

Petition for *Inter Partes* Review of
U.S. Patent No. 8,849,652 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

DIALECT, LLC,
Patent Owner

Case IPR2024-00752
U.S. Patent No. 8,849,652 B2
Issue Date: September 30, 2014

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

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,849,652 B2**

Table of Contents

	Page
I. MANDATORY NOTICES UNDER §42.8(A)(1)	11
A. Real Party-In-Interest under §42.8(b)(1).....	11
B. Related Matters under §42.8(b)(2)	11
C. Lead and Back-Up Counsel under §42.8(b)(3)	12
D. Service Information	13
II. FEE PAYMENT	13
III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D).....	13
A. Grounds for Standing	13
B. Identification of Challenge and Statement of Precise Relief Requested	13
C. Considerations Under §§ 314(a) and 325(d).....	14
IV. OVERVIEW OF THE CHALLENGED PATENT.....	16
A. Person of Ordinary Skill in the Art (POSITA)	16
B. Specification Overview	16
V. CLAIM CONSTRUCTION	18
VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	18
A. Overview of Grounds	18
B. Ground 1: Claims 1-9, 14-18, 20-26 Are Obvious Over Coffman.....	19
1. Claim 1: “A system for processing natural language utterances where recognized words of the natural language utterances alone are insufficient to completely determine one or more commands or requests, the system comprising:” (1[pre])	19

Table of Contents
(continued)

Page

(a)	“one or more physical processors programmed with one or more computer program instructions which, when executed, cause the one or more physical processors to:” (1[a]).....	24
(i)	“generate a first context stack associated with a first device, the first context stack comprising context information that corresponds to a plurality of prior utterances;” (1[a][i]).....	25
(ii)	“synchronize the first context stack with a second context stack associated with a second device such that the context information of the first context stack is updated based on related context information of the second context stack;” (1[a][ii]).....	30
(iii)	“receive a natural language utterance associated with a command or request;” (1[a][iii]).....	32
(iv)	“determine one or more words of the natural language utterance by performing speech recognition on the natural language utterance; and” (1[a][iv]).....	33
(v)	“determine the command or request based on the one or more words and the updated context information.” (1[a][v]).....	35
2.	Claim 2: “The system of claim 1, wherein the one or more physical processors are further caused to:”	37
(a)	“prompt a user associated with the natural language utterance for one or more of (i) additional information regarding the command or request or (ii) confirmation regarding the command or request; and” (2[a]).....	38

Table of Contents
(continued)

Page

(b)	“receive a non-speech input regarding one or more of the additional information or the confirmation in response to the prompt, wherein the command or request is determined further based on the non-speech input.” (2[b])	41
3.	Claim 3: “The system of claim 1, wherein the first context stack includes a plurality of context entries, and wherein the one or more physical processors are further caused to:”	42
(a)	“identify, from among the plurality of context entries, one or more context entries that correspond to the one or more words, wherein the updated context information includes the one or more context entries.” (3[a])	43
4.	Claim 4: “The system of claim 3, wherein identifying the one or more context entries comprises:”	44
(a)	“comparing the plurality of context entries with the one or more words;” (4[a])	45
(b)	“generating one or more rank scores for individual context entries of the plurality of context entries based on the comparison; and” (4[b])	45
(c)	“identifying, from among the plurality of context entries, the one or more context entries based on the one or more rank scores.” (4[c])	49
5.	Claim 5: “The system of claim 4, wherein the plurality of context entries are ordered in the first context stack, and wherein the one or more physical processors are further caused to update the ordering of the plurality of context entries in the first context stack based on the one or more rank scores.”	49

Table of Contents
(continued)

Page

6.	Claim 6: “The system of claim 5, wherein the natural language utterance is associated with a user, and wherein the one or more physical processors are further caused to:”	50
(a)	“receive a subsequent natural language utterance associated with the user; and” (6[a])	50
(b)	“determine one or more subsequent words of the subsequent natural language utterance by performing speech recognition on the subsequent natural language utterance;” (6[b]).....	51
(c)	“identify, from among the plurality of context entries, one or more other context entries that correspond to the one or more subsequent words based on the updated ordering; and” (6[c])	51
(d)	“determine a subsequent command or request based on the one or more subsequent words and the one or more other context entries.” (6[d])	52
7.	Claim 7: “The system of claim 3, wherein identifying the one or more context entries comprises identifying, from among the plurality of context entries, the one or more context entries that most closely correspond to the one or more words.”	52
8.	Claim 8: “The system of claim 3, wherein the one or more physical processors are further caused to:”	53
(a)	“identify one or more domain agents associated with the one or more context entries, wherein the one or more domain agents are configured to process the command or request; and” (8[a]).....	53
(b)	“generate a response to the command or request using the one or more domain agents.” (8[b]).....	56

Table of Contents

(continued)

	Page
9. Claim 9: “The system of claim 8, wherein each of the one or more domain agents comprises domain knowledge associated with a particular domain, the domain knowledge comprising: one or more of (i) a keyword; (ii) a link to an information source; (iii) a list of responses associated with a plurality commands or requests; (iv) a substitution list used to format the plurality of commands or requests; or (v) content including dictionaries, encyclopedias, or almanacs.”	58
10. Independent Claim 14	59
11. Claims 15-18	59
12. Independent Claim 20	59
13. Claims 21-26	60
C. Ground 2: Claims 10-13, 19 and 27 Are Obvious Over Coffman and Kennewick.....	60
1. Claim 10: “The system of claim 8, wherein the response comprises an aggregation of one or more responses generated by the one or more domain agents based on the command or request.”	60
2. Claim 11: “The system of claim 8, wherein the one or more physical processors are further caused to:”	64
(a) “obtain information relating to a license agreement that is associated with at least one of the one or more domain agents;” (11[a]).....	64
(b) “determine, based on the information related to the license agreement, that use of the at least one of the one or more domain agents to process the command or request is permitted; and” (11[b])	68
(c) “use, based on the determination that the use is permitted, the at least one of the one or more domain agents to generate the response.” (11[c])	68

Table of Contents
(continued)

	Page
3. Claim 12: “The system of claim 1, wherein the natural language utterance is associated with a user, and wherein the one or more physical processors are further caused to:” (12[pre])	69
(a) “obtain one or more of a cognitive model or an environmental model associated with the user, wherein the cognitive model comprises information relating to one or more interactions between the user and the system, and the environmental model comprises information indicative of how noisy an environment surrounding the user is,” (12[a]).....	69
(b) “wherein the one or more words are determined further based on one or more of the cognitive model or the environmental model.” (12[b]).....	74
4. Claim 13: “The system of claim 1, wherein the natural language utterance is associated with a user, and wherein the one or more physical processors are further caused to:” (13[pre])	74
(a) “obtain a first cognitive model that comprises information relating to one or more interactions between the user and the system; and” (13[a]).....	75
(b) “obtain a second cognitive model that comprises information relating to one or more interactions between the system and a plurality of users of the system,” (13[b])	75
(c) “wherein the one or more words are determined further based on the first cognitive model and the second cognitive model.” (13[c])	77
5. Claim 19	78
6. Claim 27	78
D. Ground 3: Claims 13 and 19 Are Obvious Over Coffman, Kennewick and Lee.	78

Table of Contents
(continued)

	Page
VII. CONCLUSION.....	81
CERTIFICATE OF SERVICE	83

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 8,849,652 B2 to Chris Weider er al. (filed May 20, 2013, issued September 30, 2014) (“ 652 ” or “ 652 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	Excerpts from <i>Random House Webster’s Unabridged Dictionary</i> (2001)
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	International Patent Application Publication No. WO 00/21075 to Stephane H. Maes et al. (filed October 1, 1999, published April 13, 2000) (“ Maes ”)
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 Cezar Borkowski and Marion Manzo, <i>The Complete Idiot’s Guide to Martial Arts</i> (1999)
1011	Declaration of Ingrid Hsieh-Yee, Ph.D.
1012	Prosecution history of U.S. Patent No. 8,849,652 B2

Petition for *Inter Partes* Review of
U.S. Patent No. 8,849,652 B2

List of Exhibits

Exhibit No.	Description of Document
1013	Proof of Service of Complaint, <i>Dialect, LLC v. Google LLC</i> , Case No. 1:23-cv-00378 (D. Del.)
1014	International Preliminary Examination Report for Maes (EX1007) application

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 ’652 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). Petitioner was first served with a complaint alleging infringement of the ’652 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.). The ’652 patent was also the subject of *Dialect, LLC v. Samsung Electronics, Co.*, Case No. 2:23-cv-00061-JRG (E.D. Tex.), which was terminated on September 11, 2023.

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
7,502,738 B2	IPR2024-00747
8,015,006 B2	IPR2024-00748

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. 8,849,652 B2

U.S. Patent No.	IPR Case
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
	Reuben Chen (Admission <i>pro hac vice</i> to be requested) rchen@cooley.com COOLEY LLP

LEAD COUNSEL	BACK-UP COUNSEL
	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 '652 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 27 claims, with a \$49,375 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 '652 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 of claims 1-27:

Petition for *Inter Partes* Review of
U.S. Patent No. 8,849,652 B2

Ground	Claims	Basis for Challenge under §103
1	1-9, 14-18, 20-26	Coffman (EX1003)
2	10-13, 19, 27	Coffman and Kennewick (EX1004)
3	13, 19	Coffman, Kennewick and Lee (EX1006)

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

§314(a): Petitioner is unaware of another post-grant challenge against the '652 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 pending litigation.

§325(d): During '652 patent prosecution, the proposed claims were initially rejected as obvious over U.S. Patent App. Pub. No. 2002/0133354 ("Ross"). (EX1012, pp.00150-00174.) The Examiner's Notice of References Cited accompanying the rejection listed the published divisional application (U.S. Patent App. Pub. No. 2007/0043574 A1) of Coffman's U.S. patent counterpart (U.S. 7,137,126) among twelve references, but the examiner did not otherwise discuss it. (EX1012, p.00175.) In response to the rejection, the applicant amended the pending claims. (EX1012, pp.00185-00199.) The applicant also submitted an IDS that identified both Coffman's U.S. counterpart (U.S. 7,137,126) and Kennewick among approximately 675 references. (See EX1012, pp.00216, 00224; see generally *id.*,

pp.00200-00229) A Notice of Allowance followed without any further rejections. (EX1012, pp.00441-00449.) The Notice of Allowance identified U.S. Patent App. Pub. No. 2003/0046346 (“Mumick”) and its teachings regarding “synchronizing plural browsers on a plurality of mobile devices,” and observed that “synchronization of navigation state is not the claimed synchronization of context across multiple devices.” (EX1012, p.00447.) The Notice of Allowance contrasted Mumick with claim language where “not only is context synchronized, but context stacks/sets of distinct devices are synchronized.” (EX1012, p.00447 (emphasis in original).) Because neither of Coffman’s related U.S. patent publications nor Kennewick were discussed during prosecution, it is unclear why the Examiner did not appreciate their significance. But Petitioner respectfully submits that the Examiner materially erred in failing to appreciate them. As discussed below, Coffman teaches and renders obvious the challenged claims, including limitations related to synchronizing context information and/or context stacks/sets that the Examiner identified as missing from the Mumick prior art reference. (*See* VI.C.1.a.ii *infra*; *see also* Coffman, e.g., 25:10-14 (“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”), 27:7-20.)

IV. OVERVIEW OF THE CHALLENGED PATENT

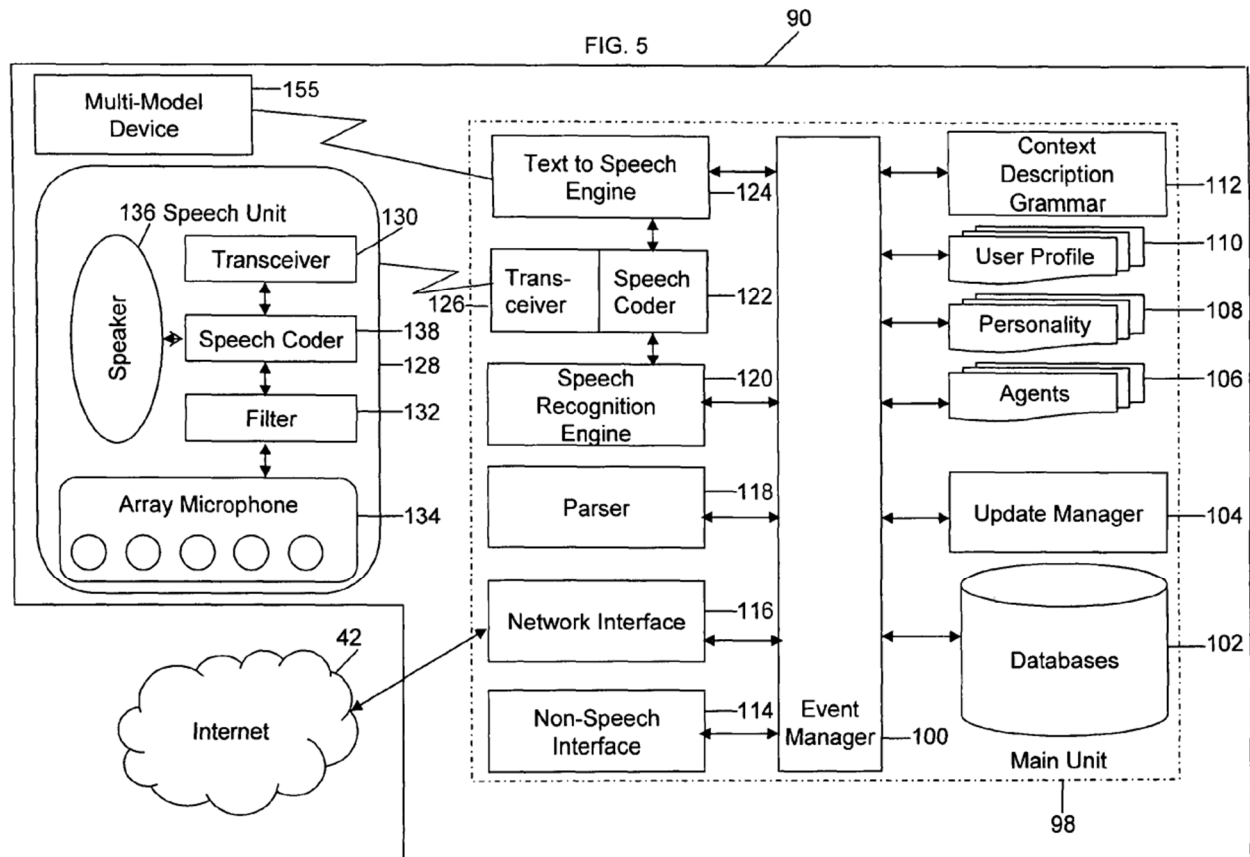
A. Person of Ordinary Skill in the Art (POSITA)

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 '652 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." ('652, 1:22-28; *see also id.*, 1:67-2:4 (acknowledging prior speech recognition and natural language processing).)

"FIG. 5 is an overall diagrammatic view of the interactive natural language processing system according to an embodiment of the invention" ('652, 15:38-40):



(’652, Fig. 5.) “The system **90** may include a main unit **98**, speech units **128**, and multi-modal devices **155**” (such as “mobile devices, stand alone or networked computers, personal digital assistances (PDAs), portable computer devices, or other multi-modal devices”). (’652, 23:38-39, 25:60-65.) “Agents **106** may include a collection of grammars, criteria handlers, and algorithms that are accessible to respond to a set of requests and/or commands.” (’652, 25:32-34.)

Speech unit **128** includes a microphone to receive a spoken utterance from a user. (’652, 25:50-52.) After initial processing, the speech is passed to speech recognition engine **120** for processing. (’652, 26:18-20.) After speech recognition,

parser **118** “may determine a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, dialog history, and context stack contents.” (’652, 34:2-5.) “For example, a question with the keywords ‘temperature’ implies a context value of weather for the question.” (’652, 34:7-8.) “Once the context for the question or command has been determined, the parser **118** can invoke the correct agent **156, 150**.” (’652, 34:34-36.) A context stack “includes an ordered list of command contexts, to enable follow-up requests.” (’652, 20:10-12; *see also id.*, 36:13-14 (“Part of the dialog history is maintained in a context stack.”).) “The other registered mobile devices may be informed of context changes through a context tracking module to enable synchronizing of context across the registered modules.” (’652, 4:29-32.)

V. CLAIM CONSTRUCTION

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

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Overview of Grounds

The challenged claims attempt to lay claim to well-known techniques for speech-enabled interfaces for interacting with and directing computer systems. The challenged claims employ standard techniques that were known long before the ’652

patent, such as using speech recognition and natural language processing to select an application for carrying out a query and/or command included in a spoken utterance.

All grounds rely primarily on Coffman (EX1003), which despite being filed nearly six years before the '652 patent, discloses a remarkably similar conversational computing system. Ground 2 combines Coffman with Kennewick (EX1004) for features recited in claim 11 (and counterpart method claim 27) related to licensing and features recited in claims 12-13 and 19 related to a “cognitive model.” Ground 3 further combines with Lee (EX1006), also for features related to a “cognitive model.”

Coffman, Kennewick and Lee are prior art to the '652 patent under at least pre-AIA §102(b). Kennewick ultimately issued as U.S. Patent No. 7,398,209 B2, which Petitioner is also challenging in IPR, as noted in section I.B above.

B. Ground 1: Claims 1-9, 14-18, 20-26 Are Obvious Over Coffman.

- 1. Claim 1: “A system for processing natural language utterances where recognized words of the natural language utterances alone are insufficient to completely determine one or more commands or requests, the system comprising:” (1[pre])**

Assuming the preamble imposes a claim limitation, it is disclosed and rendered obvious by the prior art. As explained in more detail for the claim limitations below, Coffman discloses a conversational computing “system.”

(Coffman, 36:29-31, 5:1-4, 13:8-11.) The system is “**for processing... utterances**”—Coffman explains that the system processes spoken input from a person. (Coffman, e.g., 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.”), 41:2-5, 53:22-26; Schmandt, ¶58.)² Figure 6 of Coffman illustrates:

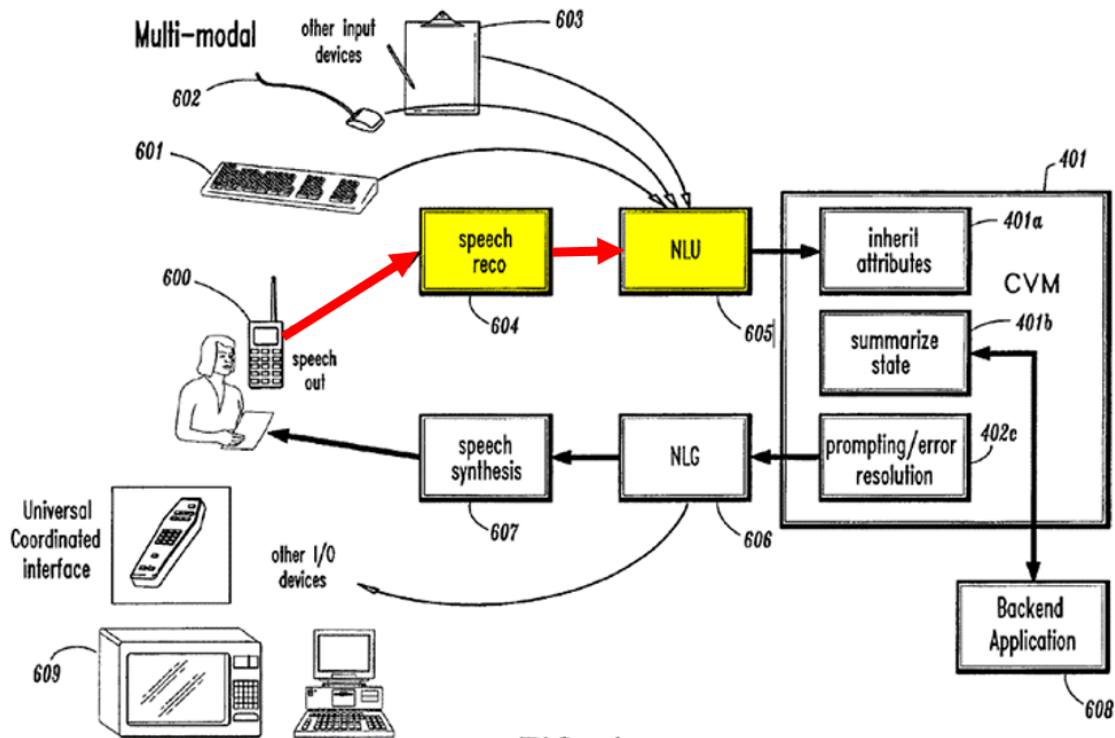


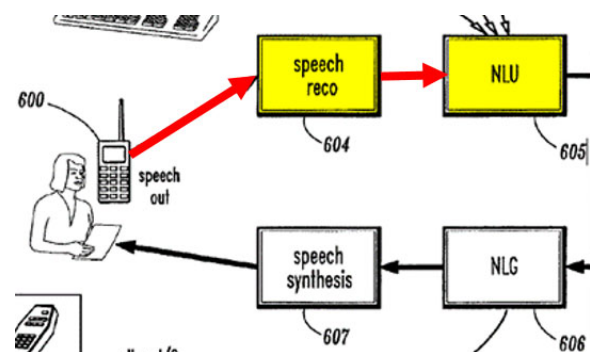
FIG. 6

² All underlining added to quoted material unless noted.

(Coffman, Fig. 6 (annotations added); *see also id.*, 37:14-22 (“Spoken input from a speech client (e.g., telephone 600)....”).)

Coffman further discloses that its conversational computing system processes “**natural language utterances.**” In the context of the ’652 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 and programming languages. (Schmandt, ¶59 (citing Gibbon (EX1008), p.0035 (defining “natural language”)).) For example, English is a natural language, while C++ is a programming language. Consistent with this understanding, 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 (“e-mail from my mother is correctly understood”), 56:25-57:2 (“grocery list,” “it should be synchronized with my PDA,” and “Do it!”); Schmandt, ¶59.)

Coffman’s processing 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, 6:6-8, 5:1-4, 37:29-32; Schmandt, ¶60.)

Coffman further discloses a system where “**recognized words of the natural language utterances alone are insufficient to completely determine one or more commands or requests.**” This portion of the preamble describes nothing more than a natural language-based system, such as Coffman’s, that enables a user to input application commands or requests without having to recite the exact syntax of the command or request. (Schmandt, ¶61.) For instance, as noted above, Coffman describes the system receiving the natural language speech utterance “I would like to open this,” along with “a mouse click or pointer/stylus pointing action,” and disambiguating that input “into a set of attribute value pairs: Command: open, Object: Windows or task selected by the last mouse click.” (Coffman, 6:32-7:3.) Coffman explains that determining the command or request requires more than simply recognizing the words in the user’s utterance, “I would like to open this.” It

requires natural language processing and use of a “context stack” to determine the particular application and command that should be invoked, as will be explained in more detail in connection with 1[a][v]. (Coffman, e.g., 29:20-32 (describing context stack), 42:1-5; Schmandt, ¶61.) It also requires what Coffman refers to as “multi-modal processing” involving speech being used in parallel with mouse, keyboard or other input devices:

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. A summarization process 401b is performed whereby the attribute value n-uples are added to the context and then verifies with the syntax of the back-end application 608 whether the query is complete, incomplete, or ambiguous.

(Coffman, 37:15-26; *see also id.*, 11:23-24 (“This is the notion of ‘multi-modality’ whereby speech is used in parallel with mouse, keyboard, and other input devices

such as a pen.”).) Using Coffman’s example, “I would like to open this,” the system determines the command in part by processing additional input—i.e., the user’s “mouse click or pointer/stylus pointer action”—to identify the object of the user’s command—i.e., “Windows or task selected by the last mouse click.” (Schmandt, ¶61.)

Coffman therefore discloses “[a] **system for processing natural language utterances where recognized words of the natural language utterances alone are insufficient to completely determine one or more commands or requests.**”

And as shown below, the claimed system is obvious.

- (a) **“one or more physical processors programmed with one or more computer program instructions which, when executed, cause the one or more physical processors to:”**
(1[a])

1[a] 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 Ex. 1009, pp.0024-0025 (“microprocessor”), p.0033 (“software”), p.0028 (“program”), p.0016 (“execute” and “executable program”).) 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 Coffman’s conversational computing

system would include “**one or more physical processors programmed with one or more computer program instructions.**” (Schmandt, ¶63.) As explained for the following limitations, a POSITA would have further understood and found it obvious that executing the computer program instructions would cause the processor(s) to perform the claimed steps. (*Id.*)

- (i) **“generate a first context stack associated with a first device, the first context stack comprising context information that corresponds to a plurality of prior utterances;” (1[a][i])**

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.” (29:18-30:3 (“The core of the CVM 401 is the context stack 405...”), 30:4-10.) Coffman’s context stack comprises “**context information that corresponds to a plurality of prior utterances.**” Coffman explains that a context stack “accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) or each active process.”³ (Coffman, 29:20-27; Schmandt,

³ An “n-uple” refers to a collection of “n” elements, where “n” is a number. Therefore, for example, a “4-uple” (or “quadruple”) refers to a collection of four elements. (Schmandt, ¶65 n.4.) Coffman’s “attribute value n-uple” refers to an n-uple having “n” attribute-value pairs. Coffman’s example, “Command: open, Object: Windows or task selected by the last mouse click” based on the user’s utterance “I would like to open this” (Coffman, 6:32-7:3), is a 2-uple (or “duple”) because it includes two attribute-value pairs, (1) “Command: open” and (2) “Object: Windows or task selected by last mouse click.” (Schmandt, ¶65 n.4.) These two attribute-value n-uples together are an example of a query argument list of attribute-value n-uples. (*Id.* (citing Coffman, 6:8-10, 6:26-28).)

¶65.) Coffman further explains that “‘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.” (Coffman, 29:27-32.) Coffman therefore makes clear that a context stack comprises context information corresponding to a plurality of prior utterances, such as full query arguments list of attribute value n-uples for multiple applications that were derived from a user’s spoken input. (Schmandt, ¶65; Coffman, e.g., 41:2-5 (“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.”), 37:20-26 (“Each input is subject to attribute acquisition (401a) whereby the attribute value n-uples are acquired from the input. A summarization process 401b is performed whereby the attribute value n-uples are added to the context and then verifies with the syntax of the back-end application 608 whether the query is complete, incomplete, or ambiguous.”).)

Coffman further discloses a “**first context stack associated with a first device**.” 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.)

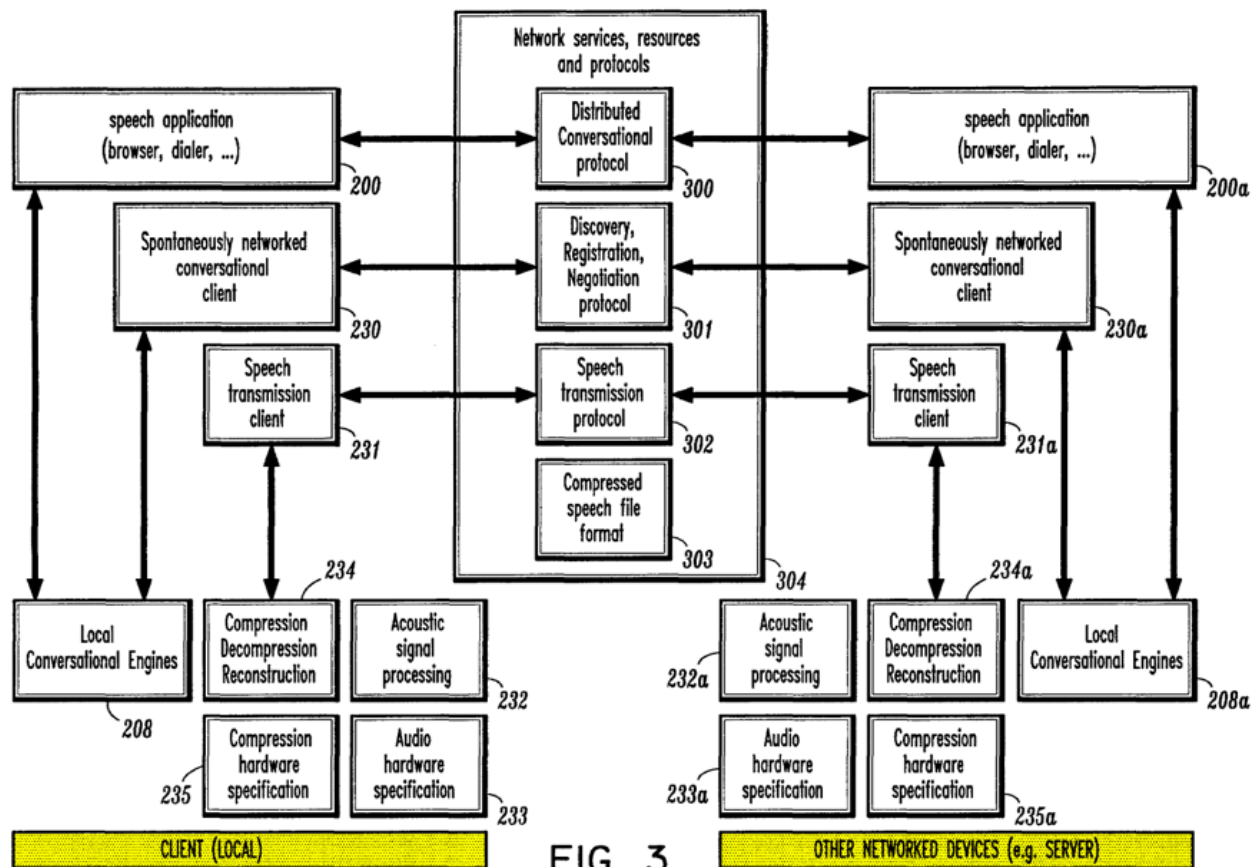


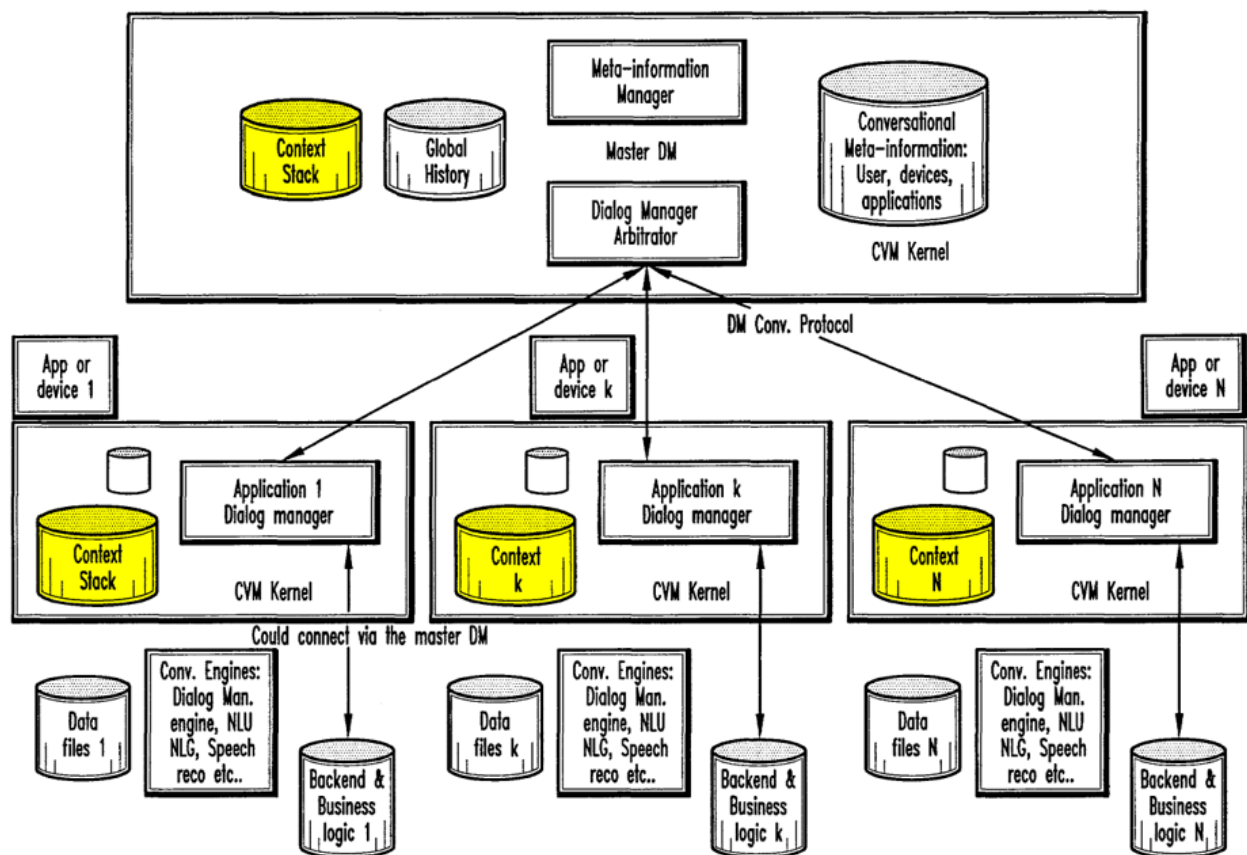
FIG. 3

(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”—i.e., each device has a “**context stack associated with [that] device**”:

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; Schmandt, ¶66.) 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).”).)

With respect to the claimed “**first device**,” Coffman explains that Figure 10 “illustrates a typical master slave topology,” which “is formed by exchanging relevant information via the DM [dialog manager] protocols.” (Coffman, 27:26-28.) In Coffman, the “**first device**” could correspond to the device in a master slave topology having the master/central dialog manager, which is “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, 27:5-7; *see also id.*, 29:18-20 (context stack operates and is managed under control of dialog manager/controller), 28:24-27 (dialog manager also referred to as dialog controller); Schmandt, ¶67.)

Lastly, referring back to 1[a], with respect to “**physical processor(s)**” executing “**computer program instructions**” to “**generate**” the claimed context stack, a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to generate the context stack, consistent with the routine and conventional operation of generic computer systems, such as PCs and PDAs, as discussed for 1[a]. (Schmandt, ¶68.) Moreover, as noted, for example, Coffman explains that the context stack is operated

and managed under the control of dialog manager software, which receives all input and output events across all modalities and “stores it appropriately in the context stack.” (Coffman, 29:18-20, 37:18-20; *see also id.*, 19:26-29 (describing dialog manager as “programming layer”), 30:4-10 (“More specifically, each new task, process, or thread creates a new stack entry and is associated with a discourse.”); Schmandt, ¶68.)

- (ii) **“synchronize the first context stack with a second context stack associated with a second device such that the context information of the first context stack is updated based on related context information of the second context stack;”**
(1[a][ii])

As discussed for 1[a][i], Coffman discloses a distributed conversational computing system having a “master slave topology” where a master/central device coordinates with one or more slave devices, each device having its own context stack. (*E.g.*, Coffman, Figs. 3 & 10, 27:1-7, 27:24-26.) The claimed **“second context stack associated with a second device”** therefore corresponds in Coffman to a context stack associated with a slave device in a master slave topology. (Schmandt, ¶69.)

Coffman further makes clear that a master device’s context stack (**“the first context stack”**) is **“synchronized”** with a slave device’s context stack (**“a second context stack associated with a second 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, ¶70.) 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 further understood and found it obvious that, as a result of synchronizing device context stacks in Coffman’s system, **“the context information of the first context stack is updated based on related context information of the second context stack.”** (Schmandt, ¶71.) The context information in the two context stacks would be **“related”** in at least two ways in Coffman. (Schmandt, ¶71.) First, the context information is **“related”** because, as Coffman explains, it can comprise **“context updates,”** which would update previous, related context information in the device’s context stack. Second, more broadly, the context information is **“related”** because the associated devices themselves are related as a result of the master slave topology relationship. The context information of the first stack would be **“updated”** based on the context information of the second context stack both by adding information to the first

context stack and by updating existing information in the first context stack.
(Schmandt, ¶71.)

Lastly, for substantially the same reasons as those explained for 1[a][i], a POSITA would have understood and found it obvious that the device's processor(s) would execute software (computer program instructions) to synchronize the first context stack as claimed. (Schmandt, ¶72.)

(iii) “receive a natural language utterance associated with a command or request;” (1[a][iii])

As discussed for the preamble, Coffman discloses “**receiv[ing] a natural language utterance,**” such as “I would like to open this.” (Coffman, 6:32-7:3; *see also id.*, 48:1-3 (“e-mail from my mother”), 56:25-57:2 (“grocery list,” “it should be synchronized with my PDA,” and “Do it!”).)

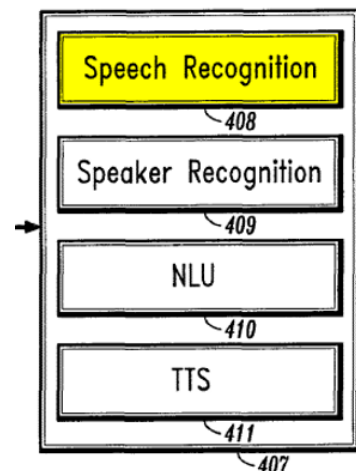
Coffman further discloses that the utterance is “**associated with a command or request.**” (Schmandt, ¶74.) For example, Coffman explains that the user's utterance, “I would like to open this,” is associated with a command or request for the system to perform an “open” action. (Coffman, 6:32-7:3 (“For 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.”).) Coffman more generally explains that the system “convert[s] voice requests into queries.” (Coffman, 13:8-13 (“In general, conversational sub-systems 18 are responsible for

converting voice requests into queries and converting outputs and results into spoken messages using the appropriate data files 17 (e.g., contexts, finite state grammars, vocabularies, language models, symbolic query maps etc.)”); *see also id.*, 38:26-27 (“[T]he input procedure provides a set of multi-mode input streams, each transcribed into an ASCII command, query, or list of attribute value n-uples.”).)

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to receive a natural language utterance as claimed in order to carry out further processing such as speech recognition and natural language understanding in Coffman’s computer-based system. (Schmandt, ¶75.)

- (iv) **“determine one or more words of the natural language utterance by performing speech recognition on the natural language utterance; and” (1[a][iv])**

Coffman’s system includes a “speech recognition engine.” (Coffman, 10:28-30 (“In general, the CVM provides conversational APIs and protocols between conversational subsystems (e.g. speech recognition engine, text-to speech etc.) and conversational and/or conventional applications.”), 29:1-4 (“The conversational system 400 further comprises a plurality of conversational resource subsystems (engines)



407 including, for example, a speech recognition system 408”), Fig. 4 (excerpted and highlighted right).) Coffman further makes clear that its speech recognition engine is used to “**determine one or more words of the natural language utterance by performing speech recognition on the natural language utterance,**” explaining that “[s]poken 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.” (Coffman, 37:20-22; *see also id.*, Fig. 6 (depicting input to and output from “speech reco” 604), 38:9-11; Schmandt, ¶76.) As noted above, Coffman describes various examples of utterances of words and phrases, including “I would like to open this,” “e-mail from my mother,” “grocery list,” and “it should be synchronized with my PDA.” (Coffman, 6:32-7:3, 48:1-3, 56:25-57:2.)

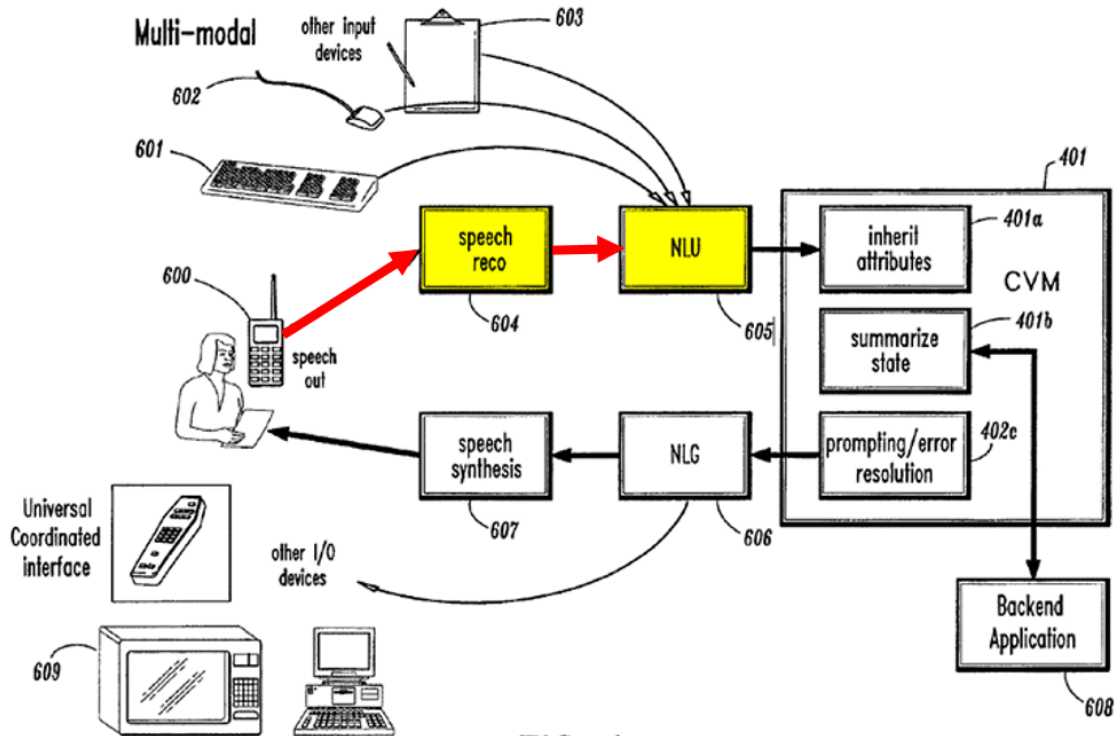


FIG. 6

(Coffman, Fig. 6 (annotations added); *see also id.*, 37:14-22 (“Spoken input from a speech client (e.g., telephone 600)...”).)

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to perform speech recognition on received natural language utterances. (Schmandt, ¶77.)

- (v) **“determine the command or request based on the one or more words and the updated context information.” (1[a][v])**

Coffman discloses that, in order to determine the command or request for the system to execute, recognized words in the utterance are matched against context information in the context stack:

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; *see also id.*, 41:2-5 (“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.”), 30:8-10 (“The context of a given discourse is also associated with, e.g., any active grammars, vocabularies and symbolic language which maps the actual query.”), 51:12-14 (“Queries can be decoded, parsed and then translated into a logic combination of queries (symbolic query) using NLU technology.”), 37:20-26 (“Each input is subject to attribute acquisition (401a) whereby the attribute value n-uples are acquired from the input. A summarization process 401b is performed whereby the attribute value n-uples are added to the context and then verifies with the syntax of the back-end application 608 whether the query is complete, incomplete, or ambiguous.”), 40:12-15; Schmandt, ¶78.)

A POSITA would have further understood and found it obvious that determining the command or request could take place after the master/central device had updated its context information, predictably resulting in determining the command or request **“based on the one or more words and the updated context information.”** (Coffman, e.g., 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. The task dispatcher and context stack accumulates the contexts associated to all the devices and will parse and dispatch commands to each device accordingly.”); Schmandt, ¶79.)

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to determine the command or request as claimed. (Schmandt, ¶80.)

2. Claim 2: “The system of claim 1, wherein the one or more physical processors are further caused to:”

As explained below, it would have been obvious in Coffman’s system for the device to perform the following steps. For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to perform those steps. (Schmandt, ¶81.)

- (a) **“prompt a user associated with the natural language utterance for one or more of (i) additional information regarding the command or request or (ii) confirmation regarding the command or request; and” (2[a])**

Coffman discloses prompting a user both (1) for **“additional information regarding the command or request,”** and (2) for **“confirmation regarding the command or request.”**

With respect to prompting a user for **“additional information regarding the command or request,”** Coffman explains that a user’s command/query that is deemed incomplete or ambiguous “will generate a request (a prompt) for the missing or ambiguous information and update the context (requested missing fields).” (Coffman, 41:16-21; *see also id.*, 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed...”), 40:9-19 (“provide feedback to the user for incomplete or ambiguous queries/command”); Schmandt, ¶83.)

With respect to prompting a user for **“confirmation regarding the command or request,”** Coffman explains that “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.” (Coffman, 42:1-4; *see also id.*, 41:30-32 (“Command may also require user confirmation before execution based on the preferences/settings coming from the CVM platform, application, or user preferences.”), 42:18-20 (“If the request is complete, it will be executed, pending possible request for confirmation by the user, e.g. when it is irreversible.”), 56:30

(“You may need to confirm the selection if your PC CVM requests it.”); Schmandt, ¶84.)

The prompted user in Coffman is “**associated with the natural language utterance**” for at least three reasons. (Schmandt, ¶85.) First, the prompted user is associated with the utterance because it is the same user who provided the utterance in the first place. (*Id.* (citing Coffman, e.g., 40:9-19, 41:16-21, 42:1-4, 42:18-20).) Second, Coffman also discloses identifying the user who spoke the utterance, thereby associating the utterance with a user. (*Id.* (citing Coffman, 5:1-4 & 47:25-27 (describing functionality for “speaker recognition” and “speaker identification,” such as “text-independent speaker identification”); citing EX1008, pp.0047 (defining “speaker recognition”), 0051 (defining “text-independent speaker recognition system”))).) For example, figure 4 of Coffman depicts conversational system 400 with speaker recognition system 409:

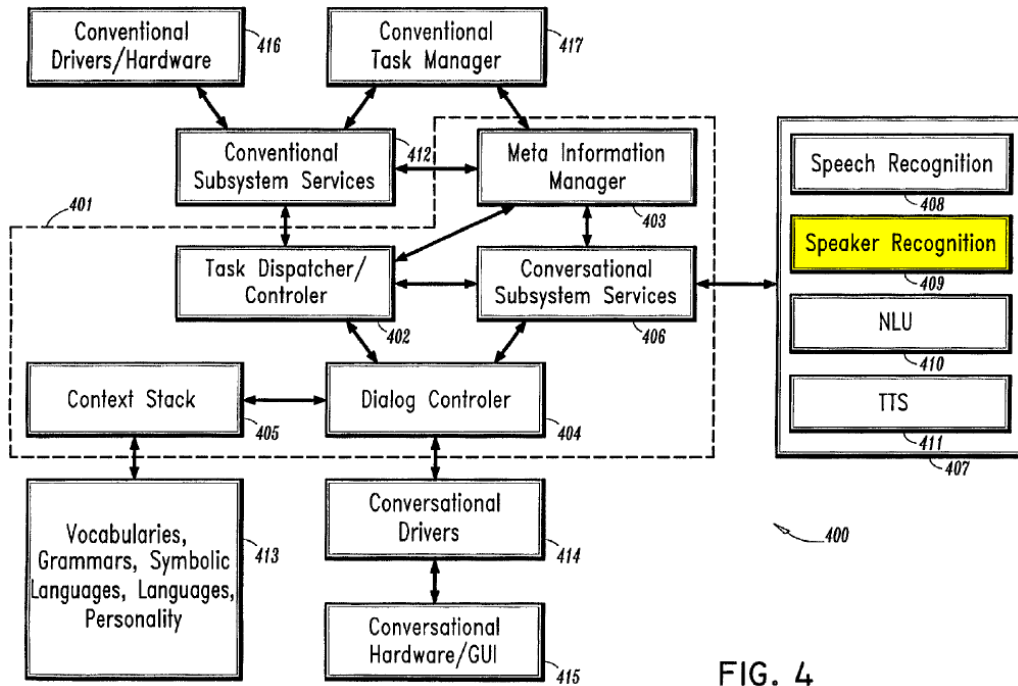


FIG. 4

(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’s speaker identification is therefore another way that Coffman discloses “**prompt[ing] a user associated with the natural language utterance**.” (Schmandt, ¶85.)

Coffman further explains that “user identity can be associated in the context of each discourse,” so that “command ambiguity can be immediately and transparently resolved (e-mail from my mother is correctly understood, independently of the user),” further confirming that a user is “**associated with the**

natural language utterance.” (Schmandt, ¶86 (quoting Coffman, 48:1-3, citing Coffman, 46:26-31, 48:11-14).)

- (b) **“receive a non-speech input regarding one or more of the additional information or the confirmation in response to the prompt, wherein the command or request is determined further based on the non-speech input.”** (2[b])

Coffman does not appear to detail the type of input Coffman’s conversational computing system receives in response to a prompt for additional information or confirmation. But it would have been obvious to a POSITA for the system to receive **“a non-speech input.”** (Schmandt, ¶87.) This is because Coffman describes a “multi-modal” system that can receive both speech input and non-speech input, such as a mouse click, to use in carrying out system operations. (*Id.* (citing Coffman 37:15-23 (“process multi-modal input...”), 11:23-24 (“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’ 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.”), 6:30-7:3.) In fact, Coffman expressly acknowledges that non-speech input can be used as an alternative to speech input, explaining that “[f]or example, with one conversational request (or by pressing a button on a keyboard or clicking a

mouse or providing a key), the user can listen to the output of a task or re-claim the input....” (Coffman, 35:2-4; Schmandt, ¶87.)

A POSITA would have also understood and found it obvious that “**the command or request is determined further based on**” the user’s input regarding the additional information or confirmation (“**the non-speech input**”). (Schmandt, ¶88.) As discussed for 2[a], Coffman explains that the additional information enables the system to complete or resolve any ambiguity in the user’s input so that the system can determine correct application and function to perform. (Coffman, 41:16-21, 37:2-4, 40:9-19.) Likewise, user confirmation allows the system to determine that it correctly understood user input and that it should actually carry out the operations that it prompted the user about. (Coffman, e.g., 42:18-20, 56:30.)

3. Claim 3: “The system of claim 1, wherein the first context stack includes a plurality of context entries, and wherein the one or more physical processors are further caused to:”

As discussed above for claim 1, Coffman explains that a context stack includes “**a plurality of context entries.**” (Coffman, 30:4-10 (“[E]ach new task, process, or thread creates a new stack entry and is associated with a discourse.”), 30:13-16 (“...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, 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.”).) A POSITA would have therefore understood and found it obvious that **“the first context stack”** would **“include a plurality of context entries.”** (Schmandt, ¶89.)

- (a) **“identify, from among the plurality of context entries, one or more context entries that correspond to the one or more words, wherein the updated context information includes the one or more context entries.”**
(3[a])

Coffman discloses **“identify[ing], from among the plurality of context entries, one or more context entries that correspond to the one or more words.”** (Schmandt, ¶90.) Coffman describes searching through a context stack to identify a context entry that matches the user’s utterance:

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.)

A POSITA would have further understood and found it obvious that identifying the one or more context entries could take place after the master/central device had updated its context information in its context stack, predictably resulting

in identifying a context entry “**wherein the updated context information includes the one or more context entries.**” (Schmandt, ¶91; Coffman, e.g., 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. The task dispatcher and context stack accumulates the contexts associated to all the devices and will parse and dispatch commands to each device accordingly.”).)

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to perform this step. (Schmandt, ¶92.)

4. Claim 4: “The system of claim 3, wherein identifying the one or more context entries comprises:”

As explained below, a POSITA would have understood and found it obvious for Coffman’s process of identifying one or more context entries to comprise the following steps.

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to perform these steps. (Schmandt, ¶94.)

(a) **“comparing the plurality of context entries with the one or more words;” (4[a])**

Coffman makes clear that its process of identifying a context entry that matches the user’s utterance involves **“comparing the plurality of context entries with the one or more words.”** (Schmandt, ¶95.) As just discussed for claim 3, Coffman describes comparing context entries in a context stack to the user’s recognized utterance. (Coffman, 42:1-5; *see also id.*, 41:2-5, 37:2-4.) For instance, taking Coffman’s example, “I would like to open this,” Coffman teaches that the system compares the recognized word “open” in the user’s utterance to a context entry for an application’s “open” command. (Schmandt, ¶95; Coffman, 6:32-7:3.)

(b) **“generating one or more rank scores for individual context entries of the plurality of context entries based on the comparison; and” (4[b])**

Coffman describes how identifying a context entry involves searching the context stack for a context entry match that is sufficiently “acceptable” and not “too improbable.” A context stack is searched to find a context that is “an acceptable match” by “fit[ting] the utterance at the level of the NLU symbolic language.” (Coffman, 42:1-5.) If the match with a context in the context stack is “too improbable,” then the next context in the context stack is considered for matching. (Coffman, 40:24-27.)

A POSITA would have understood and found it obvious that Coffman’s technique of determining whether a context entry was an “acceptable match” or “too

improbable” would involve “**scoring**” the context. (Schmandt, ¶97.) The concept of “scoring” has been well known in the field of speech recognition and processing since at least the 1970s. For example, Coffman incorporates by reference International Patent Application Publication No. WO 00/21075 (**Maes** [EX1007]) (Coffman, 22:25-29 (“...patent application IBM Docket No. YO999- 113P, filed concurrently herewith, entitled ‘System and Method for Providing Network Coordinated Conversational Services,’ which is commonly assigned and incorporated herein by reference...”); EX1014 (Maes application corresponds to “YO999-113P”).) Maes is therefore part of Coffman’s disclosures for purposes of applying it as prior art. *See, e.g., Paice, LLC v. Ford Motor Co.*, 881 F.3d 894, 906-907 (Fed. Cir. 2018) (“...effectively part of the host document as if it were explicitly contained therein”); *see also Paice LLC v. Ford Motor Co.*, 681 F. App’x 904, 917 (Fed. Cir. 2017) (“[c]ombining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness” (quoting *Boston Scientific Scimed, Inc. v. Cordis Corp.*, 554 F.3d 982, 991 (Fed. Cir. 2009))). Maes details a number of techniques for generating scores in connection with speech recognition. (Maes, 17:23-19:8; Schmandt, ¶97.) Maes further describes comparing speech recognition scores against a threshold in order to determine whether the scores are acceptable. (Maes, 16:20-17:7 (“[T]he remote server is contacted when these scores are determined to fall below a given threshold....[A]ny suitable measure

or approach to estimate the confidence or the recognized utterance or query (during or after the dialog) can be used to determine if the results obtained by a conversational system are acceptable or not”).) Similarly, in Coffman, with respect to matching context entries, a POSITA would have understood that determining whether a context was “acceptable” or “too improbable” would involve calculating some sort of probability-related score for the context entry. (Schmandt, ¶97.)

Coffman’s context entry scoring teaches “**generating one or more rank scores**” in at least three ways. (Schmandt, ¶98.) First, the scoring would be a “**rank score**” because it would serve to rank the context against some level of acceptability—similar to a pass/fail ranking system. By scoring “pass,” a context entry is selected by the system. (Coffman, 42:1-5.) Second, the scoring would be a “**rank score**” because it would serve to determine the order of activating matching context entries. Coffman makes clear that, even after identifying a first matching entry, its context-matching process continues to evaluate other context entries to identify and queue-up additional matching entries to potentially activate. (Schmandt, ¶98; 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.”), Coffman, 40:24-27 (“peruse down the context stack”).) Third, the scoring would be a “rank score,” because determining that the context

entry is an acceptable match results in reordering the context stack to move the matched context entry to the top of the stack. (Coffman, 42:4-7 (“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. Until the active command is completed, or until a new command is provided, the selected context is marked active, and pushed to the top of the context stack 405.”).) These ways in which Coffman discloses “**generating one or more rank scores**” mirror the ’652 written description. (Schmandt, ¶98; ’652, 34:13-24.) And notably, Kennewick includes the same disclosure verbatim as the ’652 patent (*see* Kennewick, ¶¶0152-0153), further confirming the knowledge of a POSITA and the obviousness of scoring and rank scores in Coffman’s system. (Schmandt, ¶98 n.5.)

Petitioner further notes that the concept of “rank” or “ranking” broadly encompasses the simple act of assigning something to a particular position or category, and does not require, for example, comparing two things to each other to determine which is first and which is second. (Schmandt, ¶99 (citing EX1005, p.003 (“**rank** ... v.t... 19. to assign to a particular position, station, class, etc.”))).) Real-world examples illustrate this understanding—such as rank assigned to enlisted military personnel, as well as belt ranking systems in Karate and other forms of martial arts. In both of these examples, the assignment of a rank is objective with

respect to particular requirements, and independent of any rankings assigned to other participants. (Schmandt, ¶99 (citing EX1010, pp.006-0015).)

Coffman therefore discloses and renders obvious **“generating one or more rank scores for individual context entries of the plurality of context entries based on the comparison.”**

- (c) **“identifying, from among the plurality of context entries, the one or more context entries based on the one or more rank scores.” (4[c])**

The analysis for 4[b] addressed the limitations of 4[c]. As discussed, Coffman describes identifying a context entry in a context stack to activate based on whether the context match is acceptable and not too improbable—i.e., **“based on the one or more rank scores.”** (Coffman, 40:24-27, 42:1-7; Schmandt, ¶101.)

- 5. **Claim 5: “The system of claim 4, wherein the plurality of context entries are ordered in the first context stack, and wherein the one or more physical processors are further caused to update the ordering of the plurality of context entries in the first context stack based on the one or more rank scores.”**

As discussed for 4[b], Coffman discloses that, based on matching a context entry, the matched entry is moved to the top of the context stack, thereby **“updat[ing] the ordering of the plurality of context entries”** in the context stack. (Coffman, 42:1-7; Schmandt, ¶102.)

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would

execute software (computer program instructions) to perform this step. (Schmandt, ¶103.)

6. Claim 6: “The system of claim 5, wherein the natural language utterance is associated with a user, and wherein the one or more physical processors are further caused to:”

For the reasons discussed for 2[a], Coffman discloses “**the natural language utterance is associated with a user**” in three ways—first, simply because by providing the natural language utterance, the user is associated with that utterance; second, by performing speaker recognition/identification on a user’s utterance; and third, by associating user identity with context to determine commands to perform. (Schmandt, ¶104 (citing Coffman, e.g., Fig. 4 (speaker recognition), 5:1-4, 29:1-4, 15:13-16:24, 47:25-27, 8:16-18, 48:1-3, 46:26-31, 48:11-14).)

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to perform the following steps. (Schmandt, ¶105.)

(a) “receive a subsequent natural language utterance associated with the user; and” (6[a])

6[a] simply corresponds to the master/central device in Coffman’s conversational system, as described for claim 1, receiving another utterance from the user, such as a new command or request. (Schmandt, ¶106.) For example, 6[a] would be satisfied by the system, after receiving the user’s utterance “I would like

to open this,” subsequently receiving another utterance from that user, such as “email from my mother” or “it should be synchronized with my PDA.” (*Id.*; Coffman, 6:32-7:3 (“I would like to open this”), 48:1-3 (“e-mail from my mother”), 56:25-57:2 (“grocery list,” “it should be synchronized with my PDA,” and “Do it!”); *see also id.*, 42:5-7 (“Until the active command is completed, or until a new command is provided, the selected context is marked active, and pushed to the top of the context stack 405.”).) Coffman expressly describes a scenario where a user’s PC in a distributed conversational system receives a series of natural language utterances from the user to perform various actions. (Coffman, 56:26-57:2; Schmandt, ¶106.)

- (b) “determine one or more subsequent words of the subsequent natural language utterance by performing speech recognition on the subsequent natural language utterance;” (6[b])**

6[b] simply corresponds to the master/central device in Coffman’s conversational system, as described for claim 1, performing speech recognition on the user’s subsequent natural language utterance. It is disclosed and obvious for the reasons explained for 1[a][iv]. (Schmandt, ¶107.)

- (c) “identify, from among the plurality of context entries, one or more other context entries that correspond to the one or more subsequent words based on the updated ordering; and” (6[c])**

6[c] simply corresponds to the master/central device in Coffman’s conversational system, as described for claim 1, identifying one or more context

entries in the context stack corresponding to the recognized words, as discussed for 3[a]. (Schmandt, ¶108; Coffman, e.g., 42:1-5.) And as discussed for claim 5, the context stack would have had an updated ordering as a result of processing the previous user utterance. (Coffman, 42:1-7.) As noted, Coffman explains that the process of identifying a matching context involves searching down the context stack. (Coffman, 40:24-27, 42:1-5.) Therefore, the claimed “**identify[ing]**” would have been “**based on the updated ordering.**” (Schmandt, ¶108.)

(d) “**determine a subsequent command or request based on the one or more subsequent words and the one or more other context entries.**” (6[d])

6[d] simply corresponds to the master/central device in Coffman’s conversational system, as described for claim 1, determining a command or request based on the recognized words in the subsequent utterance and the one or more identified context entries, as described for 6[b] and 6[c], respectively. It is disclosed and obvious for the reasons explained for 1[a][iv] and 3[a]. (Schmandt, ¶109; Coffman, e.g., 42:1-5, 41:2-5; *see also id.*, 30:8-10, 37:20-26, 40:12-15, 40:20-24.)

7. **Claim 7: “The system of claim 3, wherein identifying the one or more context entries comprises identifying, from among the plurality of context entries, the one or more context entries that most closely correspond to the one or more words.”**

It would have been obvious for the identified one or more context entries in Coffman to be “**the one or more context entries that most closely correspond to**

the one or more words.” This is because, as discussed, Coffman explains that the system can prompt the user to confirm that the identified context entry correctly corresponds to the user’s input. (Schmandt, ¶110; Coffman, 42:1-5, 41:30-32, 42:18-20.) By providing confirmation, the user indicates that the identified context most closely corresponds to the user’s utterance. (Schmandt, ¶110.)

8. Claim 8: “The system of claim 3, wherein the one or more physical processors are further caused to:”

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the device’s processor(s) would execute software (computer program instructions) to perform the following steps. (Schmandt, ¶111.)

(a) “identify one or more domain agents associated with the one or more context entries, wherein the one or more domain agents are configured to process the command or request; and” (8[a])

8[a] 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, ¶112.) Consistent with this general understanding, the ’652 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.” (’652, 27:14-17, 28:11-12; Schmandt, ¶112.)

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). (Schmandt, ¶113; 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.”).) 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. (Schmandt, ¶114; 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”).)

A conversationally-aware application, consistent with the ’652 patent’s description of “domain agent,” also usually provides functionality related to a specific domain, such as a particular subject matter (such as email or spreadsheet) or device (such as a PDA or smart phone). (Schmandt, ¶114; 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.”).)

A conversationally-aware application in Coffman is “**associated with the one or more context entries.**” (Schmandt, ¶115.) As discussed for claim 1[a][i], Coffman explains that context stack entries are created for and associated with applications. (Coffman, 30:4-10 (“[E]ach 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 id.*, 29:18-30:3.)

A conversationally-aware application in Coffman is “**configured to process the command or request.**” (Schmandt, ¶116.) As discussed for claim 1[a][v],

Coffman describes completing and activating a context entry's command/query for the associated application to process. (Coffman, 37:2-4; *see also id.*, 42:1-5, 41:2-5, 40:12-15.)

(b) “generate a response to the command or request using the one or more domain agents.” (8[b])

As discussed above, Coffman discloses processing a command or request using a conversationally-aware application (“**domain agent**”). (Coffman, e.g., 37:2-4.) Coffman further discloses that a conversationally-aware application returns output as a result of processing the command or request, which is used to generate a response to provide to the user. (Schmandt, ¶117.) For example, Coffman describes executing completed queries and “return[ing] output of executed or running processes” and presenting them to user using, for example, text-to-speech, as Coffman illustrates in Figure 6:

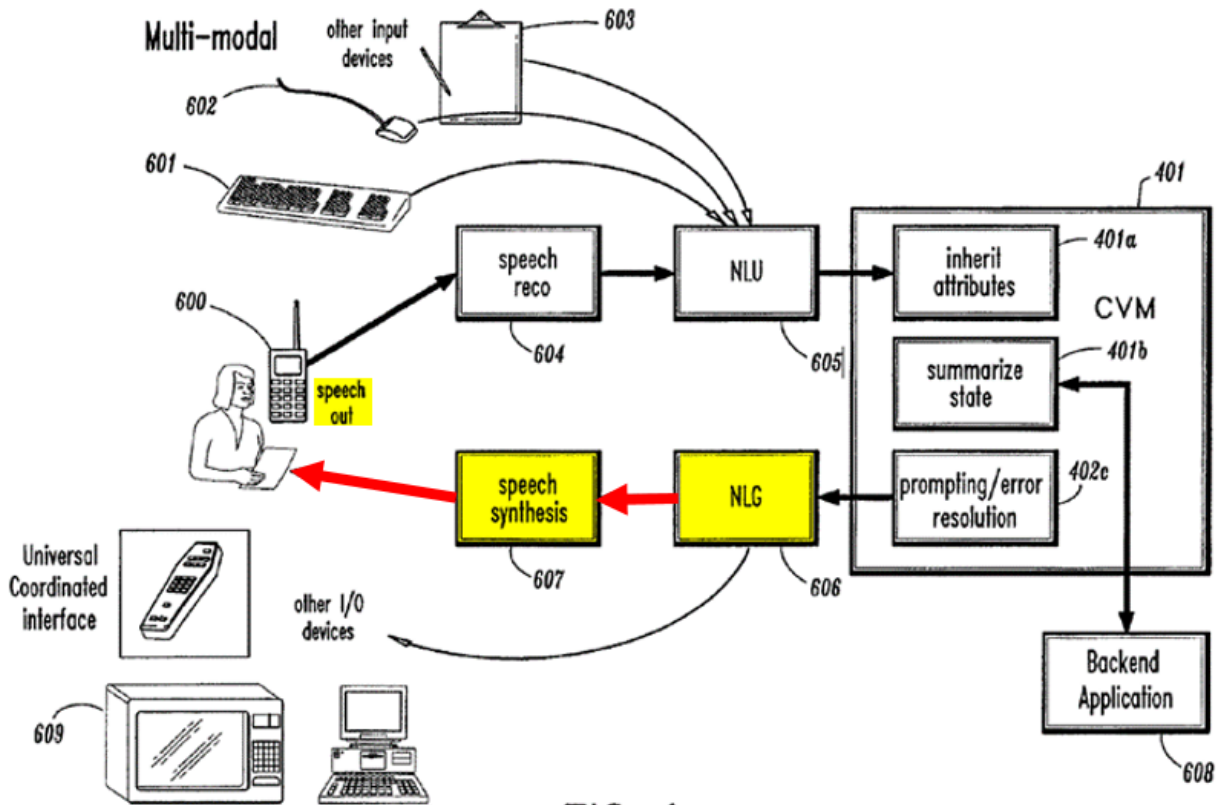


FIG. 6

(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”), 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.”).)

9. **Claim 9: “The system of claim 8, wherein each of the one or more domain agents comprises domain knowledge associated with a particular domain, the domain knowledge comprising: one or more of (i) a keyword; (ii) a link to an information source; (iii) a list of responses associated with a plurality commands or requests; (iv) a substitution list used to format the plurality of commands or requests; or (v) content including dictionaries, encyclopedias, or almanacs.”**

As explained for 8[a], 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., 57:23-27, 36:31-32.)

With respect to claim 9, a conversationally-aware application’s “**domain knowledge**” includes its data, such as its vocabulary, grammar, language model(s) and NLU symbolic mapping. (Schmandt, ¶119.) This data is “**associated with a particular domain**” and comprises at least “**a keyword**” because it is supplied by the application in connection with registration and includes words that Coffman’s system uses to recognize a user’s utterance and map it to a query associated with an application. (*Id.*) For example, using Coffman’s example, the word “open” (“**keyword**”) in the request “I would like to open this” matches an “open” command in an application’s vocabulary and associated symbolic language. (Coffman, 6:30-7:3.) More generally, Coffman describes “match[ing]” recognized words and phrases against an NLU symbolic language and “mapping” user input into a symbolic language. (Coffman, 42:1-7; *see also id.*, 53:22-26 (“User inputs can be

represented by their transcriptions or their mappings into a symbolic language after parsing and NLU.”).) Coffman’s example is an instance of the well-known concept of “keyword detection,” also known as “word spotting.” (Schmandt, ¶119 (citing EX1008, p.0029 (defining “keyword detection” as “[t]he detection of a given word or word sequence in a stream of speech or text” and identifying “word spotting” as a synonym))).)

10. Independent Claim 14

Claim 14 is not meaningfully different from claim 1 for purposes of this Petition, reciting the “computer-implemented method” counterpart to the “system” of claim 1. It therefore would have been obvious for the reasons explained for claim 1. (Schmandt, ¶120 (side-by-side claim comparison).)

11. Claims 15-18

Claims 15-18 are not meaningfully different from claims 2-4 and 8, respectively, for purposes of this Petition, reciting the “computer-implemented method” counterpart to the “system” of claims 2-4 and 8. Claims 15-18 therefore would have been obvious for the reasons explained for claims 2-4 and 8. (Schmandt, ¶¶121-124 (side-by-side claim comparison).)

12. Independent Claim 20

Claim 20 is not meaningfully different from claim 1 for purposes of this Petition, reciting the “computer-implemented method” counterpart to the “system”

of claim 1. While claim 20 recites a “**context set**” instead of a “**context stack**,” this is not a meaningful difference, as a POSITA would have understood that a “set” is generally more broadly understood as a collection of information and therefore that a “context stack” is a type of “context set.” (Schmandt, ¶125.) Claim 20 therefore would have been obvious for the reasons explained for claim 1. (*Id.*, ¶125 (side-by-side claim comparison).)

13. Claims 21-26

Claims 21-26 are not meaningfully different from claims 2-3 and 7-10, respectively, for purposes of this Petition, reciting the “computer-implemented method” counterpart to the “system” of claims 2-3 and 7-10. Claims 21-26 therefore would have been obvious for the reasons explained for claims 2-3 and 7-10. (Schmandt, ¶¶126-131 (side-by-side claim comparison).)

C. Ground 2: Claims 10-13, 19 and 27 Are Obvious Over Coffman and Kennewick.

1. Claim 10: “The system of claim 8, wherein the response comprises an aggregation of one or more responses generated by the one or more domain agents based on the command or request.”

The additional limitation of claim 10 would have been obvious over Coffman alone or in view of **Kennewick**. (Schmandt, ¶¶132-138.)

As discussed for claim 8, 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, ¶133.) 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.) 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.) 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] comprises an aggregation of one or more responses generated by the one or more domain agents based on the command or request.”** (Schmandt, ¶133.)

Claim 10 would have also been obvious in view of **Kennewick**. Kennewick shares a number of named inventors with the ’652 patent, and the ’652 patent appears to incorporate a substantial amount of Kennewick’s disclosures. (Schmandt, ¶48.) Like Coffman and the ’652 patent, Kennewick discloses a speech-based interface for interacting with and directing computer systems and software. (Kennewick, ¶0003.) With respect to claim 10, 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.*) 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 generating a response that “**comprises an aggregation of one or more responses generated by the one or more domain agents based on the command or request.**” (Schmandt, ¶134.)

Rationale and Motivation to Combine (Coffman and Kennewick). It would have been obvious to a POSITA to combine Coffman and Kennewick, predictably resulting in using a conversationally aware application (“**domain agent**”) in Coffman’s conversational system to generate a response “**compris[ing] an aggregation of one or more responses generated by**” a conversationally aware application (“**the one or more domain agents**”) “**based on the command or request.**” (Schmandt, ¶135.)

Both Coffman and Kennewick are analogous references in the same field as the ’652 patent and would have been reasonably pertinent to a number of problems

facing the patent inventors, including how to use application(s) to generate responses based on a user's command or request. (Schmandt, ¶136; '652, 1:22-25; Coffman, e.g., 1:7-8, 12:28-30, 57:23-27, Figs. 1, 4 & 6; Kennewick, ¶0003, 0088-0094.)

A POSITA would have been motivated to combine Coffman and Kennewick. As discussed above, Coffman and Kennewick disclose similar systems for controlling a number of applications via a speech-based interface. 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 information gathered when Coffman's conversationally-aware application processed the request. (Schmandt, ¶137.) 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. (*Id.*) As such, a POSITA would have recognized that aggregating information would allow the system to provide more complete and useful responses to the user. (*Id.*)

A POSITA would have had a reasonable expectation of success, as incorporating Kennewick's teachings into Coffman's conversational system would have involved the straightforward application of Kennewick's technique to improve

Coffman's similar agent-based conversational computing system in the same way without undue experimentation. (Schmandt, ¶138.)

2. Claim 11: “The system of claim 8, wherein the one or more physical processors are further caused to:”

As explained below, the additional steps recited in claim 11 would have been obvious over Coffman in view of Kennewick. For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the processor(s) of Coffman's master/central device would execute software (computer program instructions) to perform the following steps. (Schmandt, ¶139.)

(a) “obtain information relating to a license agreement that is associated with at least one of the one or more domain agents;” (11[a])

Coffman discloses that its conversational computing system includes help and support services providing functionality to install, upgrade and re-install conversationally-aware applications (“**domain agents**”):

In yet another aspect, conversational help, manuals and support is provided through a ubiquitous coordinated conversational interface, using local and remote resources, usage history of a user and agents to complete request, guide through procedure, search for information and upgrade/install new applications.

(Coffman, 9:4-7; *see also id.*, 52:7-11 (“Different degrees and modalities of help are provided....It can take the form of conversational technical support involving local or remote agents (e.g. to upgrade or re-install an application in the background).”);

21:15-18 (“It is to be appreciated that services such as conversational help and assistance, as well as some dialog prompts (introduction, questions, feedback etc) provided by the CVM system...”).) Coffman also describes providing the ability for parties to develop conversationally aware applications for the CVM system, explaining that “development tools are provided for developer to build, simulate and debug conversational aware application for CVM.” (Coffman, 7:29-30; *see also id.*, 18:16-18, 13:21-23 (“[T]he conversational programming languages/scripts provide the conversational APIs that allow an application developer to hook (or develop) conversationally aware applications 200”).)

Software licenses were commonly known at the time and even casual computer users would recall being prompted to click “I agree” to an end-user license agreement presented during software installation. (Schmandt, ¶141.) Coffman does not detail licensing-related issues associated with installing and using conversationally-aware applications, but the claimed step of “**obtain[ing] information relating to a license agreement that is associated with**” a conversationally-aware application (“**at least one of the one or more domain agents**”) would have been obvious in view of Kennewick. (Schmandt, ¶141.) With respect to claim 11, Kennewick describes an “agent manager” that verifies that software use is within allowed license terms and conditions, and if so, allows the software to be used:

The agent manager **154** may provide a license management client capable of executing most any license terms and conditions. When a particular agent **106** and/or database element **102** is required by a command, the agent manger [sic] **154** verifies that the use of the agent or data element is within the allowed terms and conditions, and if so, invokes the agent or allows access to the data element. License management schemes that can be implemented through the agent manager **154** include outright purchase, subscription for updates, one time or limited time use. Use of shared agents and data elements (such as those down-loaded from web sites maintained by groups of domain experts) may also be managed by the agent manager **154**.

(Kennewick, ¶0104.) These disclosures are nearly identical those in the '652 patent.

(*Compare* Kennewick, ¶0104 with '652, 28:63-29:9.)

The rationale and motivation to combine Coffman and Kennewick discussed in connection with claim 10 applies here as well. (Schmandt, ¶142.) With respect to claim 11, combining Coffman and Kennewick would have predictably resulted in Coffman's conversational system having functionality to verify that use of conversationally-aware applications was within allowed license terms and conditions. (*Id.*) Combining the teachings of Coffman and Kennewick in this manner would have therefore resulted in **"obtain[ing] information relating to a license agreement that is associated with"** a conversationally-aware application (**"at least one of the one or more domain agents"**), as claimed. (*Id.*) A POSITA

would have been motivated to combine. (*Id.*) As discussed above, Coffman and Kennewick disclose similar systems for controlling a number of applications via a speech-based interface. As is commonly known, software application developers often make their software available pursuant to license terms and conditions, and expect that a user of their software will not violate those terms and conditions. (*Id.*) Application developers have various motivations for terms and conditions, including revenue generation. (*Id.*) If the terms and conditions cannot be adequately enforced, the application developer may opt to not develop the application, or to not develop the application for a particular platform. (*Id.*) A POSITA would have readily appreciated that this would have the effect of both making the application unavailable, as well as weakening the overall value of the platform by reducing the size of its application ecosystem. (*Id.*) A POSITA would have therefore been motivated by these market-forces-related issues to include license management functionality as taught by Kennewick as part of Coffman's conversational system and thereby increase the value and usability of the system to its users. (*Id.*)

A POSITA would have had a reasonable expectation of success, as the combination mostly relates to non-technical licensing issues and would have involved the straightforward application of conventional techniques and technologies without undue experimentation. (Schmandt, ¶143.)

- (b) **“determine, based on the information related to the license agreement, that use of the at least one of the one or more domain agents to process the command or request is permitted; and” (11[b])**

11[b] was addressed in the analysis of 11[a]. As discussed, Kennewick discloses “agent manager” functionality that verifies that software use is within allowed license terms and conditions, and if so, allows the software to be used. (Kennewick, ¶0104.) And as discussed, it would have been obvious to combine Kennewick’s teachings regarding license management with Coffman. The combination would have predictably resulted in Coffman’s conversational system having functionality to **“determine, based on the information related to the license agreement, that use of”** the conversationally-aware application (**“the at least one of the one or more domain agents”**) **“to process the command or request is permitted.”** (Schmandt, ¶144.)

- (c) **“use, based on the determination that the use is permitted, the at least one of the one or more domain agents to generate the response.” (11[c])**

11[c] was also addressed in the analysis of 11[a]. The combination of Coffman and Kennewick in the manner described would have predictably resulted in Coffman’s conversational system having functionality to **“use, based on the determination that the use is permitted,”** the conversationally-aware application (**“the at least one of the one or more domain agents”**) **“to generate the response.”** (Schmandt, ¶145.)

3. Claim 12: “The system of claim 1, wherein the natural language utterance is associated with a user, and wherein the one or more physical processors are further caused to:” (12[pre])

As discussed for 2[a], Coffman discloses that the natural language utterance is “**associated with a user**” because the system receives spoken natural language input from a person in connection with using the system; and moreover, Coffman discloses identifying the user who spoke the utterance, thereby associating the utterance with a user. (Schmandt, ¶146 (citing Coffman, e.g., 5:1-4, 29:1-4, 47:25-27, 8:16-18).)

For substantially the same reasons as those explained for 1[a], a POSITA would have understood and found it obvious that the system’s processor(s) would execute software (computer program instructions) to perform the following steps. (Schmandt, ¶147.)

- (a) “obtain one or more of a cognitive model or an environmental model associated with the user, wherein the cognitive model comprises information relating to one or more interactions between the user and the system, and the environmental model comprises information indicative of how noisy an environment surrounding the user is,” (12[a])**

With respect to a “**cognitive model**,” the ’652 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.” (’652, 24:1-11; Schmandt, ¶148.)

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 relating to one or more interactions between the user and the system,**” 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 interactions with the device that the system may use in connection with recognizing words in the user’s natural language utterance. (Schmandt, ¶149.) The cognitive model is “**obtained**” by collecting information about the user, as well as when the collected information is accessed in connection with performing recognition. (*Id.*)

Kennewick further confirms and renders obvious the claimed “**cognitive model.**” (Schmandt, ¶150.) Like Coffman, Kennewick similarly discloses that its “system may create, store and use extensive personal profile information for each user.” (Kennewick, ¶0029.) The personal profile information includes

“information relating to one or more interactions between the user and the system,” 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.”).)

Kennewick further discloses that the collected user profile information may be used in connection with speech recognition, 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.) 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; *see also id.* (“The word is looked up in the dictionary **112** and the pronunciation for the word is added to either the dictionary, the agent **106**, or the user’s profile **110**. The word pronunciation can then be associated with the domain, the question, the context and the user. Though this process the speech recognition engine learns with time and improves accuracy.”).)

The rationale and motivation to combine Coffman and Kennewick discussed in connection with claim 10 applies here as well. (Schmandt, ¶152.) In addition, with respect to claim 12, a POSITA would have been motivated 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. (*Id.*) 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 “**obtain[ing] one or more of a cognitive model** [or an environmental model associated with the user], **wherein the cognitive model comprises information relating to one or more interactions between the user and the system** [, and the environmental model comprises information indicative of how noisy an environment surrounding the user is]” as claimed. (*Id.*)

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. (Schmandt, ¶153 (citing 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.”)).) A POSITA

therefore would have been motivated to incorporate Kennewick's techniques into Coffman's similar system to obtain the same benefits and advantages. (*Id.*)

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, ¶154 (citing Kennewick, ¶0006; EX1008, p.0024 ("adaptive language model" entry)).)

(b) "wherein the one or more words are determined further based on one or more of the cognitive model or the environmental model." (12[b])

As discussed for 12[a], both Coffman and Kennewick disclose that collected user profile information ("**cognitive model**") is used in performing speech recognition, i.e., determining one or more words of the natural language utterance. (*E.g.*, Coffman, 37:5-10; Kennewick, ¶¶0149, 0151.) 12[b] is therefore obvious over Coffman in view of Kennewick. (Schmandt, ¶155.)

4. Claim 13: "The system of claim 1, wherein the natural language utterance is associated with a user, and wherein the one or more physical processors are further caused to:" (13[pre])

As discussed for 2[a] (and 12[pre]), Coffman discloses that the natural language utterance is "**associated with a user**" because the system receives spoken natural language input from a person in connection with using the system; and moreover, Coffman discloses identifying the user who spoke the utterance, thereby

associating the utterance with a user. (Schmandt, ¶156 (citing Coffman, e.g., 5:1-4, 29:1-4, 47:25-27, 8:16-18).)

- (a) **“obtain a first cognitive model that comprises information relating to one or more interactions between the user and the system; and” (13[a])**

For the same reasons explained for 12[a], Coffman and Kennewick disclose and render obvious the claimed **“first cognitive model.”** (Schmandt, ¶157.)

- (b) **“obtain a second cognitive model that comprises information relating to one or more interactions between the system and a plurality of users of the system,” (13[b])**

As discussed above for 12[a], the written description of the '652 patent describes a “personalized cognitive model **810**.” ('652, 24:1-11.) In addition to that model, the '652 patent also describes a second, “general cognitive model **806**” that “corresponds to interaction patterns with the system for multiple users.” ('652, 24:12-21.)

Kennewick discloses the claimed **“second cognitive model”** in the form of profile information about other users. (Schmandt, ¶159.) 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.” (*Id.* (quoting Kennewick, ¶0149).) As discussed above for claim 12[a], 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 system and a plurality of users of the system”** because Kennewick explains that its system supports multiple users in a given session—e.g., multiple users of the same device. (Schmandt, ¶159 (citing 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. (Schmandt, ¶159 (citing Coffman, 47:25-26, 47:27-48:7, 29:22-23).) Like the first cognitive model, the second cognitive model is **“obtained”** by collecting information about the users, as well as when the collected information is accessed in connection with performing recognition. (*Id.*)

The rationale and motivation to combine Coffman and Kennewick discussed for 12[a] and applies here as well. (Schmandt, ¶160.) With respect to claim 13[b], 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.*)

(c) **“wherein the one or more words are determined further based on the first cognitive model and the second cognitive model.” (13[c])**

As discussed for 12[a] (for the **“first cognitive model”**) and 13[b] (for the **“second cognitive model”**), Kennewick discloses that collected user profile information (**“first cognitive model”** and **“second cognitive model”**)) are used in performing speech recognition, i.e., determining one or more words of the natural language utterance. (Kennewick, *e.g.*, ¶¶0149, 0151.) 13[c] is therefore obvious over Coffman in view of Kennewick. (Schmandt, ¶162.)

5. Claim 19

Claim 19 is not meaningfully different from claim 13 for purposes of this Petition, reciting the “method” counterpart to the “system” of claim 13. Claim 19 therefore would have been obvious for the reasons explained for claim 13. (Schmandt, ¶163 (side-by-side claim comparison).)

6. Claim 27

Claim 27 is not meaningfully different from claim 11 for purposes of this Petition, reciting the “computer-implemented method” counterpart to the “system” of claim 11. Claim 27 therefore would have been obvious for the reasons explained for claim 11. (Schmandt, ¶164 (side-by-side claim comparison).)

D. Ground 3: Claims 13 and 19 Are Obvious Over Coffman, Kennewick and Lee.

As discussed above in the Ground 2 analysis, claims 13 and 19 recite using a “**second cognitive model**” in performing speech recognition, i.e., determining one or more words of the user’s natural language utterance. As explained, Kennewick discloses the claimed “**second 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 further explained, using other users’ profile information when performing speech recognition could help the system recognize likely input from the user. (Kennewick, ¶0149.)

Lee 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, ¶166.) 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 Coffman and Kennewick). It would have been obvious to a POSITA 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 and Kennewick described in connection with Ground 2. (Schmandt, ¶167.) 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 determining one or more words of the user's natural language utterance using a "**second cognitive model**" as claimed. (Schmandt, ¶167.)

Lee is an analogous reference in the same field as the '652 patent and would have been reasonably pertinent to problems facing the '652 patent inventors, including techniques for improving speech recognition accuracy, including in multi-user systems. (Schmandt, ¶168; '652, 1:22-27; Lee, Abstract, ¶0002.)

A POSITA would have been motivated to combine. (Schmandt, ¶169.) 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

Petition for *Inter Partes* Review of
U.S. Patent No. 8,849,652 B2

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. (Schmandt, ¶169 (citing Lee, ¶¶0016, 0024, 0026).) A POSITA therefore would have been motivated to incorporate Lee’s teachings into Coffman’s conversational system to obtain the same benefits and advantages. (*Id.*)

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, ¶170 (citing Kennewick, ¶0006; EX1008, p.0024 (“adaptive language model” entry)).)

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 13,839 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. 8,849,652 B2**, including all exhibits (**Nos. 1001-1014**) 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

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