

Petition for *Inter Partes* Review of
U.S. Patent No. 7,398,209 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

DIALECT, LLC,
Patent Owner

Case IPR2024-00746
U.S. Patent No. 7,398,209 B2
Issue Date: July 8, 2008

Title: SYSTEMS AND METHODS FOR RESPONDING TO
NATURAL LANGUAGE SPEECH UTTERANCE

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 7,398,209 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.....	15
A. Person of Ordinary Skill in the Art (POSITA)	15
B. Specification Overview	15
V. CLAIM CONSTRUCTION	17
VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	17
A. Overview of Grounds	17
B. Ground 1: Claims 1-2, 4-5 Are Obvious Over Coffman in View of Kanevsky, Kneser, and Zadrozny.	18
1. Claim 1: “A method...”	18
(a) “receiving...” (1[a]).....	19
(b) “maintaining...” (1[b])	21
(c) “recognizing...” (1[c]).....	29
(d) “parsing...” (1[d]).....	31
(e) “selecting...” (1[e])	33

Table of Contents (continued)

	Page
(f) “formulating...” (1[f])	37
(g) “invoking...” (1[g])	41
(h) “presenting....” (1[h]).....	42
2. Claim 2.....	43
(a) “dynamically updating...” (2[a]).....	43
(b) “determining...” (2[b]).....	43
(c) “associating...” (2[c]).....	45
3. Claim 4.....	49
(a) “matching...” (4[a]).....	49
(b) “scoring...” (4[b]).....	51
(c) “determining....” (4[c])	53
4. Claim 5.....	54
(a) “requesting...” (5[a]).....	54
(b) “determining...” (5[b])	54
C. Ground 2: Claim 2 is Obvious Over Ground 1 Prior Art in View of Maes.	55
1. Claim 2.....	55
D. Ground 3: Claim 3 is Obvious Over Ground 2 Prior Art in Further View of Maes ’101 and Franceschi.....	58
1. Claim 3.....	58
(a) “requesting...” (3[a]).....	58
(b) “updating...” (3[b])	61
E. Ground 4: Claim 6 is Obvious Over Ground 1 Prior Art in Further View of Chai.....	62
1. Claim 6.....	62
(a) “determining...” (6[a])	62
(b) “extracting...” (6[b])	68

Table of Contents
(continued)

	Page
(c) “inferring...” (6[c]).....	69
(d) “transforming....” (6[d]).....	70
F. Ground 5: Claims 7-9 and 12 are Obvious Over Ground 1 Prior Art in Further View of Redfern.....	71
1. Claim 7	71
(a) “directing...” (7[a])	76
(b) “querying...” (7[b])	77
2. Claim 8	77
(a) “submitting...” (8[a])	77
(b) “a synchronously receiving...” (8[b])	79
(c) “scoring....” (8[c]).....	80
3. Claim 9	80
(a) “determining...” (9[a])	81
(b) “extracting...” (9[b])	82
(c) “evaluating...” (9[c]).....	84
(d) “selecting....” (9[d])	85
4. Claim 12	86
(a) “determining...” (12[a])	86
(b) “resolving....” (12[b])	86
VII. CONCLUSION.....	87
CERTIFICATE OF SERVICE	89

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 7,398,209 B2 to Robert Kennewick et al. (filed June 3, 2003, issued July 8, 2008) (“ 209 ” or “ 209 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 No. 5,897,616 to Dimitri Kanevsky and Stephane Herman Maes (filed June 11, 1997, issued April 27, 1999) (“ Kanevsky ”)
1005	U.S. Patent No. 6,157,912 to Reinhard Kneser et al. (filed March 2, 1998, issued December 5, 2000) (“ Kneser ”)
1006	Wlodek Zadrozny et al., “Natural Language Understanding with a Grammar of Constructions”, Proc. 15th Int’l Conf. on Computational Linguistics (Vol. 2) (1994) (“ Zadrozny ”)
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	U.S. Patent No. 6,073,101 to Stephane Herman Maes (filed January 28, 1997, issued June 6, 2000) (“ Maes ’101 ”)
1011	U.S. Patent No. 6,321,196 B1 to Carlos Antonio Franceschi (filed July 2, 1999, issued November 20, 2001) (“ Franceschi ”)

List of Exhibits

Exhibit No.	Description of Document
1012	U.S. Patent No. 6,829,603 B1 to Joyce Yue Chai, Wlodek Zadrozny et al. (filed February 2, 2000, issued December 7, 2004) (“ Chai ”)
1013	U.S. Patent No. 6,078,914 to Darren M. Redfern (filed December 9, 1996, issued June 20, 2000) (“ Redfern ”)
1014	U.S. Patent Application Publication No. 2003/0144846 to Lawrence A. Denenberg and Christopher M. Schmandt (filed January 31, 2002, published July 31, 2003) (“ Denenberg ”)
1015	Excerpts from <i>Merriam-Webster’s Collegiate Dictionary</i> (10th ed. 1994)
1016	Excerpts from Richard F. E. Sutcliffe et al., <i>Industrial Parsing of Software Manuals</i> (1996) (“ Sutcliffe ”)
1017	Christopher Schmandt, “The Intelligent Ear: A Graphical Interface to Digital Audio” in Proc. Int’l Conf. Cybernetics and Society, pp.393-397 (1981)
1018	Christopher Schmandt et al., “The Intelligent Voice-Interactive Interface,” Proc. Human Factors in Comp. Sys., pp.363-366 (1982)
1019	Christopher Schmandt et al., “A Robust Parser and Dialog Generator for a Conversational Office System” in Proc. AVIOS Conf. (1986)
1020	Declaration of Ingrid Hsieh-Yee, Ph.D.
1021	Prosecution history of U.S. Patent No. 7,398,209 B2
1022	Proof of Service of Complaint, <i>Dialect, LLC v. Google LLC</i> , Case No. 1:23-cv-00378 (D. Del.)
1023	International Preliminary Examination Report for Maes (EX1007) application

Claim Listing

	Claim 1
1[pre]	1. A method responsive to a user generated natural language speech utterance, comprising:
1[a]	receiving the user generated natural language speech utterance, the received user utterance containing at least one request;
1[b]	maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance;
1[c]	recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;
1[d]	parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases;
1[e]	selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context;
1[f]	formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;
1[g]	invoking the selected domain agent to process the formulated request; and
1[h]	presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.
	Claim 2
2[pre]	2. The method according to claim 1, wherein recognizing the words and phrases contained in the received utterance includes:
2[a]	dynamically updating the information in the dictionary and phrase tables based on the dynamic set of prior probabilities or fuzzy possibilities;
2[b]	determining an identity of the user based on voice characteristics of the utterance; and

Claim Listing

2[c]	associating the recognized words and phrases and a pronunciation of the recognized words and phrases with the identity of the user, the determined context, and the at least one request when the recognized words and phrases satisfy a predetermined confidence level.
	Claim 3
3[pre]	3. The method according to claim 2, wherein recognizing the words and phrases contained in the received utterance further includes:
3[a]	requesting the user to spell one or more of the words and phrases when the recognized words and phrases fail to satisfy the predetermined confidence level; and
3[b]	updating the information in the dictionary and phrase tables based on a phonetic alphabet spelling of the requested words and phrases, the phonetic alphabet spelling provided by the user in one or more subsequent natural language speech utterances.
	Claim 4
4[pre]	4. The method according to claim 1, wherein determining the context for the at least one request contained in the utterance includes:
4[a]	matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords, the dynamically updated set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities;
4[b]	scoring one or more possible contexts for the matched keywords using one or more of a profile of the user, data content associated with the domain agents, or a context stack of one or more recent contexts; and
4[c]	determining the context by selecting one or more of the possible contexts having a score satisfying a predetermined confidence level.
	Claim 5
5[pre]	5. The method according to claim 4, wherein determining the context for the at least one request contained in the utterance further includes:
5[a]	requesting the user to verify the recognized words and phrases when none of the possible contexts have a score satisfying the predetermined confidence level; and

Claim Listing

5[b]	determining the context based on information provided by the user in one or more subsequent natural language speech utterances.
Claim 6	
6[pre]	6. The method according to claim 1, wherein formulating the at least one request in accordance with the grammar used by the selected domain agent includes:
6[a]	determining one or more required values and one or more optional values for a request in the grammar used by the selected domain agent;
6[b]	extracting one or more criteria and one or more parameters from the keywords contained in the recognized words and phrases using procedures sensitive to the determined context;
6[c]	inferring one or more criteria and one or more parameters using the dynamic set of prior probabilities or fuzzy possibilities; and
6[d]	transforming the extracted and the inferred criteria and parameters into a format compatible with the grammar used by the selected domain agent, the transformed criteria and parameters including all of the required values and one or more of the optional values.
Claim 7	
7[pre]	7. The method according to claim 1, wherein the at least one request contained in the utterance includes at least one question and at least one command, wherein the invoked domain agent processes the formulated request by:
7[a]	directing the at least one command to local or remote processing devices; and
7[b]	querying local or network information sources to resolve information requested in the at least one question.
Claim 8	
8[pre]	8. The method according to claim 7, wherein querying the local or network information sources to resolve the information requested in the at least one question includes:

Claim Listing

8[a]	submitting a plurality of asynchronous queries to the information sources, the plurality of queries including a plurality of duplicate queries submitted to distinct information sources;
8[b]	a synchronously receiving a plurality of results of the queries from the information sources; and
8[c]	scoring a relevance of each of the asynchronously received results using the dynamic set of prior probabilities or fuzzy possibilities to determine one or more best responses to the at least one question.
	Claim 9
9[pre]	9. The method according to claim 8, wherein determining the one or more best responses to the at least one question includes:
9[a]	determining one or more tokens required to formulate a response to the question;
9[b]	extracting one or more values for the determined tokens from the asynchronously received results, the one or more values extracted by applying one or more scraping criteria to the asynchronously received results;
9[c]	evaluating the extracted values using the dynamic set of prior probabilities or fuzzy possibilities, wherein evaluating the extracted values includes resolving ambiguous, incomplete, or conflicting information; and
9[d]	selecting one or more best values for the determined tokens from the evaluated values, wherein the one or more best responses include the one or more selected best values.
	Claim 12
12[pre]	12. The method according to claim 9, wherein determining the one or more best responses to the at least one question further includes:
12[a]	determining that the dynamic set of prior probabilities or fuzzy possibilities has failed to resolve the ambiguous, incomplete, or conflicting information; and
12[b]	resolving the ambiguous, incomplete, or conflicting information by requesting the user to provide additional information for resolving the ambiguous, incomplete, or conflicting information.

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 ’209 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 ’209 patent on April 6, 2023. (EX1022, 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 ’209 patent was also the subject of *Dialect, LLC v. Samsung Electronics Co.*, Case No. 2:23-cv-00061 (E.D. Tex.) (filed Feb. 17, 2023), which was terminated in September 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. 7,398,209 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 '209 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 10 claims, with a \$41,500 payment.

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

A. Grounds for Standing

Petitioner certifies that the '209 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 1-9 and 12:

Ground	Claims	Basis for Challenge under §103
1	1, 2, 4, 5	Coffman (EX1003), Kanevsky (EX1004), Kneser (EX1005), Zadrozny (EX1006)
2	2	Ground 1 Prior Art, Maes (EX1007)
3	3	Ground 2 Prior Art, Maes '101 (EX1010), Franceschi (EX1011)
4	6	Ground 1 Prior Art, Chai (EX1012)
5	7-9, 12	Ground 1 Prior Art, Redfern (EX1013)

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

§314(a): Petitioner is unaware of another post-grant challenge against the '209 patent. The pending litigation involving Petitioner is still in early stages, with no discovery or claim construction having occurred and no trial date set, and as noted, was recently transferred to the Northern District of California. If IPR is instituted, Petitioner intends to bring a motion to stay the litigation.

§325(d): During '209 patent prosecution, an applicant-submitted supplemental IDS identified Coffman's U.S. counterpart (i.e., 7,137,126), but it was not substantively discussed. (EX1021, pp.00254-00256.) Why the Examiner did not appreciate the Coffman counterpart's significance is unclear, but the applicant had already disclosed over one hundred other references and the Examiner had already issued an office action rejecting pending claims over different prior art before it was disclosed. (EX1021, pp.0088-0094, 00183-00196, 00206-00209.) The Examiner also did not have the other references that this Petition relies on, which

provide further context and detail for understanding Coffman and how its disclosures provide compelling grounds of obviousness.

IV. OVERVIEW OF THE CHALLENGED PATENT

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

A POSITA as of June 3, 2002² 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 '209 patent states that it “relates to the retrieval of online information or processing of commands through a speech interface,” as illustrated:

² Petitioner does not concede a June 3, 2002 priority date, but that issue is irrelevant here because the prior art here predates that date.

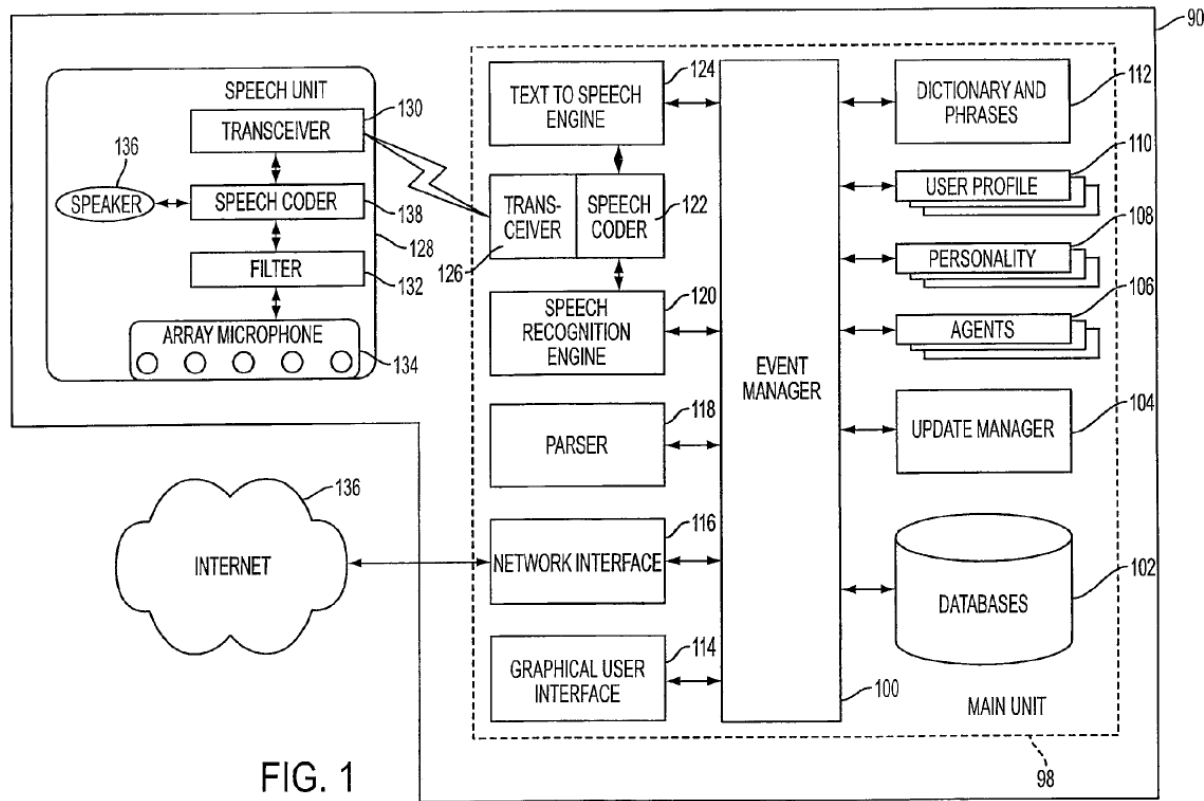


FIG. 1

(’209, Fig. 1, 1:13-15.) As shown, system **90** includes speech unit **128**, speech recognition engine **120**, dictionary and phrases **112**, parser **118**, and agents **106**. Speech unit **128** includes a microphone to receive a spoken utterance. (’209, 10:23-25.) Speech recognition engine **120** processes the speech “using data in the dictionary and phrases module **112** and received from the agents **106**.” (’209, 10:49-53.) Then, “[t]he recognized words and phrases are processed by the parser **118**, which transforms them into complete commands and questions using data supplied by an agent **106**.” (’209, 10:54-56.) “The agent **106** then process the commands or questions.” (’209, 10:57-58.) To select the correct agent to invoke to process the command or question, “parser **118** determines a context for an utterance by applying

prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history.” (’209, 17:45-51, 18:8-10.)

V. CLAIM CONSTRUCTION

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

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Overview of Grounds

The challenged claims attempt to lay claim to well-known techniques for speech-enabled interfaces for interacting with and directing computer systems. The claims employ standard techniques known long before the ’209 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 discloses a remarkably similar conversational computing system. All grounds combine Coffman with **Kanevsky** (EX1004), **Kneser** (EX1005) and **Zadrozny** (EX1006), which disclose known techniques related to speech recognition and natural language processing. Grounds 2-5 combine these references with **Maes** (EX1007), **Maes ’101** (EX1010), **Franceschi** (EX1011), **Chai** (EX1012) and/or **Redfern** (EX1013) for features recited in challenged dependent claims. As detailed below, these

references share links to each other and thus comprise a collective body of work describing aspects of the same systems and methods—for example, Coffman expressly incorporates Maes and Kanevsky by reference (Coffman, 22:25-29, 37:10-13), and Kanevsky in turn discusses Zadrozny (Kanevsky, 14:22-25); Coffman, Maes, Kanevsky, and Maes ’101 share a common co-inventor (Stephane Maes), as do Zadrozny and Chai (Wlodek Zadrozny); and the inventors and authors of Coffman, Maes, Kanevsky, Zadrozny, Maes ’101, Franceschi and Chai are identified as associated with IBM. *See Paice, LLC v. Ford Motor Co.*, 881 F.3d 894, 906-907 (Fed. Cir. 2018) (incorporated material “effectively part of the host document as if it were explicitly contained therein”); *Boston Scientific Scimed, Inc. v. Cordis Corp.*, 554 F.3d 982, 991 (Fed. Cir. 2009) (“[c]ombining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness”).

The references are prior art under at least pre-AIA §§102(b) (Coffman, Kanevsky, Kneser, Zadrozny, Maes, Maes ’101, Redfern), 102(a) (Franceschi) and 102(e) (Chai). (*See* EX1020, ¶¶25-48 (Zadrozny).)

B. Ground 1: Claims 1-2, 4-5 Are Obvious Over Coffman in View of Kanevsky, Kneser, and Zadrozny.

1. Claim 1: “A method...”

As shown below, the prior art discloses and renders obvious the claimed method.

(a) “receiving...” (1[a])

Coffman discloses “receiving a user generated...speech utterance”:

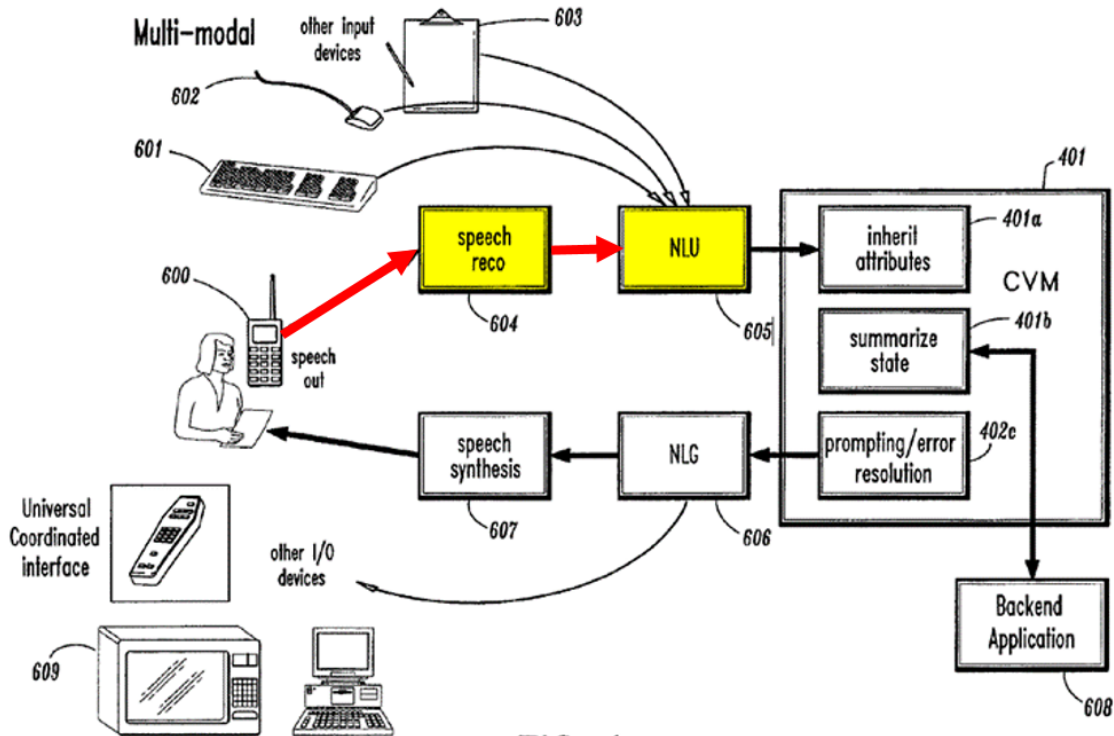


FIG. 6

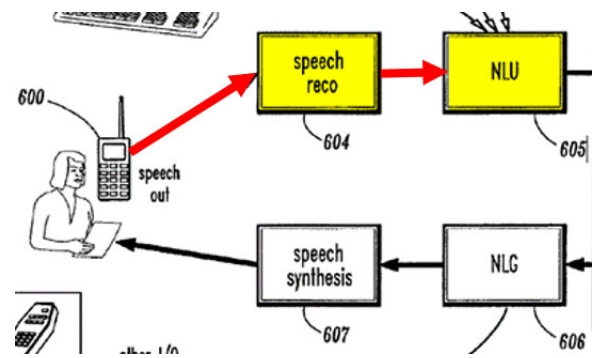
(Coffman, Fig. 6 (annotations added); *see also id.*, 53:8-10 (“Voice input from a microphone (e.g. the standard voice input)...”), 37:14-22 (“Spoken input from a speech client (e.g., telephone 600)...”), 41:2-5 (“...speech inputs being transcribed/understood...”), 53:22-26 (“speech”).)³

Coffman’s spoken input is a “**natural language** speech utterance.” In the art, “**natural language**” generally refers to the forms of language used by a person for communication (e.g., English). (Schmandt, ¶88 (citing Gibbon (EX1008),

³ All underlining added to quoted material unless noted.

p.0035 (“natural language”)).) Coffman discloses several examples of natural language speech utterances. (Coffman, 6:32-7:3 (“the command ‘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!”).)

Coffman’s receipt of spoken “**natural language**” input is confirmed by Coffman’s explanation that spoken input, once recognized, is subjected to natural language parsing and understanding (“NLU”) processes. (Coffman, Fig. 6 (annotated excerpt at right), 53:22-26 (“parsing and NLU”), 53:8-10 (“symbolic language representation provided by NLU”), 37:14-22 (“Spoken input from a speech client (e.g., telephone 600) is subject to a speech recognition process 604 and other input (e.g., keyboard, mouse clicks etc) are subject to NLU processing 605.”), 6:6-8 (“natural dialog with NLU”), 37:29-32, 5:1-4; Schmandt, ¶89.)



Coffman further discloses and renders obvious that the speech utterance “**contain[s] at least one request**,” explaining that the system “convert[s] voice requests into queries.” (Coffman, 13:8-13.) The examples from Coffman noted above—e.g., “I would like to open this”—are examples of spoken utterances that “**contain at least one request**.” (Schmandt, ¶90.)

(b) “maintaining...” (1[b])

By way of background, a “language model” generally refers to a formal representation of the structure of a natural language. (Schmandt, ¶92; EX1008, p.0031 (“language model”).) “In most speech recognisers language models consist of an n-gram model, i.e. probabilities of the occurrence of words, words pairs, sequences of three words, etc.” (Schmandt, ¶92 (quoting EX1008, p.0031, citing EX1008, pp.0035 (“n-gram language model”), 0050 (“stochastic language model”))).) A “trigram” language model, for example, therefore refers to an n-gram language model that uses the probabilities of the occurrence of sequences of three words (i.e., the “n” in “n-gram” is three). (Schmandt, ¶99; EX1008, p.0053 (defining “trigram” as “a sequence of three words”).)

Turning to “**maintaining a...set of prior probabilities or fuzzy probabilities,**” Coffman’s system includes speech recognition and natural language understanding, although Coffman does not offer much specific detail about these well-known concepts. (Schmandt, ¶93; Coffman, 19:16-18, 29:1-4.) *See, e.g., Bayer Healthcare LLC v. Baxalta Inc.*, 989 F. 3d 964, 982 (Fed. Cir. 2021) (“[a] patent need not teach, and preferably omits, what is well known in the art”). Coffman does explain, though, that a conversationally-aware application, when registering with the conversational system, provides “language models to control the application.” (Coffman, 57:22-58:11.) Additionally, Coffman incorporates

Kanevsky (EX1004) (Coffman, 37:10-13), which further explains that a speech recognition engine may be implemented using a “trigram language model.” (Kanevsky, 13:50-56; *see also id.*, 13:48-61 (describing such a speech recognition engine as an “existing component”).)

Coffman and Kanevsky disclose and render obvious “**maintaining a...set of prior probabilities.**” The “**set of prior probabilities**” corresponds to the set of probabilities associated with the trigrams in the language model used by the speech recognition engine and maintained by the conversational system as a result of application registration. A POSITA would have understood and found it obvious that the probabilities associated with the language model trigrams are “**prior probabilities**” because they each correspond to the probability of observing some phenomenon estimated from previously collected training data, independent of future observations. (Schmandt, ¶95 (citing EX1008, p.0043 (“prior probability”) and Sutcliffe (EX1016), p.006 (a language model is “extracted or learnt from a corpus, a representative set of texts in the domain of the speech recognizer.”))).)

Rationale and Motivation to Combine (Coffman and Kanevsky). It would have been obvious to combine Coffman and Kanevsky, predictably resulting in Coffman’s conversational system where language models provided by conversationally-aware applications and used by the system for speech recognition were n-gram (e.g., trigram) language models. (Schmandt, ¶96.) Under the

combination, the n-gram language models would comprise a set of “**prior probabilities**” that would have been maintained by the conversational system.

Coffman and Kanevsky are analogous references in the field of the '209 patent, and would have been reasonably pertinent to problems facing the patent inventors, including how to implement a system for a person to interact with a computer system using speech and techniques for performing speech recognition. (Schmandt, ¶97.)

A POSITA would have been motivated to combine. First, Coffman expressly incorporates Kanevsky (Coffman, 37:10-13) and is therefore part of Coffman’s disclosures for purposes of applying it as prior art. *See Paice*, 881 F.3d at 906-907; *Boston Scientific Scimed*, 554 F.3d at 991. Moreover, both references share a co-inventor, Stephane Maes, and identify IBM as the assignee/applicant. (Coffman, Kanevsky face pages.) Therefore, a POSITA would have consulted both of these references to understand the features and functionality of Coffman’s conversational system and/or to actually implement or adapt the system and features it provides. (Schmandt, ¶98.)

Moreover, Coffman already discloses that a conversationally-aware application provides its “language models” to the conversational system in connection with processing a user’s input. A POSITA would have understood that a language model in this context likely refers to an n-gram language model, and

Kanevsky provides express confirmation, explaining that a speech recognition engine can be implemented using a trigram language model. (Schmandt, ¶99 (citing EX1008, pp.0031 (“language model”), 0035 (“n-gram language model”), Kanevsky, 13:50-56).) Moreover, the use of an n-gram language model as Kanevsky teaches would have involved a known technique to implement or improve Coffman’s speech recognition functionality in the same way. (*Id.*)

A POSITA would have had a reasonable expectation of success, as the combination would have involved applying well-known technologies and techniques without undue experimentation. (Schmandt, ¶100 (citing ’209, 1:38-43; Kanevsky, 13:48-61).)

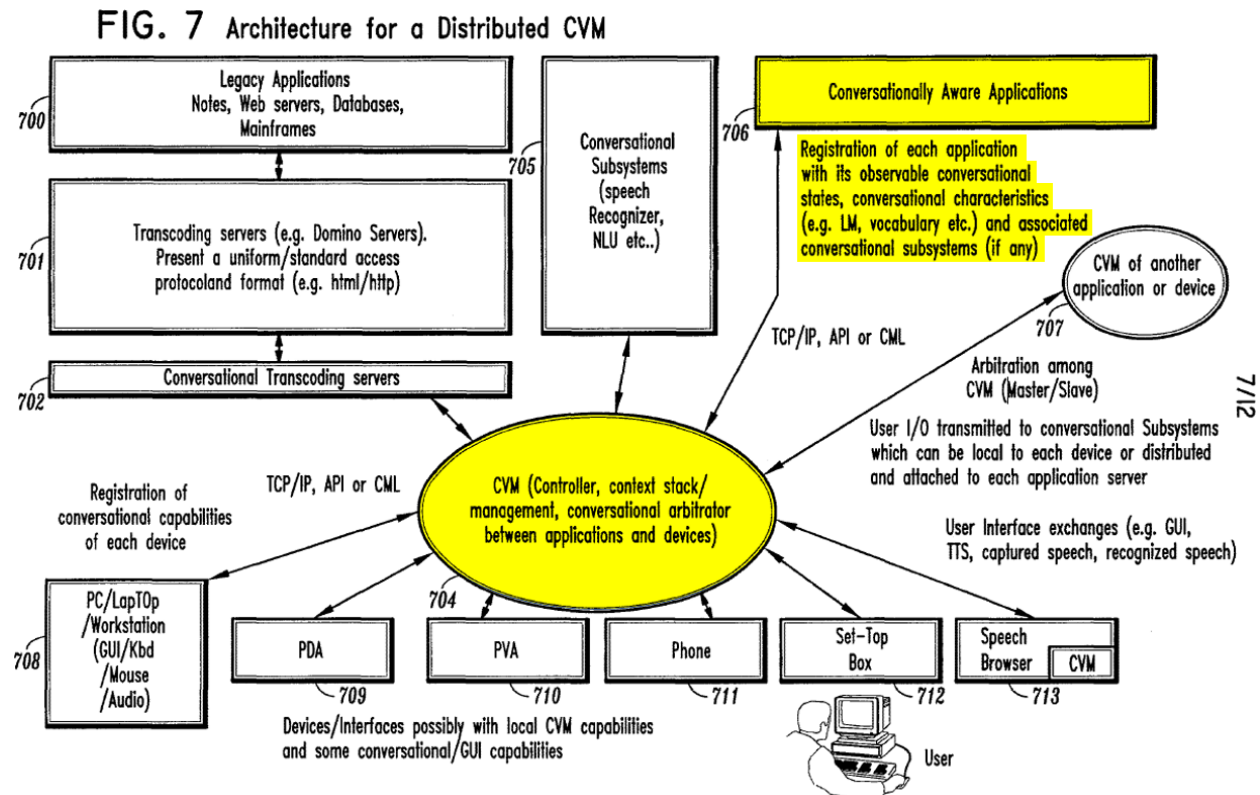
A “**dynamic set of prior probabilities**” is disclosed and obvious in two ways. The **first** way is through the addition or removal of applications with which the conversational system can communicate over time. (Schmandt, ¶101.) Notably, claim 1 recites a “**dynamic set**” of prior probabilities, and not for example, “dynamic prior probabilities.” Therefore, it is enough for the set of prior probabilities itself to be dynamic or changeable (for example, by adding or removing elements to or from the set) even if none of the probability values of the constituent prior probability elements ever changes. With this understanding, Coffman explains that when an application registers with a CVM and its associated engines, it provides its language models. For example:

With CVM running on the desktop and on the PVA and CVM aware smart phone and PDA, the application developer must only hook to the CVM API. It involves registering all its conversational capabilities and requirements:

1. Active vocabulary, finite state grammar and language models to control the application;
- 2 Symbolic mapping if NLU is supported or at list a context state list;
- ...

Again, the API conveys all the information for the CVM to transform queries into application calls and conversely converts output into speech, appropriately sorted before being provided to the user.

(Coffman, 57:22-58:11; *see also id.*, 7:12-14, 13:8-13.) Coffman figure 7 illustrates:



(Coffman, Fig. 7 (highlighting added).)

Coffman further discloses that, over time, the applications registered with the conversational system can change. (Coffman, 25:18-24, 7:12-14.) Therefore, the combination of Coffman and Kanevsky teaches and renders obvious that the set of trigrams (for example) and their associated probabilities used by the CVM changes as applications register with it—i.e., the conversational system maintains a “**dynamic set**” of prior probabilities. (Schmandt, ¶102.)

The **second** way that the prior art discloses and renders obvious a “**dynamic**” set of prior probabilities is through the conventional use of an “adaptive” n-gram language model for speech recognition, where the probabilities associated with the n-grams are dynamically updated with use. (Schmandt, ¶103; EX1008, p.0024 (“adaptive language model”).) Adaptive n-gram language models and their advantages were well known, and their use would have been obvious over **Kneser** (EX1005). (Schmandt, ¶103; Kneser, 1:22-29; *see generally id.*, 1:13-37.)

Rationale and Motivation to Combine (Coffman and Kanevsky with Kneser). It would have been obvious to further combine Coffman and Kanevsky with Kneser’s teachings regarding an adaptive language model, predictably resulting in language models in Coffman’s conversational system having a “**dynamic**” set of prior probabilities. (Schmandt, ¶104.)

Kneser is an analogous reference in the field of the ’209 patent, and would have been reasonably pertinent to problems facing the ’209 inventors, including

techniques for performing speech recognition and how to implement a speech recognition system that uses a “dynamic” set of prior probabilities. (Schmandt, ¶105.)

Kneser provides express motivation for incorporating “adaptive” language model techniques into Coffman’s conversational system—improving speech recognition reliability and performance. (Schmandt, ¶106.) Coffman itself confirms this motivation in the context of collecting user-specific information to perform “more robust recognition.” (*Id.* (citing Coffman, 37:5-10).) Adaptive language models as taught by Kneser provide a way in which Coffman’s system could advantageously improve speech recognition reliability and performance. The use of adaptive language models was well known in the art, and a POSITA would have appreciated that using adaptive language models in Coffman’s conversational system for speech recognition would have improved a similar system in the same way. (*Id.*)

A POSITA would have had a reasonable expectation of success, because the combination would have involved conventional techniques and technologies without undue experimentation. (Schmandt, ¶107.)

Lastly, under the combination, the set of n-gram language model probabilities (“dynamic set of prior probabilities”) is “usable at each stage of processing the received user utterance.” While the claim does not expressly identify any of its

steps as being a “**stage**,” it does describe processing the received user utterance in connection with recognizing words and phrases in the utterance (1[c]). The claim also describes a step of determining the meaning of the utterance (1[d]), although that step more precisely involves processing the recognized words and phrases, and not processing “**the received user utterance**.” Claim 1 therefore appears to recite one “**stage**” of processing the received user utterance (or two, to the extent 1[d] is also found to be such a “**stage**”). The set of n-gram language model probabilities (“**dynamic set of prior probabilities**”) is usable at each of these stage(s). (Schmandt, ¶108.) As explained for 1[c]-[d], the n-gram language model probabilities under the combination are used for the initial step of recognizing speech in the user’s utterance and, in turn, the results of speech recognition are used for the step of natural language understanding to determine a meaning of the utterance (thereby also using the probabilities for that step). (*Id.*) Moreover, as Petitioner’s expert explains, as to 1[d], Coffman’s use of a “statistical parser” “to identify the active context and complete the query” suggests that Coffman’s natural language parsing can use language model probabilities to “**determine a meaning of the utterance**” because the probabilities help inform a word’s meaning and usage. (*Id.* (citing Coffman, 40:12-15; Maes, 18:13-16).)

(c) “recognizing...” (1[c])

Under the combination, the speech recognition process to identify trigram(s) corresponding to the utterance discloses “**recognizing words and phrases contained in the received utterance**” because trigrams comprise words and sequences of words (i.e., phrases). (Schmandt, ¶109 (citing EX1008, p.0053).) And as explained below, recognizing trigrams from the user’s spoken input “**us[es] information in one or more dictionary and phrase tables,**” for at least two independent reasons.

First, the trigrams are recognized “**using information in one or more dictionary and phrase tables**” because a POSITA would have understood and found it obvious to organize the n-gram language model(s) as a **table**. (Schmandt, ¶110 (citing EX1016, p.006 (with n-gram language model, “syntactic knowledge is captured in a large table of word bigrams (pairs), trigrams (triples),...N-grams”), Denenberg (EX1014), ¶0040 (“N-gram language model [] works from tables of probabilities of sequences of words.”))).) Thus, a trigram language model itself comprises “**one or more dictionary and phrase tables**” because it is organized as a table and includes a list of words (“**dictionary**”) and sequences of words (“**phrases**”) to refer to during speech recognition. (Schmandt, ¶110 (citing EX1015, p.005 (“dictionary” encompasses “a list (as of items of data or words) stored in a computer for reference (as for information retrieval or word processing)”).).)

Second, a POSITA would have understood and found it obvious that the words comprising a language model in Coffman would have been based on the “active vocabulary,” alone or in combination with baseforms (pronunciation rules), provided by the application. (Schmandt, ¶111; Coffman, 57:22-27 (application registering its “active vocabulary” with CVM), 60:30-61:2 (describing vocabulary and baseforms as necessary data files), 24:30-25:3 (describing registration including active vocabulary and “baseforms (pronunciation rules)”).) The active vocabulary, alone or in combination with associated baseforms, discloses and renders obvious **“one or more dictionary and phrase tables.”** (Schmandt, ¶111 (citing EX1008, pp.0026 (“active vocabulary”), 0056 (“vocabulary”), 0034 (describing “lexicon” as a collection of “words or idioms”))).) And to the extent the term “dictionary and phrase tables” required not just a list of words and phrases (e.g., idioms) but additional information about them, Coffman explains that application registration includes providing pronunciation rules (called “baseforms”) for the application’s vocabulary. (Coffman, 57:22-27, 60:30-61:2, 24:30-25:3; Schmandt, ¶111.)

A POSITA would have further understood and found it obvious for Coffman’s vocabulary (alone or in combination with baseforms) to be organized as a **“table.”** (Schmandt, ¶112.) A table would have been one of a finite number of identified ways of structuring lexical information in a speech recognition system. (*Id.* (citing EX1008, pp.0033 (“lexical architecture” definition describing lexicon structured as

“table of items”), 0043 (describing “pronunciation table”), 0041-0042 (“phoneme” and “phonemic transcription”), 0038 (“orthography”).) And by using the active vocabulary (via the associated language model), alone or in combination with using pronunciation rules, for speech recognition, Coffman therefore discloses and renders obvious **“recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables”** in a second way.
(*Id.*)

(d) “parsing...” (1[d])

A POSITA would have understood and found it obvious that Coffman’s natural language understanding involved **“parsing the recognized words and phrases.”** (Schmandt, ¶114.) Coffman’s conversational system includes “natural language understanding and natural language parsing system 410” and Coffman repeatedly describes “NL parsing” and “NLU.” (Coffman, 29:1-6, 19:16-19, Fig. 4 (NLU 410), Fig. 6 (NLU 605).) Moreover, Kanevsky explains that NLU can be implemented using “a combination of classical parsing capabilities with key word spotting and speech recognition.” (Kanevsky, 14:1-7.)

Coffman further teaches that parsing the recognized words and phrases as part of natural language understanding is to **“determine a meaning of the utterance.”** This is because Coffman uses NLU as part of the process to determine a “context” for the user’s request to properly carry out the request. In particular, Coffman

explains that the conversational system includes a “context stack” where the “context,” including the state and history of each active process, such as an application process, is tracked. (Coffman, 29:18-30:3, 30:4-10.) The context stack is searched to determine a context that matches the user’s request based on NLU processing. (Coffman, 42:1-5, 41:2-5, 30:8-10.) Coffman therefore discloses **“wherein determining the meaning includes determining a context for the at least one request contained in the utterance.”**

Coffman also discloses and renders obvious that the **“context”** for the request is determined **“based on one or more keywords contained in the recognized words and phrases.”** This is because the context stack is searched to identify the context that “fits the utterance at the level of the NLU symbolic language,” that recognized speech is “run down the context stack **405** to extract the active query and complete it,” and that “active grammars, vocabularies and symbolic language (attribute value n-uple) [map] the actual query.”⁴ (Coffman, 42:1-5, 41:4-5, 30:20-22; Schmandt, ¶118.) A POSITA would have therefore understood and found it obvious that word(s) (**“keywords”**) in the recognized speech would be matched against words in, for example, an active vocabulary, grammar, or NLU

⁴ An “n-uple” refers to a collection of “n” elements; for example, a “4-uple” (or “quadruple”) is a collection of four elements. (Schmandt, ¶118 n.5.)

symbolic language associated with an application to determine the context corresponding to the user's query. (Schmandt, ¶118.) For example, the word "open" in the request "I would like to open this" matches the "open" command in an application's symbolic language associated with a context for the application in the context stack. (Coffman, 6:32-7:3.)

(e) "selecting..." (1[e])

The '209 patent describes "agents" as "autonomous executables that receive, process and respond to user questions, queries and commands" and states that they "provide complete, convenient and re-distributable packages or modules of functionality, typically for a specific domain or application." ('209, 2:50-59.) 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, ¶119.)

Coffman discloses a "**domain agent**": the executable software of a conversationally-aware application (and associated data, such as vocabulary, grammar, language models and NLU symbolic mapping). (Schmandt, ¶120; Coffman, 13:26-29.) Coffman's conversationally-aware application receives, processes, and responds to user questions, queries and commands. (Coffman, 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”).) Coffman’s conversationally-aware application can also provide 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). (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”).) Coffman also describes modular development tools for conveniently building complete and re-distributable applications. (Coffman, 58:12-20; *see also id.*, 58:27-28, 7:29-8:9.) Coffman explains that these applications can be distributed for installation by conversational systems. (Coffman, 52:9-11 & 52:18-21 (describing agents to upgrade or install applications).)

A conversationally-aware application executable is a “**domain agent**” because, as explained, it provides functionality for a particular area or realm, such as a particular subject matter or a device. (Schmandt, ¶121; Coffman, 16:35-17:7, 56:11-14, 58:30-32, 56:22-25, 36:29-31 (“[T]he present invention provides the capability to have a natural dialog with NLU, NLG and mixed initiative across multiple applications, with multiple domains.”).)

1[e] further recites that the “**domain agent**” is “**an autonomous executable.**” Coffman teaches that a conversationally-aware application (“**domain agent**”) is “**an autonomous executable**” because it automatically (without user involvement) registers with a conversational system to be able to receive, process and respond to requests, and moreover, operates automatically in response to the system selecting it to handle a request. (Schmandt, ¶122.) Coffman explains, for example, that “conversational devices or applications [] register their conversational capabilities” and describes “automatically discover[ing] local and network conversationally aware systems.” (Coffman, 7:12-14, 25:18-24; *see also id.*, 23:16-20.) Coffman also explains that “[t]he first task to have its dialog completed by producing a non-ambiguous query is executed” (Coffman, 37:2-4), making clear that an application is automatically selected and carries out a request as a result of processing by the conversational system. (*See also id.*, 42:1-7; Schmandt, ¶122.)

The autonomous nature of Coffman’s conversationally-aware applications is confirmed by **Maes (EX1007)**, which Coffman incorporates by reference and which claims priority to the same provisional patent applications as Coffman. (Coffman, 22:25-29; Coffman 1:2-3 & Maes, 1:4-5; EX1023 (Maes application corresponds to “YO999-113P”); Schmandt, ¶123.) Coffman specifically points to Maes for aspects of its conversational computing system. (Coffman, 22:25-29, 26:21-25, 45:11-22,

61:5-6.) Maes describes techniques for conversationally-aware applications to automatically communicate and coordinate with each other. (Maes, 3:4-10.)

A conversationally-aware application (“**domain agent**”) is “select[ed]...**based on the determined meaning**” because activation of an application for processing the user’s query is based on the application’s association with the determined context (determined as part of “**determin[ing] a meaning of the utterance**” as explained for 1[d]). (Schmandt, ¶123.) Coffman explains:

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 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, 42:1-7; *see also id.*, 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed....”).)

Lastly, as shown for 1[g]-[h] below, a conversationally-aware application (“**domain agent**”) also “**receives, processes, and responds to requests associated with the determined context.**” For example, the conversational system formulates an application call/query associated with a context from the user’s request, and the

application processes and responds by returning output to the conversational system for output to the user.

(f) “formulating...” (1[f])

As explained above, Coffman describes using natural language understanding to translate a recognized utterance into a symbolic mapping and query/call to be carried out by an application. (Schmandt, ¶126; Coffman, 13:8-13 (“CVM 14 to transform queries into application calls”), 30:8-10 (“symbolic language (attribute value n-uple) which maps the actual query”).) For example, Coffman teaches that a user’s utterance “I would like to open this” would be processed into “a set of attributes value pairs: Command: Open, Object: Window” and result in an “open” command being performed. (Schmandt, ¶126 (citing Coffman, 6:32-7:3).) And Coffman, as noted, explains that an application provides its active vocabulary, grammar, and language models to the CVM for controlling the application. (Coffman, 24:31-25:9, 57:22-27, 36:31-32 (“each application will provide the CVM 401 its own parsing and translation arguments”).)

A POSITA would have understood and found it obvious that, in a system such as Coffman’s involving speech recognition using language model(s) and natural language understanding, a grammar associated with an application would be used to formulate the request as a symbolic mapping (such as a set of attribute-value pairs such as “Command: Open, Object: Window”) representing the system’s

understanding and to, in turn, formulate an application query/call in accordance with that understanding. (Schmandt, ¶¶127-128 (citing Coffman, 53:25-26, 29:1-4, 6:32-7:3; EX1008, pp.0035 (“Natural Language Processing”), 0040 (“parsing”)).) Both the symbolic mapping and the application query/call would separately correspond to the claimed **“request” “formulated...in accordance with a grammar used by the selected domain agent to process requests associated with the determined context.”**

Zadrozny (EX1006) confirms that it would have been obvious to formulate a request **“in accordance with a grammar used by”** the selected conversationally-aware application (**“the selected domain agent”**). (Schmandt, ¶129.) **Zadrozny** is referenced in Kanevsky as an example of a semantic analyzer. (Kanevsky, 14:22-25.) Zadrozny describes a “grammar of constructions” associated with an application (such as a calendar application) for performing natural language understanding to formulate a recognized utterance as an intended action and parameters and then translate the action and parameters into “a form that can be executed by the application.” (Zadrozny, pp.0019-0020.) Zadrozny illustrates the technique with the example utterance “I want you to arrange a conference in my office at 5.” (Zadrozny, p.0021.) Using a grammar of constructions, the utterance is “analyzed as *sent(assert, svoc)*, an assertive sentence consisting of a subject, verb,

an object and a complement” and formulated as an action (“schedule”) and series of parameters:

```
[ [ action_name schedule]
  [ event_name
    [ a conference]
  [ event_time
    [ [ minute 0]
      [ hour
        [ 5 am_or_pm]
      [ event_place
        [ my office]
```

(Zadrozny, pp.0021-0022.) This action and parameters are then translated into a form for the application to execute (i.e., an application call/query).

Zadrozny’s process is just like Coffman’s descriptions of formulating a user’s utterance as a set of attribute-value pairs (e.g., “Command: Open, Object: Window”) and then converting them into an application call/query. (Schmandt, ¶130; *see also* Coffman, 30:8-10 (“symbolic language (attribute value n-uple) which maps the actual query”), 13:8-13 (CVM system transforms queries into “application calls”).)

Rationale and Motivation to Combine (Coffman, Kanevsky and Kneser with Zadrozny). It would have been obvious to combine Zadrozny’s teachings regarding processing speech using a grammar of constructions with the conversational computing system taught by the combination of Coffman, Kanevsky and Kneser. The combination would have predictably resulted in the conversational

system in which user requests were formulated in accordance with a grammar used by a conversationally-aware application. (Schmandt, ¶131.)

Zadrozny is an analogous reference in the field of the '209 patent, and would have been reasonably pertinent to problems facing the '209 inventors, including techniques for understanding natural language speech input and how to determine and formulate actions for an application to undertake in response to the input. (Schmandt, ¶132.)

A POSITA would have been motivated to combine. Kanevsky expressly cites Zadrozny as an example of a “semantic analyzer.” (Kanevsky, 14:22-25.) And authors of Zadrozny are identified as being associated with IBM, the assignee/applicant for both Coffman and Kanevsky. (Zadrozny, p.0019 (“IBM Research Division”).) A POSITA therefore would have turned to Zadrozny for detail about how to implement the systems and techniques described in Coffman and Kanevsky. (Schmandt, ¶133.)

A POSITA would have appreciated that the disclosures of Coffman and Zadrozny confirm their fit. (Schmandt, ¶134.) As noted, Coffman does not describe the natural language understanding process in detail, mostly describing at a high level: applications provide grammars, vocabularies, language models and symbolic mappings, a context is matched by “fit[ting] the utterance at the level of the NLU symbolic language” and “CVM 14 [] transform[s] queries into application calls.”

Because Kanevsky expressly cites Zadrozny as an example of a semantic analyzer, a POSITA would have turned to it for its teachings regarding the use of grammars in natural language understanding to translate a recognized utterance into an application call. (*Id.*) Moreover, as noted, Zadrozny's process is just like Coffman's descriptions of formulating a user's utterance as a set of attribute-value pairs and then converting them into an application call/query. (*Id.*)

A POSITA would have had a reasonable expectation of success. (Schmandt, ¶135 (describing Zadrozny's working implementation).)

(g) “invoking...” (1[g])

As explained above, Coffman teaches that the conversationally-aware application associated with the determined context (“**the selected domain agent**”) is invoked to process the user's request. (Coffman, 37:2-4, 13:8-13 & 58:9-11, 40:9-19, 30:4-10.) Under the combination, as explained for 1[f], the conversationally-aware application processes the “**formulated request**” (the symbolic mapping and/or application call/query) because the application carries out processing in accordance with that request. (Schmandt, ¶136.) In Zadrozny's conference scheduling example, the calendar application would carry out processing to schedule a conference in the system in accordance with the specified action and parameters. (*Id.*)

(h) “presenting....” (1[h])

Coffman discloses presenting to the user results generated by the conversationally-aware application (“**the invoked domain agent**”) as a result of processing the symbolic mapping and/or application call/query (“**the formulated request**”). (Schmandt, ¶137.) Coffman describes “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:

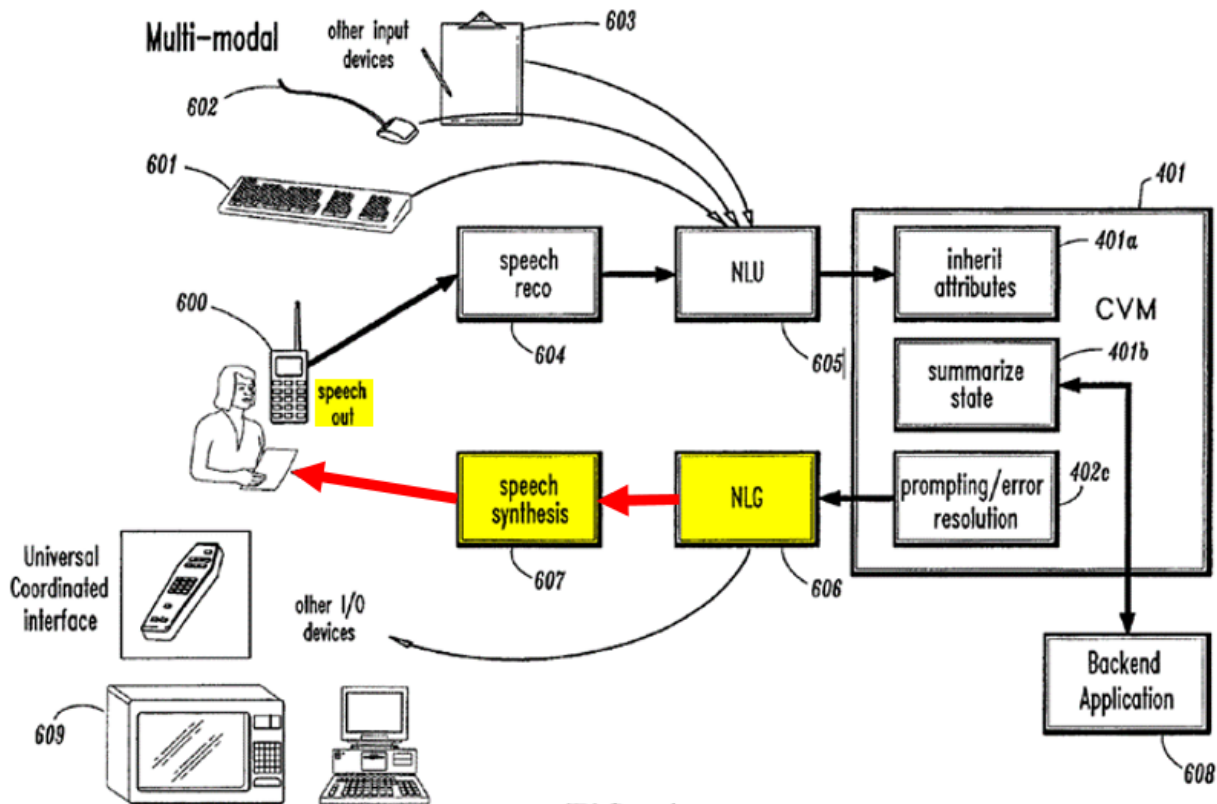


FIG. 6

(Coffman, Fig. 6 (annotations added), 40:9-19; *see also id.*, 13:8-13 & 58:9-11, 39:3-5.)

2. Claim 2

It would have been obvious for claim 1's "**recognizing**" step to include the following. (Schmandt, ¶138.)

(a) "**dynamically updating...**" (2[a])

Under the combination, because the **prior probabilities** are included in the trigram language model(s) ("**the dictionary and phrase tables**"), 2[a] corresponds to **dynamically updating** the trigram language model(s) in accordance with model adaptation as taught by Kneser. (Schmandt, ¶¶139-140.)

(b) "**determining...**" (2[b])

Coffman discloses "**determining an identity of the user**" because its conversational system includes functionality for "speaker recognition" and "speaker identification," such as "text-independent speaker identification" using a user's command or request, illustrated in figure 4:

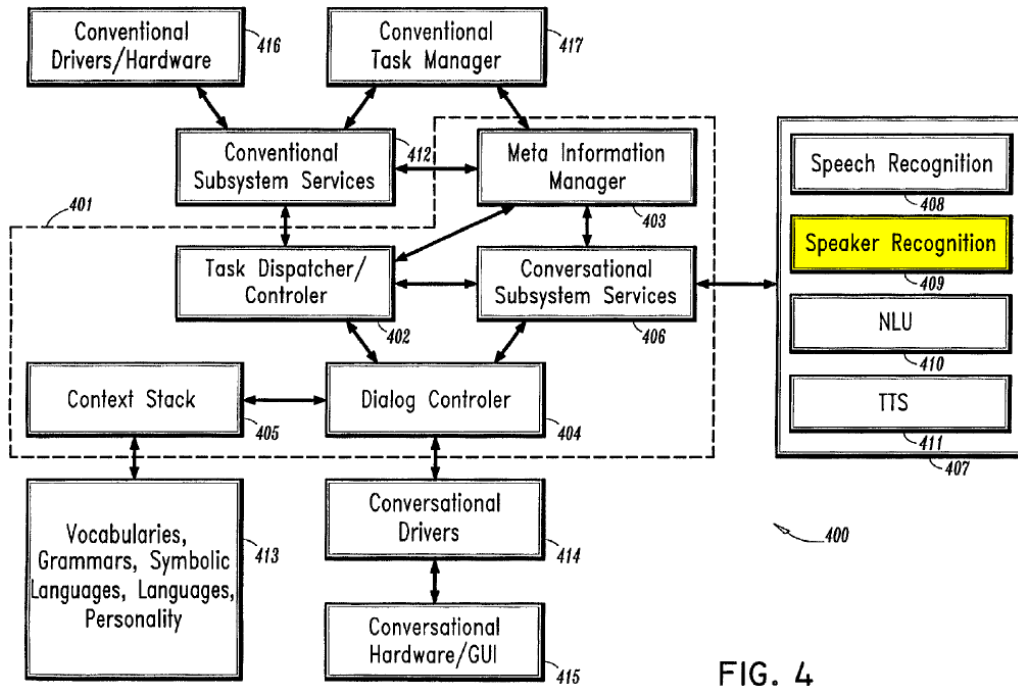


FIG. 4

(Coffman, Fig. 4 (highlighting added); *see also id.*, 5:1-4 (“speaker recognition”), 29:1-4, 15:13-16:24 (listing speaker-identification-related “conversational foundation classes” including “Acoustic Identify speaker,” “Acoustic Verify speaker” and “Speech biometrics identify.”).) Coffman further discloses that “[i]n a speech-based system, each command can be used to perform text-independent speaker identification” (Coffman, 47:25-27; *see also id.*, 8:16-18), which involves using speech signal features to determine the speaker’s identity and does not require the utterance to comprise specific linguistic content. (Schmandt, ¶141 (citing EX1008, pp.0047 (“speaker recognition”), 0051 (“text-independent speaker recognition system”))).)

Kanevsky, which Coffman specifically cites for speaker identification (Coffman, 37:5-13), further describes text-independent speaker recognition that involves comparing “**voice characteristics**” of the user’s utterance with a model to verify the user’s identity. (Kanevsky, 6:66-7:8, 6:12-15 (“The identity claim may be made from...voice characteristics...”), 2:6-8 (describing prior art systems that “employ voice-based verification” using “acoustic characteristics of the utterances spoken by the user”).) The rationale and motivation to combine Coffman and Kanevsky in the analysis of claim 1 applies here also. (Schmandt, ¶142.)

(c) “**associating...**” (2[c])

2[c] would have been obvious in at least two different ways. First, Coffman’s process of matching the words and phrases to identify an acceptable context discloses “**associating the recognized words and phrases...with the identity of the user, the determined context, and the at least one request.**” This is because, as Coffman explains, each context can be associated with a user identity, and a context is also associated with the user’s request. With respect to “**the identity of the user,**” Coffman 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).” (Coffman, 48:1-3; *see also id.*, 46:26-31 (“the stack of contexts should contain the identity of the user as the most recently authenticated identity”), 48:11-

14.) With respect to “**the determined context**” and “**the at least one request,**” as discussed, Coffman teaches that the words and phrases are used to match a context and its associated formulated request (e.g., in symbolic language or application call/query). (Coffman, 42:1-7; *see also id.*, 37:2-4.) Therefore, as a result of identifying a context, the words and phrases are associated with the user’s identity, the determined context, and the request associated with the context. (Schmandt, ¶143.)

2[c] further recites “**associating...a pronunciation of the recognized words and phrases with the identity of the user, the determined context, and the at least one request.**” This would have been obvious in at least two separate ways. **First**, as discussed for 1[c], Coffman explains that application registration involves providing baseforms (pronunciation rules) for the active vocabulary to the conversational system. (Coffman, 57:22-27, 60:30-61:2, 24:30-25:3.) A POSITA would have therefore understood that successful recognition of words and phrases would have involved successfully matching the user’s utterance to the pronunciation rules. (Schmandt, ¶144.) Therefore, the subsequent matching of the recognized words and phrases to a context would also result in associating the pronunciations specified by the baseforms of those words and phrases with the user’s identity, the determined context, and the request associated with the context. (*Id.*)

Second, as explained for 2[b], identifying the user based on voice characteristics of the user's utterance is disclosed by and obvious over Coffman and Kanevsky. A POSITA would have understood that a user's pronunciation of words and phrases would be reflected in the voice characteristics of their utterance. (Schmandt, ¶145 (citing Kanevsky, 9:39-42 ("prosody" & "accent"), 10:34-38, 11:53-58; EX1008, pp.0056 ("voice characteristic" includes "dialect"), 0044 ("prosody"); EX1015, p.004 ("dialect" includes "pronunciation")).) Therefore, a POSITA would have understood and found it obvious that, by performing speaker identification in accordance with 2[b], the user's pronunciation of the recognized words and phrases would have been **"associated" "with the identity of the user, the determined context, and the at least one request."** (*Id.*)

2[c] further recites that the **"associating"** takes place **"when the recognized words and phrases satisfy a predetermined confidence level."** This also would have been obvious over Coffman and Kanevsky. The concept of a confidence level has been well known in the art since at least the 1970s. (Schmandt, ¶146 (citing EX1017, pp.002-003).) Coffman itself mentions "confidence and ambiguity results" (Coffman, 27:16), applications providing a confidence level for its interpretation of an input (Coffman, 31:26-28) and "provid[ing] feedback to the user for incomplete or ambiguous queries/commands" (Coffman, 40:16-17). It would have been obvious in view of Kanevsky to perform the claimed **"associating"** only

when speaker identification based on the recognized words and phrases satisfied a predetermined confidence level. (Schmandt, ¶146.) Doing so would help ensure that the system did not incorrectly associate an input with a particular user, context and query and would further the security objectives described in both Coffman and Kanevsky. (*Id.* (citing Coffman, 37:5-14 (describing conversational biometrics for query authorization), 8:10-15 (describing preventing unauthorized access), 46:17-32 (describing user authentication and authorization)).) For example, Kanevsky describes calculating a score from text-independent speaker recognition, comparing the score to “a predetermined threshold value,” and allowing access only when the score is “above or within an acceptable predetermined range of the threshold value.” (Kanevsky, 6:66-7:24.) **Ground 2** discusses a second way in which the claimed “associating...when the recognized words and phrases satisfy a predetermined confidence level” would have been obvious.

As noted, 2[c] would have also been obvious in a second way: storing a history of the user’s actions in a “global history” and “meta-information repository.” (Schmandt, ¶147.) For instance, Coffman describes a “global history” that stores “the history of the dialog between user and machine” that includes “the information that is associated with all the applications and actions taking [sic] during a conversational session.” (Coffman, 21:28-32.) Coffman also describes a “meta-information repository **21** [sic; **218**] that includes a user-usage log based on user

identity.” (Coffman, 21:14-15.) “Past history and context and other information” is “stored as part of the meta information.” (Coffman, 30:11-12.) Because Coffman describes storing usage and context information in a global history and/or meta information repository, a POSITA would have understood the claimed “**associating**” to take place when the relevant information was stored as past information, which would record the association between the recognized words and phrases and the context, and the user identity and request associated with the context. (Schmandt, ¶147.) It would also have been obvious for this “**associating**” to take place “**when the recognized words and phrases satisfy a predetermined confidence level**” because, as explained above, it would have been obvious for the conversational system to match a context (and therefore record that match in the history) only when speaker identification based on the recognized words and phrases satisfied a predetermined confidence level. (*Id.*)

3. Claim 4

It would have been obvious for “**determining the context for the at least one request contained in the utterance**” to include the following steps. (Schmandt, ¶148.)

(a) “**matching...**” (4[a])

As explained for 1[d], a POSITA would have understood and found it obvious that in Coffman’s system, word(s) in the recognized speech (“**the keywords**

contained in the recognized words and phrases”) would be “**matched**” against a “**set of keywords**”—words in, for example, an active vocabulary, grammar or NLU symbolic language provided by a conversationally-aware application—to determine the context corresponding to the user’s query. (Schmandt, ¶149.) Using Coffman’s example, the word “open” in the request “I would like to open this” matches an “open” command in an application’s symbolic language associated with a context. (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.) Kanevsky, as noted, similarly teaches performing natural language understanding “preferably” with a “combination of classical parsing capabilities with key word spotting.” (Kanevsky, 14:1-4.)

An active vocabulary, grammar, and NLU symbolic language in Coffman discloses “**a dynamically updated set of keywords**” for at least three independent reasons. (Schmandt, ¶150.) **First**, each is a “**a dynamically updated set of keywords**” because, as explained above for 1[b], they are dynamically provided by an application to a conversational system to update the system’s recognition and understanding capabilities. (Schmandt, ¶150; Coffman, 57:22-58:11, 7:12-14, 13:8-13, 25:18-24.) **Second**, Coffman teaches that the particular words in an active vocabulary, grammar, and NLU symbolic language that are searched for when trying

to match a context to a user's query change depending on the context—for example, completing a query by matching particular fields. (Coffman, 42:1-7; *see also id.*, 41:2-5, 37:29-38:2, 41:16-21.) **Third**, Coffman teaches that a user can modify the data files (such as vocabulary, grammar, and NLU symbolic language) for an application. (Coffman, 58:12-17; *see also id.*, 13:8-10 (listing “appropriate data files” including “contexts, finite state grammars, vocabularies, language models, symbolic query maps etc.”).) A user modifying a conversationally-aware application's data files is another way that Coffman discloses a dynamically updated set of keywords.

Lastly, as explained for claim 1, under the combination the **“dynamic set of prior probabilities or fuzzy possibilities”** corresponds to probability information for trigram language model(s) used for speech recognition. The **“dynamically updated set of keywords [is] associated with”** that probability information because, first, they are associated with the same conversationally-aware applications, and second, a context is matched based on the recognized words and phrases resulting from using that probability information in recognizing those words and phrases. (Schmandt, ¶151.)

(b) “scoring...” (4[b])

Coffman discloses determining acceptability of **“one or more possible contexts for the matched keywords using...a context stack of one or more recent**

contexts.” (Schmandt, ¶152.) Coffman describes searching a context stack 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 further understood and found it obvious that Coffman’s technique of determining whether a context was an “acceptable match” or “too improbable” would involve “**scoring**” (a well-known concept) the context, such as determining the likelihood or goodness of the match to compare it against some level of acceptability. (Schmandt, ¶153.) Kanevsky confirms the obviousness of “**scoring**” a context, by expressly describing multiple different analogous situations (speaker recognition and semantic understanding) in which a score is generated (and compared against a predetermined threshold value) to determine further action to take. For example, Kanevsky describes a process for recognizing the speaker that involves both scoring a user’s spoken input against a stored answer and scoring a user’s voice sample against a stored user model. (Kanevsky, 6:34-42, 7:12-20.) A combined score is then compared “to a predetermined threshold value and if the score is one of substantially equivalent to and above the threshold value,” the speaker is permitted access to the system. (Kanevsky, 3:40-48; *see also id.*, 7:12-20.)

A POSITA would have found it obvious to apply these scoring/matching techniques to the context matching described in Coffman. (Schmandt, ¶154.) The rationale and motivation to combine Coffman and Kanevsky discussed above applies here also. (*Id.*) Specifically, Coffman describes rejecting a context where a match is “too improbable,” which confirms that there is some level of “probability” required. Kanevsky confirms that using a scoring technique would have been a straightforward and obvious choice—applying Kanevsky’s known “scoring” techniques to improve Coffman’s methods for determining a context matching a user’s query amounts to improving a similar method in the same way, with predictable results. (*Id.*)

(c) “determining....” (4[c])

As explained for 4[b], a POSITA would have understood and found it obvious that **“determining a context”** would involve **“selecting one or more of the possible contexts having a score satisfying a predetermined confidence level.”** (Coffman, 42:1-5, 40:24-27; Kanevsky, 3:40-48, 6:34-42, 7:12-20.) A POSITA would have understood that a context score would be compared to a **“predetermined confidence level”** because the score would reflect the system’s confidence whether the context was acceptable. (Schmandt, ¶155.)

4. Claim 5

It would have been obvious for the step of “**determining the context for the at least one request contained in the utterance**” to include the following.
(Schmandt, ¶156.)

(a) “requesting...” (5[a])

Coffman teaches that the conversational system will prompt the user to verify their input as the system understood it (“**recognized words and phrases**”) if the system determines that input was incomplete or ambiguous—e.g., context scores do not satisfy a predetermined confidence level. (Schmandt, ¶157; Coffman, 41:16-21 (“generate a request (a prompt) for the missing or ambiguous information and update the context (requested missing fields)”); *see also* Coffman, 37:2-4, 40:9-19 (“provide feedback to the user for incomplete or ambiguous queries/command”).)

(b) “determining...” (5[b])

Because Coffman describes prompting a user to provide missing or ambiguous information, resulting in an updated/completed context, a POSITA would have understood and found it obvious for Coffman’s system to “**determine the context based on [the additional] information provided by the user.**” (Schmandt, ¶158.) A POSITA would have understood and found it obvious for the user to provide the additional information via “**one or more subsequent natural**

language speech utterances.” (*Id.* (citing Coffman, 14:27 (“spoken prompt”), 5:18 (“speech input”)).)

C. Ground 2: Claim 2 is Obvious Over Ground 1 Prior Art in View of Maes.

1. Claim 2

Ground 2 addresses the second way in which it would have been obvious for the claimed **“associating”** in 2[c] to occur **“when the recognized words and phrases satisfy a predetermined confidence level.”** In all other respects, the mapping and analysis of claim 2 provided in Ground 1 applies here also.

This portion of claim 2[c] would have also been obvious where the claimed **“associating”** occurs only when Coffman’s speech recognizer’s determination that the recognized words and phrases correspond to the user’s utterance meets a certain confidence threshold/score. (Schmandt, ¶160.) If a threshold confidence level is not met, the system prompts the user to repeat themselves, for example, rather than proceeding with further processing and query handling.

For example, Coffman incorporates by reference **Maes (EX1007)**. (Coffman, 22:25-29.) Maes describes rejecting speech recognition results “based on confidence measures” and determining whether speech recognition results meet “a given threshold.” (Maes, 16:20-17:7 (“the rejection of results for speech recognition and NLU may occur...based on confidence measures....[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”), 17:23-18:16.)

Rationale and Motivation to Combine (Ground 1 Prior Art and Maes). It would have been obvious to combine Maes’s teachings about using confidence levels in connection with speech recognition results with the conversational system of Coffman (as informed and/or adapted by the teachings of Kanevsky, Kneser and Zadrozny). The combination would have predictably resulted in Coffman’s conversational system in which determinations about further action to take—such as identifying a context to activate—depend on whether speech recognition results meet a certain confidence threshold. (Schmandt, ¶162.)

Maes is an analogous reference in the field of the ’209 patent, and would have been reasonably pertinent to problems facing the ’209 inventors, including techniques for improving the reliability of speech recognition results and in determining actions to take based on those results. (Schmandt, ¶163.)

A POSITA would have been motivated to combine. (Schmandt, ¶164.) Maes and Coffman claim priority to the same United States provisional patent applications. (Coffman, 1:2-3; Maes, 1:4-5.) Maes and Coffman incorporate each other by reference, with each referring to each other by title and IBM docket number. (Coffman, 22:25-29; Maes, 7:12-16.) Additionally, the first-named inventor on

Maes is also an inventor on Coffman. (Maes, face page (75); Coffman, face page (75).)

Additionally, as noted, Coffman itself mentions “confidence and ambiguity results,” applications providing a confidence level for its interpretation of an input, and “provid[ing] feedback to the user for incomplete or ambiguous queries/commands,” but does not detail how confidence levels may be used by the system. (Coffman, 27:16, 31:26-28, 40:16-17.) A POSITA would have turned to Maes for further information and teachings about how confidence levels could be incorporated into Coffman’s conversational system, especially because of the relationship between the references noted above. (Schmandt, ¶165.)

A POSITA would have appreciated the advantages of incorporating speech recognition confidence levels into Coffman’s system, such as improved confidence in the results and subsequent processing performed based on the results. (Schmandt, ¶166.) As noted, Coffman and Kanevsky both teach that improved system robustness and reliability is desirable, and the use of confidence levels as taught by Maes would further those aims. Incorporating Maes’s teachings would have improved a similar system in the same way. (*Id.*)

A POSITA would have had a reasonable expectation of success because the combination would have involved conventional techniques and technologies, without undue experimentation. (Schmandt, ¶167 (citing Maes, 17:4-7).)

D. Ground 3: Claim 3 is Obvious Over Ground 2 Prior Art in Further View of Maes '101 and Franceschi.

1. Claim 3

It would have been obvious for claim 1's "**recognizing**" step to include the following. (Schmandt, ¶168.)

(a) "requesting..." (3[a])

As noted for 2[c] in **Ground 2**, **Maes** describes rejecting speech recognition results when a confidence measure is below a given threshold. But rather than outright rejecting the user's utterance and, for example, asking the user to repeat themselves, it would have been obvious to prompt the user to spell word(s). (Schmandt, ¶169.) This would further the goal of an adaptive language model to be able to better recognize a user's utterances. (Schmandt, ¶170.) Voice spelling would also provide a convenient way to provide input in situations where other forms of input were limited or less desirable. (*Id.*) **Maes '101 (EX1010)**, explains that "unrecognized utterances can be defined by the user during enrollment or later enrollment correction by prompting the user to spell the word whenever speech recognition is unsuccessful." (Maes '101, 6:52-55.) Maes '101 discloses that, by spelling words, the system's performance can be improved and adapted to the user by adding the spelled word to the speech recognizer's vocabulary. (Maes '101, 6:55-66.)

“Phonetic alphabet spelling” Issue: While 3[a] simply recites “**requesting the user to spell,**” 3[b] refers to “**a phonetic alphabet spelling of the requested words and phrases.**” It would have been obvious for the user to provide a “**phonetic alphabet spelling**” of the words and phrases. (Schmandt, ¶171.) This is confirmed by **Franceschi (EX1011)**, which explains motivations and benefits of phonetic alphabet spelling, and that such functionality was routine and conventional, with commercially-available speech recognition systems already including it. (Franceschi, 1:38-52, 2:8-13.)

Rationale and Motivation to Combine (Ground 2 Prior Art with Maes ’101 and Franceschi). Further combining Ground 2 prior art with Maes ’101 and Franceschi would have been obvious. The combination would have predictably resulted in Coffman’s conversational system with the added ability to prompt the user to provide a phonetic alphabet spelling if speech recognition results failed to meet a certain confidence level. (Schmandt, ¶172.)

Maes and Franceschi are analogous references in the field of the ’209 patent, and would have been reasonably pertinent to problems facing the ’209 inventors, including techniques for receiving speech input from a user and improving the reliability of speech recognition results by resolving ambiguity. (Schmandt, ¶173.)

A POSITA would have been motivated to combine. Coffman, Maes and Maes ’101 all name a common co-inventor, Stephane Maes, and identify the same

assignee/applicant, IBM. (Coffman, Maes, Maes '101, face pages.) Franceschi likewise identifies IBM as the assignee. (Franceschi, face page.) Given the connections between these references, therefore, a POSITA would have appreciated that Maes '101 and Franceschi could be considered to understand Coffman's conversational system. (Schmandt, ¶174.)

Moreover, as noted, Coffman teaches that the system will prompt the user to verify their input if the system determines that input was incomplete or ambiguous. (Coffman, 41:16-21, 37:2-4, 40:9-19.) Franceschi teaches that “it frequently occurs that the best approach for an unrecognized, difficult, or new word is for the speaker to spell it.” (Franceschi, 1:38-40.) A POSITA would have therefore appreciated that this “frequently best approach”—prompting a user to spell a word—would help resolve incomplete or ambiguous input, and that the ability to provide a phonetic alphabet spelling as taught by Franceschi would help avoid ambiguity when processing the user's spelling input, resulting in an improved user experience. (Schmandt, ¶175.)

A POSITA would have had a reasonable expectation of success, as it would have involved conventional techniques and technologies without undue experimentation. (Schmandt, ¶176 (citing Franceschi, 2:8-13).)

(b) “updating...” (3[b])

Updating information in the trigram language model(s) (“**the dictionary and phrase tables**”) “**based on a phonetic alphabet spelling of the requested words and phrases**” would have been obvious. As noted for 3[a], Maes ’101 discloses that, by spelling words, the system’s performance can be improved and adapted to the user, by adding the spelled word to the speech recognizer’s vocabulary. (Maes ’101, 6:55-66.) Adding words to a speech recognizer’s vocabulary would also further Kneser’s goal of improving a speech recognizer’s language model, “[s]ince the quality of the adaptation of the language model values has a considerable influence on the reliability of the speech recognition.” (Kneser, 1:25-29.) It would have been obvious, therefore, to update trigram language model(s) used for speech recognition to incorporate information about the additional vocabulary provided by the user (“**updat[e] the information in the dictionary and phrase tables**”). (Schmandt, ¶178.)

With respect to “**the phonetic alphabet spelling provided by the user in one or more subsequent natural language speech utterances,**” under the combination, and as Maes ’101 teaches, the spelling would take place by voice after the user is prompted (“**in one or more subsequent natural language speech utterances**”). (Schmandt, ¶179; Maes ’101, 6:52-66.)

E. Ground 4: Claim 6 is Obvious Over Ground 1 Prior Art in Further View of Chai.

1. Claim 6

It would have been obvious for claim 1's "**formulating**" step to include the following. (Schmandt, ¶180.)

(a) "determining..." (6[a])

As explained for 1[f], a POSITA would have understood that, in a system like Coffman's involving speech recognition using language model(s) and natural language understanding, a grammar associated with an application would be used to understand the recognized speech so that a symbolic mapping and application query/call could be formulated in accordance with the understood meaning. (Schmandt, ¶181.) This is confirmed by Zadrozny, which discloses a "grammar of constructions" that is likewise used to determine an intended action and its parameters for an application to execute from a user's natural language input. (Zadrozny, pp.0019-0020 (describing parsing, determining intended action and parameters and executing the action by the application); *see also id.*, p.0019 (describing grammar of constructions and MINCAL, natural language speech-enabled interface to on-line calendar system).) Therefore, under the combination, where the application would be a conversationally-aware application of Coffman ("**the selected domain agent**"), the determined action and parameters for criteria

using Zadrozny’s “grammar of constructions” would correspond to “**a request in the grammar used by the selected domain agent.**” (Schmandt, ¶181.)

For example, Zadrozny describes using a grammar of constructions to translate the statement “I want you to arrange a conference in my office at 5” into an action (“action_name = schedule”) with various criteria and parameters (e.g., “event_name [a conference]”; “event_place [my office]”):

For example, the sentence *I want you to arrange a conference in my office at 5* is analyzed as *sent(assert, svoc)*, an assertive sentence consisting of a subject, a verb, an object and a complement.

...

This result of parsing is then interpreted by the domain interpreter to produce:

```
***Slots:
[ [ action_name schedule]
  [ event_name
    [ a conference]
  [ event_time
    [ [ minute 0]
      [ hour
        [ 5 am_or_pm]
      ]
    ]
  [ event_place
    [ my office]
```

(Zadrozny, pp.0021-0022.)

As shown in the excerpt above, Zadrozny identifies the various fields for actions and criteria/parameters as “slots” (“***Slots:”).⁵ Zadrozny’s technique of mapping the words and phrases of the parsed input to various slots related to a grammar of constructions is consistent with Coffman’s descriptions of determining a context and query by filling in symbolic fields for a natural language query. (Schmandt, ¶183; Coffman, 38:15-16, 42:4-5, 30:8-10.)

Zadrozny does not expressly disclose that an action based on a grammar of constructions (“**a request in the grammar used by the selected domain agent**”) would have “**one or more required values and one or more optional values**” but this would have been obvious over Chai (EX1012).

Like Coffman and Zadrozny, Chai describes techniques for processing a user’s natural language request to determine an application action to execute. Like the symbolic field filling described in Coffman and the slot filling disclosed in Zadrozny, Chai describes filling “slots” in “action templates” based on natural language input to determine the application action to execute. (Chai, 4:51-5:11 (“A natural language parser parses one or more natural language phrases...into one or more logical forms (parsed user input)....[M]odified logical forms are matched

⁵ As Petitioner’s expert explains, the “slots” technique was well known in the art. (Schmandt, ¶183 n.6.)

against a set of predefined action templates, the best matching template is selected, corresponding action requests are sent to an action manager...”).)

For instance, in an online shopping example, Chai describes processing two user requests, “I want to buy one 30 inch wide White Color Cooktop with Electric PowerSource and with Electric:Radiant Glass CookTop Surface and of KenMore company” followed by “I want to buy one Black CookTop of previous type.” (Chai, 12:28-32.) Chai explains that the user’s input is used to fill various slots of matched action templates, with some slots being “**required**”—for example, a value for the “item name” slot is required:

Suppose the user input **205** was “I want to buy one 30 inch wide White Color Cooktop with Electric PowerSource and with Electric:Radiant Glass CookTop Surface and of Ken-More company” followed by another request “I want to buy one Black CookTop of previous type” Before taking discourse context into consideration, the following slots **807** in matching templates **806** with “buy” transaction are filled.

First Template (Used for Pants and Shirts):

TRANSACTION: buy

TEMPLATE SLOTS:

item name: CookTop

sku:

quantity: **1**

size:

color: Black

company:

REQUIRED SLOTS: item name, quantity, company

(Chai, 12:32-44 (highlighting added); *see also id.*, 12:45-56 (second template), 12:21-22 (“choosing the template with the most number of required slots filled”).)

On the other hand, referring to the example action template above, because a value for the “color” slot is not identified as being required, a POSITA would have understood it to be optional. Chai therefore discloses both “**one or more required values**” (e.g., the value for the “item name” slot in Chai’s example) and “**one or more optional values**” (e.g., the value for the “color” slot in Chai’s example). (Schmandt, ¶186.)

Rationale and Motivation to Combine (Ground 1 Prior Art with Chai).

It would have been obvious to combine Chai’s teachings regarding required and optional slot values for action templates to Zadrozny’s technique for formulating a request using a grammar of constructions. This would have predictably resulted in Coffman system’s step of determining the call/query for the conversationally-aware application (“**selected domain agent**”) to include determining required and optional values for the application call/query (“**determining one or more required values and one or more optional values for a request in the grammar used by the selected domain agent**”). (Schmandt, ¶187.)

Chai is an analogous reference in the field of the ’209 patent, and would have been reasonably pertinent to problems facing the ’209 inventors, including how to formulate application queries using natural language. (Schmandt, ¶188.)

A POSITA would have been motivated to combine. Chai lists Wlodek Zadrozny, lead author of Zadrozny (**EX1006**), as a co-inventor and identifies IBM

as the assignee. (Chai, face page.) A POSITA would have appreciated Chai for understanding Zadrozny's techniques for formulating application queries and Coffman's conversational system. (Schmandt, ¶189.) While Chai focuses on online shopping examples, Chai makes clear that its techniques are applicable in other situations similar to those discussed in Coffman, such as retrieving stock quotes. (Chai, 5:10 ("Retrieving the latest price of a stock"), 8:16-65 (describing "stock trading application"); Coffman, 42:50-52 ("check the news or stock quotes").) A POSITA would have therefore appreciated the broad applicability of Chai's techniques to formulating queries for conversationally-aware applications in Coffman's system. (Schmandt, ¶190.) A POSITA would have also appreciated that the ability to formulate queries with required and optional values would have benefitted Coffman's system by providing a more flexible system for responding to user utterances. For example, the ability to provide additional, but not strictly necessary, information as part of a request would enable a user to more quickly and directly locate desired information rather than receive less precise information or engage in a lengthier colloquy with the system to refine the response to the user's request. (*Id.*)

A POSITA would have had a reasonable expectation of success, as it would have entailed well known and conventional computer programming techniques without undue experimentation. (Schmandt, ¶191.)

(b) “extracting...” (6[b])

The '209 patent generally describes “**criteria**” as attributes of a request and “**parameters**” as the values of the attributes (although they are also described somewhat interchangeably). For example, for a request in a weather context, a criterion could be location, and the parameter would be the specific value for the location (e.g., “Alexandria”). ('209, 18:26-28; *see also id.*, 25:11-16 (same), 11:58-60.)

“**[E]xtracting one or more criteria and one or more parameters from the keywords contained in the recognized words and phrases**” would have been obvious. As explained, Coffman describes determining a context by “fit[ting] the utterance at the level of the NLU symbolic language” (Coffman, 42:4-5) and filling symbolic fields for a query (Coffman, 38:15-16, 30:8-10, 41:16-21, 6:32-7:3). Coffman does not provide an example of how exactly an utterance is “fit...at the level of the NLU symbolic language” or how “symbolic fields are filled,” but it would have been obvious under the combination with Zadrozny and Chai to extract criteria and parameters from words contained in the recognized utterance. (Schmandt, ¶193.) For example, as explained, Chai describes analyzing a user’s input (“I want to buy one Black CookTop of previous type”) to identify various slots (“**one or more criteria**”) and their values (“**one or more parameters**”) for an action template.

It would have also been obvious for the claimed **“extracting”** to **“us[e] procedures sensitive to the determined context.”** As explained, Coffman describes using context to determine a conversationally-aware application and call/query. Therefore, by matching a user’s words and phrases to fields in the symbolic language of an application call/query, Coffman’s techniques, under the combination, would have predictably resulted in **“extracting one or more criteria and one or more parameters...using procedures sensitive to the determined context.”** (Schmandt, ¶194.)

(c) **“inferring...”** (6[c])

Chai discloses filling “slots” in action templates with values based on words and phrases input by a user, such as the user’s statement, “I want to buy one Black CookTop of previous type.” (Chai, 12:28-32.) The user’s statement, however, does not fill all required slots in the action template(s)—e.g., a value for the required “company” slot is missing. (Chai, 12:28-56.) Chai teaches that missing slots can be filled in by **“inferring”** values from context. (Chai, 12:1-20 (“...determine if any of the missing slot **807** values can be inferred from context...”).) Chai therefore discloses **“inferring one or more criteria and one or more parameters”** by reviewing context to infer that a value (**“parameter”**) corresponds to a particular slot (**“criteria”**) whose value is missing from an action template. (Schmandt, ¶¶195-196.)

Applying Chai's techniques to Coffman would have been obvious. (Schmandt, ¶197.) As noted, Coffman discloses a "context" for an active process that includes similar information to Chai, including "past history of the discourse, its current state, and the specific characteristics and full query arguments" and "any data needed for input understanding." (Coffman, 29:18-30:3.) A POSITA would have therefore readily appreciated that Chai's techniques to "infer" criteria and parameters could be applied to Coffman's system, and recognized the similar benefits that would flow from it, including the ability to complete a user's query without prompting the user to supply information that the system could determine on its own. (Schmandt, ¶197.) It therefore would have been obvious that recognized speech using trigram probabilities ("**using the dynamic set of prior probabilities or fuzzy possibilities**") from an utterance could provide context information used to fill in missing slot values, predictably resulting in the "**inferring**" step taking place by "**using the dynamic set of prior probabilities or fuzzy possibilities.**" (Schmandt, ¶198.)

(d) "**transforming....**" (6[d])

6[d] simply corresponds to the step of filling in required and optional query fields (or action "slots") based on a grammar of constructions as taught by the combination of Coffman, Zadrozny and Chai. For example, "**the extracted and the inferred criteria and parameters**" correspond to required and optional slots and

their values, which are recognized in words and phrases from the user. The recognized information would have been transformed into a request having a format consistent with the grammar of constructions and its slots as taught by Zadrozny, which would in turn be transformed into a call/query for the conversationally-aware application to process. And both of these formats would have been “**a format compatible with the grammar used by the selected domain agent**”—the initial transformation because it would use the grammar of constructions to identify and fill in slots, and the second transformation (for an application call/query) because it is based on and derived from that initial transformation. (Schmandt, ¶199.)

A POSITA would have also understood and found it obvious for “**all of the required values and one or more of the optional values**” to have been included in the transformed requests. (Schmandt, ¶200.) As noted, Coffman describes executing “completed” queries, while providing feedback for “incomplete or ambiguous queries/commands.” (Coffman, 40:9-19.) Chai teaches that a “complete” query can involve a query where all required slots are filled and some or all of the optional fields are filled. (Chai, e.g., 13:15-19.)

F. Ground 5: Claims 7-9 and 12 are Obvious Over Ground 1 Prior Art in Further View of Redfern.

1. Claim 7

Claim 7 would have been obvious in further view of Redfern (EX1013). Redfern describes a meta-search application where a user can provide a single

natural language query to the application, and the application will formulate queries to multiple search engines and return what the application deems the best search results by evaluating the results using the natural language query. (Redfern, Abstract.)

Under the combination with Redfern, “**the invoked domain agent**” would take the form of a meta-search application configured in accordance with Coffman to be conversationally-aware. (Schmandt, ¶202.) The meta-search application receives a natural language query and evaluates it to produce queries to submit to multiple information sources, such as the well-known Lycos, AltaVista and Excite web-based search engines. (Redfern, 2:52-3:9, 4:8-34, 8:66-9:12, 15:43-65.) The application then scores the results and extracts relevant information to present to the user. (Redfern, 2:52-3:9, 15:19-41, 16:47-60.)

Rationale and Motivation to Combine (Ground 1 Prior Art and Redfern).

It would have been obvious to combine Redfern’s teachings regarding a meta-search application with Coffman’s teachings regarding conversationally-aware applications. The combination would have predictably resulted in a meta-search application as taught by Redfern with the ability to interface with Coffman’s conversational system, thereby advantageously enabling, for example, a user to use the meta-search application by providing spoken natural language input and with a variety of devices with different input and output capabilities. (Schmandt, ¶203.)

Redfern is an analogous reference in the field of the '209 patent, and would have been reasonably pertinent to problems facing the '209 inventors, including how to interface and control an application using natural language input and how to query multiple information sources using natural language input. (Schmandt, ¶204.)

A POSITA would have been motivated to combine. (Schmandt, ¶205.) Coffman itself describes conversational search functionality as well as integration with websites. (Coffman, 48:25-49:16 (search capability), 58:32-59:3.) Redfern provides an express motivation for a natural-language-based meta-search, explaining that it is more efficient and less burdensome for the user. (Redfern, 1:66-2:8, 2:38-44.) Moreover, Redfern itself already describes the application's ability to receive natural language search data. (Redfern, 4:28-31.) A POSITA would have readily appreciated that adapting and incorporating Redfern's meta-search application for Coffman's conversational system would enhance a user's ability to input natural language search queries and instructions. (Schmandt, ¶205.)

A POSITA would have considered Redfern and Coffman to be combinable using conventional techniques and technologies, with the particular architecture of the conversational system being a matter of design choice. (Schmandt, ¶206 (citing Redfern, 4:28-31, 10:25-38, 4:8-9; Coffman, 12:5-7, 13:1-3, 22:32-23:3, 55:13-19).) A POSITA would have therefore had a reasonable expectation of success, without undue experimentation. (Schmandt, ¶207.)

Turning back to the claim language, with respect to a request that “**includes at least one question and at least one command**,” the ’209 patent states that a question and a command are similar but that a question typically results in a response while a command results in an action, and that an utterance can include “both aspects of a question and a command.” (’209, 5:53-56, 10:60-64, 3:39-45.)

It would have been obvious for the request in the user’s utterance to “**include[] at least one question and at least one command**.” Redfern provides an example of “**at least one question**”—“Where do Monarch butterflies spend the winter?”—which is translated into the request “rank1(Monarch) phrase(butterflies spend) phrase (winter)” to be processed by the meta-search application. (Redfern, 15:42-65.) It would have been obvious for the user’s question to be provided as part of a spoken utterance. (Schmandt, ¶209.)

The “**at least one command**” could have taken various forms. For example, the command could relate to selecting the set of search engines/information sources to query. Redfern explains that the command to specify the set of search engines/information sources to query could be explicit or implicit. (Redfern, 9:22-27.) It would have been obvious for a user to explicitly specify the set of search engines/information sources for a search with an utterance such as “Search Lycos, AltaVista and Excite for where do Monarch butterflies spend the winter?” (Schmandt, ¶210.) As another example, the command for the set of search

engines/information sources for a search could be implicitly specified by the particular keywords in the user's query, consistent with the '209 patent's statement that an utterance can include aspects of both a question and a command. (*Id.*; '209, 3:39-45.) As another example, a well-known command during web browsing or other network-based activities is to stop/cancel a pending operation. It would have been obvious for a user's utterance to include a command to stop/cancel a pending search operation to enter a new one. (Schmandt, ¶210.) These examples include a “**command**” because they instruct the system to take some action—such as select and use particular search engines or stop processing. The ability to enter commands of this nature using speech utterances would have been especially useful in situations and devices with limited input/output capabilities beyond voice as contemplated by Coffman. (*Id.*)

As explained for 1[g], “**the formulated request**” corresponds to an application call/query to a conversationally-aware application (“**the invoked domain agent**”). Therefore, under the combination with Redfern, “**the formulated request**” would correspond to an application call/query to the meta-search application that included search data resulting from processing and formatting the user's natural language query. (Schmandt, ¶211; Redfern, 4:35-8:65 (describing processing and formatting), 15:44-49.)

(a) “**directing...**” (7[a])

With respect to a “**command**” to select (explicitly or implicitly) the search engines/information sources to query, it also would have been obvious for the conversationally-aware meta-search application (“**the invoked domain agent**”) to process the application call/query (“**process the formulated request**”) by “**directing the at least one command**” to “**local or remote processing devices.**” For example, Redfern explains that, to select and use particular search engines, the application consults “a database of search engine capabilities, requirements and addresses (URL’s or other appropriate address information)...to determine the appropriate parameters for each search engine in the selected set of search engines.” (Redfern, 9:34-38.) By consulting such a database in this manner, the application (“**the invoked domain agent**”) “**directs the at least one command**” to the database system (“**to local or remote processing devices**”). (Schmandt, ¶214.)

7[a] would have also been obvious for a “**command**” to cancel a pending search. For example, a POSITA would have understood that canceling a pending search operation would have entailed instructing local and remote processing devices to stop processing related to the search and close network connections. (Schmandt, ¶215.)

(b) “querying...” (7[b])

It would have been obvious for the conversationally-aware meta-search application (“**the invoked domain agent**”) to process the application call/query (“**process the formulated request**”) by “**querying local or network information sources to resolve information requested in the at least one question**” (e.g., by sending search requests to multiple web-based search engines or other information sources). (Schmandt, ¶216; Redfern, 15:42-65, 10:25-33, 2:60-63.)

2. Claim 8

It would have been obvious for claim 7’s “**querying**” step to include the following steps as part of the conversationally-aware meta-search application (“**invoked domain agent**”) processing the formulated request. (Schmandt, ¶217.)

(a) “submitting...” (8[a])

As discussed, Redfern’s meta-search application submits a “**plurality of...queries**” to “**the information sources.**” For example, to resolve the user’s question “Where do Monarch butterflies spend the winter?” the application submits a query of “Monarch+butterflies+spend+winter” to the Lycos search engine, a query (and further instruction to list results having “Monarch” in them first) of “Monarch+AND+(butterflies+OR+spend)+AND+(winter) Monarch ranked first” to the AltaVista search engine, and a query of “Monarch+AND+(butterflies+OR+spend)+AND+(winter)” to the Excite search

engine. (Redfern, 15:42-65; *see also id.*, 25:19-21 (describing AltaVista “selection criteria” and “results ranking criteria”).)

Redfern further discloses that the submitted queries are “**asynchronous**” because the meta-search engine submits each query without waiting for an earlier one to be completed or time-out. (Redfern, 15:63-65, 10:38-44; *see also id.*, Fig. 5 (steps **528** and **532** showing that all queries are dispatched in step **528** and all results received at step **532**); Schmandt, ¶219.) Moreover, it would have been obvious for the queries to be “asynchronous” to more quickly receive and return results to the user by having multiple search engines processing queries substantially simultaneously. (Schmandt, ¶219.) Redfern acknowledges the need for a timely response to the user, describing obtaining results before “a predetermined time limit has been exceeded.” (Redfern, 10:38-44.)

As to “**the plurality of queries including a plurality of duplicate queries submitted to distinct information sources,**” a POSITA would have understood and found it obvious that “**a plurality of duplicate queries**” would be submitted by the meta-search application across the selected set of search engines/information sources. (Schmandt, ¶220.) Nothing in the ’209 patent suggests that queries must be textually verbatim or in an identical format to be considered “**duplicate queries.**” Redfern’s example queries about Monarch butterflies are three “duplicate” queries because they are substantively identical in seeking the same information. (*Id.* (citing

'209, 19:22-24).) But Redfern also discloses and renders obvious textually verbatim **“duplicate queries.”** For example, Redfern identifies an Excite query of **“Monarch+AND+(butterflies+OR+spend)+AND+(winter).”** This is identical to the AltaVista query, **“Monarch+AND+(butterflies+OR+spend)+AND+(winter).”** (*Id.*)

Lastly, the Lycos, AltaVista and Excite search engines (as well as others described by Redfern (Redfern, 9:4-9, 9:66-10:1)) are clearly **“distinct information sources”**—the whole point of Redfern’s meta-search application is to enable a user to conveniently query multiple distinct information sources at once. (Redfern, 2:47-51, 2:37-40.)

(b) “a synchronously receiving...” (8[b])

Redfern describes **“a synchronously [sic] receiving a plurality of results of the queries”**⁶ from the search engines (**“the information sources”**). (Redfern, 10:38-50.) Moreover, for substantially the same reasons as those discussed above for submitting “asynchronous” queries, “asynchronously receiving” the query results would have been obvious. (Schmandt, ¶222.)

⁶ The word “a synchronously” in 8[b] is a typo of “asynchronously.” (Schmandt, ¶222 n.7.)

(c) “scoring....” (8[c])

Redfern describes a process of determining the most relevant search hits to present to the user (i.e., “**determin[ing] one or more best responses to the at least one question**”) that involves “scoring” search hits (“**the asynchronously received results**”) by matching words and phrases from the user’s natural language input against text from each search hit. (Redfern, 10:44-11:10.) The search hit(s) having the highest scoring “segments,” or excerpts, with scores “no less than a preselected level” are selected to present to the user. (Redfern, 13:9-11, 14:41-46, 15:19-31, 34:13-35 (highest scoring segments for question “Where do Monarch butterflies spend the winter?”).)

Redfern’s process, under the combination, “**us[es] the dynamic set of prior probabilities or fuzzy possibilities**” to determine the most relevant search hits because the process would have used the recognized words and phrases provided from the user, which as discussed for claim 1 were determined using the dynamic set of prior probabilities in the trigram language model (“**the dynamic set of prior probabilities or fuzzy possibilities**”). (Schmandt, ¶225.)

3. Claim 9

It would have been obvious for the step of “**determining the one or more best responses to the at least one question**” to include the following steps. (Schmandt, ¶226.)

(a) “determining...” (9[a])

Redfern discloses “**formulat[ing] a response to the question**” by describing a process for identifying and extracting the best “segment,” or portion, of web pages corresponding to the search hits returned by each of the search engines that are responsive to the user’s question. (*See generally* Redfern, 10:45-15:41, Figs. 6-6c.) Redfern discloses that this process involves “**determining one or more tokens**,” i.e., identifying words and phrases in the user’s search data that are used in scoring the search hits.

A “**token**” in information processing is generally understood as an individuated or non-reducible text element in data being parsed. (Schmandt, ¶228 (citing EX1009, p.0035 (“token”)).) Redfern explains that one of the first steps in identifying and extracting the best “segment” is to take the search data corresponding to the user’s question and treat each group in the search data as a separate candidate for scoring. Redfern describes example search data having four separate candidates: (1) “or(phrase(contact), phrase(personnel), phrase(names))”; (2) “phrase(people)”; (3) “rank1(Gravis)”; and (4) “rank1(Logitech).” (Redfern, 10:64-11:10.)

Redfern’s scoring candidates contain “**tokens required to formulate a response to the question**” because they specify individuated words and phrases from the search data that are used to identify a segment and determine whether the segment meets a scoring threshold for being responsive to the user’s question.

(Schmandt, ¶229.) In Redfern’s example above, the tokens would include “contact,” “personnel,” “names,” “people,” “Gravis,” and “Logitech.” In Redfern’s example question “Where do Monarch butterflies spend the winter?,” which is translated into the search data “rank1(Monarch), phrase(butterflies spend), phrase(winter),” the tokens would include “Monarch,” “butterflies spend” (and separately “butterflies” and “spend”) and “winter.” (Redfern, 11:4-10, 15:43-49.)

Redfern further explains that, if a score for a segment is too low, such as below a required minimum preselected level, then it is not provided as a response to the user. (Redfern, 15:19-31.) Therefore, a POSITA would have understood that the tokens were “**required to formulate a response to the question**” because if the token matching failed to reach the minimum level, then no segments (“**response to the question**”) would be provided to the user. (Schmandt, ¶230.)

(b) “extracting...” (9[b])

Redfern explains that the search hits (“**asynchronously received results**”) generally include a URL to the hit’s information source; e.g., in the hit shown below, the URL <<http://www.isit.com/butterfly/articles/sendmon.htm>>:

91% Monarch Population Plummets in Mexico [More Like This]
URL: <http://www.isit.com/butterfly/articles/sendmon.htm>
Summary: Urquhart tagged the first monarchs 59 years ago--and graciously agreed to tag the first paper monarchs for this symbolic migration! All migratory monarchs east of the Rocky Mountains spend the winter in just 9 major sanctuaries in Mexico!

(Redfern, 28:37-42; *see also id.*, 24:38-43 (AltaVista example).)

Redfern “**extract[s] one or more values for the determined tokens from the asynchronously received results**” by having the meta-search application automatically download web pages pointed to by search hit URLs and search through the downloaded web pages to find and record values in the pages matching search data tokens (for example, finding and recording “Monarch” in the web pages). (Schmandt, ¶232; Redfern, 10:48-63 (“...retrieves the complete information sources **54** for each of a preselected maximum number of hits from each search engine **36**.”), 11:31-38 (“[T]he information source is examined to determine each match between its contents and the groups in the score table. For each match, an entry is made in the score table....”).)

Redfern further discloses that “**the one or more values [are] extracted by applying one or more scraping criteria to the asynchronously received results.**” “**Scraping**” in this context generally refers to techniques where a computer program extracts data from human-readable output from another program. (Schmandt, ¶233.) Redfern’s meta-search application extracts the value(s) “**by applying one or more scraping criteria**” by identifying and removing redundant search hits, retrieving a preselected maximum number of hits from each search engine, and then searching each retrieved hit for values matching the specific tokens corresponding to the search data. (*Id.*; Redfern, 10:48-63, 11:31-38.)

(c) “evaluating...” (9[c])

Redfern, under the combination, discloses “**evaluating the extracted values using the dynamic set of prior probabilities or fuzzy possibilities**” because the words and phrases in the retrieved web pages are matched and scored using tokens resulting from the speech recognition process described in Coffman. (Schmandt, ¶234.) As explained, that speech recognition process uses the “dynamic set of prior probabilities” related to, for example, a trigram language model. For example, under the combination, the user’s question “Where do Monarch butterflies spend the winter?” would have been recognized from the user’s speech using prior probabilities of a language model. Words in the user’s question are then used as search data tokens “Monarch,” “butterflies spend” (and separately “butterflies” and “spend”), and “winter.” (Redfern, 11:4-10, 15:43-49.)

Redfern’s process of matching and scoring (“**evaluating the extracted values**”) “**includes resolving ambiguous, incomplete, or conflicting information**” because it identifies the best-matching responses from the search hits and excludes results that are incomplete or ambiguous and score below a preselected level, such as segments that are shorter in length or poorer matches. (Schmandt, ¶235 (citing Redfern, 11:39-47 (exclude results having fewer than three matches), 13:1-14 (scoring calculation weights against segments that are shorter in length, and against

hits with a high number of matches in just a few tokens and few matches for remaining tokens)).)

(d) “selecting....” (9[d])

Redfern’s meta-search application “select[s] **one or more best values for the determined tokens from the evaluated values**” by selecting the best segments including values matching the tokens as a result of the scoring process. (Schmandt, ¶¶236-237; Redfern, 15:19-28 (highest scoring segments “no less than a preselected level” selected for output to user).) For example, for the user’s question “Where do Monarch butterflies spend the winter?,” Redfern identifies the segment shown below as the best response, with highlighting added to indicate values matching tokens:

Ranking: 3.3608

From: dnr.state.il.us

Found with: Alta Vista

While most insects hibernate, the monarch is the only species of butterfly which actually flies to warmer weather (migrates) in winter. Monarchs from Illinois spend their winters in California and Mexico. In the fall, people have reported seeing entire trees covered with thousands of migrating monarchs! However, only about 1 percent of these monarchs actually survive the journey back to Illinois.

(Redfern, 34:15-23; *see also id.*, 24:38-43 (search hit corresponding to best segment), 16:44-45.)

4. Claim 12

It would have been obvious for the step of “**determining the one or more best responses to the at least one question**” to include the following. (Schmandt, ¶238.)

(a) “determining...” (12[a])

As explained, Redfern discloses only providing responses whose scores are no less than a preselected level (Redfern, 15:19-27), and the scoring process and use of a preselected level operates to resolve ambiguous, incomplete or conflicting information by eliminating search hits whose low scores would reflect those hits having ambiguous, incomplete or conflicting information. A POSITA would have therefore understood and found it obvious that a determination of no response would represent a determination that the matching and scoring process (use of “**the dynamic set of prior probabilities or fuzzy possibilities**”) “**failed to resolve the ambiguous, incomplete, or conflicting information.**” (Schmandt, ¶239.)

(b) “resolving....” (12[b])

Under the combination, where the meta-search application failed to provide a response to the user’s question, prompting the user to refine or restate their question and resubmit to identify a search hit meeting the meta-search application’s scoring criteria (“**requesting the user to provide additional information for resolving the ambiguous, incomplete, or conflicting information**”) would have been obvious.

Petition for *Inter Partes* Review of
U.S. Patent No. 7,398,209 B2

For example, Redfern describes “not obtaining acceptable results without multiple and/or recursive search attempts.” (Redfern, 2:9-11.) Coffman describes prompting the user for missing or ambiguous information. (Coffman, 41:16-21.) Applying Coffman’s known technique to improve Redfern’s similar process would have been obvious. (Schmandt, ¶241.)

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 13,952 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. 7,398,209 B2**, including all exhibits (**Nos. 1001-1023**) 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