitive model**” similarly “**includes information on one or more prior interactions between the device and a plurality of users.**” As explained, Kennewick discloses the claimed “**second**

cognitive model” and **“general cognitive model”** in the form of profile information about other users, which includes users’ prior interactions with the system, such as “history of questions asked” and “session histories.” (Kennewick, ¶¶0114, 0029, 0149, 0151.) As discussed, using other users’ profile information when performing speech recognition could help the system recognize likely input from the user. (Kennewick, ¶0149.)

Lee (EX1006) offers more detail and provides confirmation that it would have been obvious to use other users’ profile information involving users’ interactions with the device, such as prior questions and commands, in connection with performing speech recognition. (Schmandt, ¶144.) Lee describes techniques for assisting accurate speech recognition of a user by using not only information collected about that user but also information collected about other users. (Lee, Abstract, ¶¶0002, 0016, 0022 (describing using user’s dialogue history), Fig. 3 (showing dialog history **148** and popularity engine **144**).) Lee describes a “multi-scan control unit **32**” that may use “recognition assisting databases **48**” to supplement recognition of speech input, including a database that “maintains a personalized profile that have already been recognized for a particular user or for users who have previously submitted requests which are similar to the request at hand.” (Lee, ¶0016.) Lee also describes a “popularity engine **144**” that “contains data compiled from multiple users’ histories that has been calculated for the

prediction of likely user requests.” (Lee, ¶0024.) Lee explains that “[t]his increases the accuracy of word recognition”; for example, with this information, the system can “scan[] for the most probable match for words in the user utterance based on past requests.” (Lee, ¶¶0024, 0026.)

Rationale and Motivation to Combine (Lee with Ground 1 Prior Art). It would have been obvious to combine Lee’s teachings regarding using information collected about multiple users to improve speech recognition accuracy with the conversational system as taught by the combination of Coffman, Kennewick and Ross described in connection with Ground 1. (Schmandt, ¶145.) The combination would have predictably resulted in Coffman’s conversational system using information on one or more prior interactions between a device and a plurality of users when performing speech recognition. (*Id.*) As discussed above, both Coffman and Kennewick already disclose collecting user profile information, including users’ prior interactions. (Kennewick, ¶¶0114, 0029, 0149, 0151; Coffman, 33:1-5, 37:8-10.) Incorporating Lee’s teachings to use prior interaction information for improved speech recognition would have therefore resulted in Coffman’s conversational system generating a speech-based transcription using a “**second cognitive model**” (claims 5 and 13) and “**general cognitive model**” (claim 14) as claimed. (Schmandt, ¶145)

Lee is an analogous reference in the same field as the '607 patent and would have been reasonably pertinent to problems facing the '607 patent inventors, including techniques for improving speech recognition accuracy, including in multi-user systems. ('607, 1:20-25; Lee, Abstract, ¶0002; Schmandt, ¶146.)

A POSITA would have been motivated to combine. (Schmandt, ¶147.) As discussed above, Lee, Coffman and Kennewick all describe collecting similar profile information about users, such as dialog history. (Lee, ¶¶0016, 0024, 0026; Kennewick, ¶¶0029, 0114, 0149-0151; Coffman, 33:1-5, 37:8-10.) Kennewick also describes using user profiles for improved word recognition accuracy. (Kennewick, ¶0149.) Lee provides further detail and confirmation that using information collected about other users when performing speech recognition can improve speech recognition accuracy. (Lee, ¶¶0016, 0024, 0026; Schmandt, ¶147.) A POSITA therefore would have been motivated to incorporate Lee's teachings into Coffman's conversational system to obtain the same benefits and advantages. (Schmandt, ¶147.)

A POSITA would have had a reasonable expectation of success, as the combination would have involved the application of known technologies and techniques without undue experimentation. (Schmandt, ¶148 (quoting Kennewick, ¶0006; EX1008, p.0024 ("adaptive language model" entry)).)

Petition for *Inter Partes* Review of
U.S. Patent No. 8,447,607 B2

VII. CONCLUSION

Petitioner respectfully requests IPR institution.

Dated: April 5, 2024

Respectfully submitted,

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW
Suite 700
Washington, DC 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

By: / Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this petition complies with the type-volume limits of 37 C.F.R. § 42.24(a)(1)(i) because it contains 13,817 words, according to the word-processing system used to prepare this petition, excluding the parts of this petition that are exempted by 37 C.F.R. § 42.24(a) (including the table of contents, a table of authorities, mandatory notices, a certificate of service or this certificate word count, appendix of exhibits, and claim listings).

DATED: April 5, 2024

COOLEY LLP
ATTN: Patent Docketing
1299 Pennsylvania Avenue NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

/ Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673

CERTIFICATE OF SERVICE

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 8,447,607 B2**, including all exhibits (**Nos. 1001-1013**) and related documents, are being served via Federal Express on the 5th day of April, 2024, the same day as the filing of the above-identified document in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon Patent Owner by serving the correspondence address of record with the USPTO as follows:

174899 - The Law Offices of David A. Gerasimow, P.C.
211 W. Wacker Dr.
Ste. 1717
Chicago, IL 60606
UNITED STATES

And, via Federal Express upon counsel of record for Patent Owner in the litigation pending before the U.S. District Court for the District of Delaware entitled *Dialect, LLC v. Google LLC*, Case No. 1:23-cv-00378 (D. Del.) as follows:

Garland Stephens
BLUE PEAK LAW GROUP LLP
3139 W Holcombe Blvd, PMB 8160
Houston, TX 77025
Telephone: 281-972-3036
garland@bluepeak.law

DATED: April 5, 2024

/ Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673
COOLEY LLP
1299 Pennsylvania Ave. NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001