Trials@uspto.gov
571-272-7822

Paper 11
Date: October 15, 2024

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

DIALECT, LLC,
Patent Owner.

IPR2024-00746
Patent 7,398,209 B2

Before BRIAN J. McNAMARA, HYUN J. JUNG, and JON M. JURGOVAN, *Administrative Patent Judges.*

McNAMARA, *Administrative Patent Judge*.

DECISION
Denying Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

Google LLC. ("Petitioner") filed a petition, Paper 2 ("Petition" or "Pet."), to institute an *inter partes* review ("IPR") of claims 1–9 and 12 (the "challenged claims") of U.S. Patent No. 7,398,209 B2 ("the '209 patent"). 35 U.S.C. § 311. Dialect, LLC ("Patent Owner") filed a Preliminary Response, Paper 8 ("Prelim. Resp."), contending that the Petition should be denied as to all challenged claims. We have jurisdiction under 35 U.S.C. § 314, which provides that an *inter partes* review may not be instituted unless the information presented in the Petition "shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition."

A decision to institute under § 314 may not institute on fewer than all claims challenged in the petition. *SAS Inst., Inc. v. Iancu*, 138 S. Ct. 1348, 1359–60 (2018). In addition, per Board practice, if the Board institutes trial, it will institute "on all of the challenged claims and on all grounds of unpatentability asserted for each claim." *See* 37 C.F.R. § 42.108(a).

Having considered the arguments and the associated evidence presented in the Petition and the Preliminary Response, for the reasons described below, we decline to institute *inter partes* review.

II. REAL PARTIES IN INTEREST

The Petition identifies Google LLC as its real party-in-interest. Pet. 11. Patent Owner identifies Dialect, LLC as its real party-in-interest. Paper 5, 1.

III. RELATED MATTERS

The Petition states that the '209 patent is asserted in the following litigation: *Dialect, LLC v. Google LLC f/k/a Google Inc. et al.*, CA No.: 1:23-cv-00378 (D. Del. Apr. 3, 2023). Pet. 11, *see also* Paper 5, 1.

Petitioner notes that 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. Pet. 11.

IV. THE ’209 PATENT

The ’209 patent relates to the retrieval of online information or processing of commands through a speech interface that allows users to submit natural language speech questions and commands. Ex. 1001, 1:13–17. The ’209 patent uses context, prior information, domain knowledge, and user specific profile data to achieve a natural environment for one or more users making queries or issuing commands in multiple domains. *Id.* at 1:65–2:1. A feature of the ’209 patent is the organization of domain specific behavior and information into “agents.” *Id.* at 2:48–49. The ’209 patent describes agents as “autonomous executables that receive, process and respond to user questions, queries and commands.” *Id.* at 2:50–51.

Figure 1 of the ’209 patent, reproduced below, is an overall diagrammatic view according to one embodiment. Ex. 1001, 8:48–49.

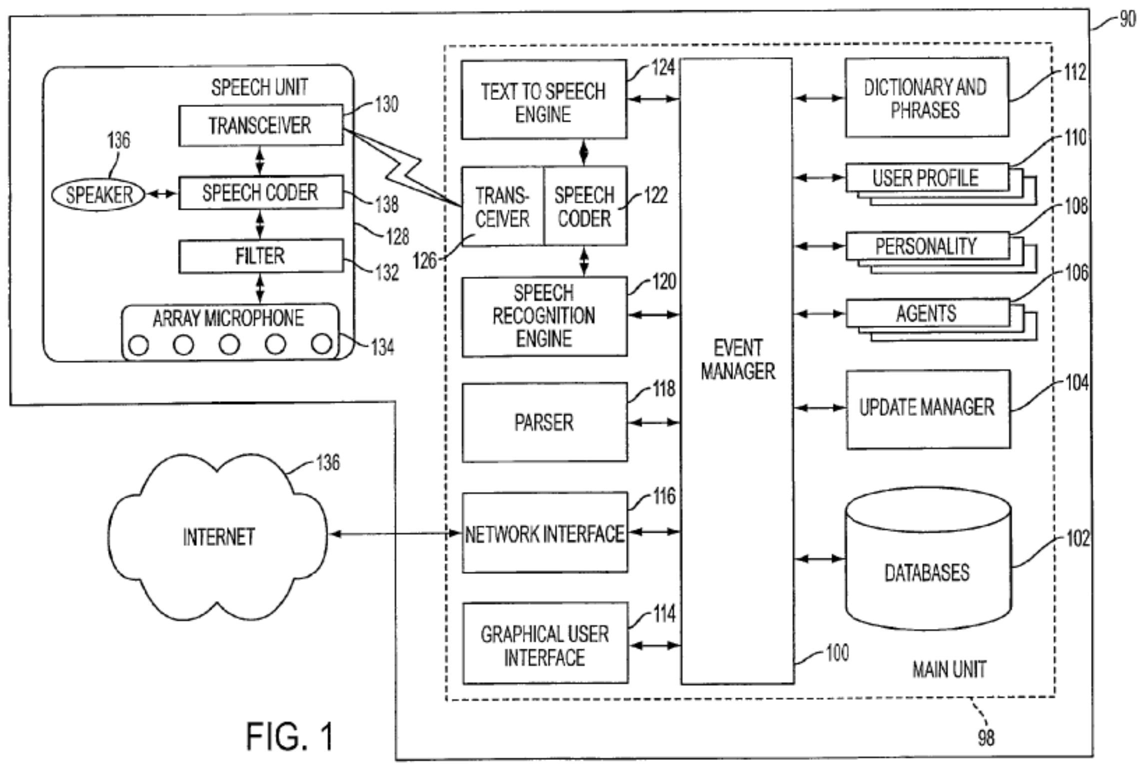


FIG. 1

Ex. 1001, Fig. 1. System 90 includes main unit 98 (shown as the dashed box on the right) and one or more speech units 128 (shown as the solid line box on the left); the speech units can be configured as separate units. *Id.* at 9:66–10:2, 10:21–23. Speech units 128, which provide encoded speech to transceiver 126 in main unit 98 over a bi-directional link, include microphone array 134 that receives utterances, transceiver 130, speech coder 139 that encodes and compresses speecvh, and filter 132. *Id.* at 10:21–39.

Main unit 98 includes generic and domain specific agents 106 that receive and return events to event manager 100. *Id.* at 11:45–46. User profiles 110 store specific data, parameters, session, and history information that determines the behavior of agents 106. *Id.* at 10:12–15.

"Coded speech received at transceiver 126 of main unit 98 is passed to speech encoder 122 for decoding and decompression." *Id.* at 10:49–51.

Decoded speech is processed by speech recognition module 120 using data in dictionary and phrases module 112 and data returned from agents 106 to allow parser 118 to process recognized words and phrases into commands and questions; based on keywords and structures of the questions and commands, parser 118 invoke the required agents. *Id.* at 10:49–64, 11:45–50; Fig. 2 (showing domain agents 156). Once words are recognized by speech recognition engine 120, tokens and user identification are passed to parser 118. *Id.* at 17:41–43. Parser 118 uses a scoring system to determine the most likely context or domain for a user's question or command, e.g., parser 118 determines a context for the utterance by applying prior probabilities or fuzzy probabilities to the keyword and determines the domain agent 156 to be invoked. *Id.* at 17:41–18:10.

Agents 106 create and present a response string to text-to-speech engine 124; text-to-speech engine 124 creates the responsive utterances to be encoded and compressed by speech coder 122 and transmitted by transceiver 126 to speech unit 128, where they are decompressed and output by speaker 136. Ex. 1001, 10:65–11:10. Graphical user interface 114 can substitute or complement the speech interface. *Id.* at 11:11–20.

V. ILLUSTRATIVE CLAIM

Claim 1, reproduced below with the paragraph designations used in the Petition, is representative of the subject matter of the '209 patent,

> 1[pre]. 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.

VI. ASSERTED GROUNDS

Petitioner asserts that claims 1–9 and 12 would have been unpatentable on the following grounds:

Claim(s) Challenged	35 U.S.C. §	Reference(s)
1, 2, 4, 5	103[1]	Coffman,[2] Kanevsky,[3]

[1] The Leahy-Smith America Invents Act ("AIA") included revisions to 35 U.S.C. §§ 102, 103 that became effective on March 16, 2013. Because the '209 patent issued from an application filed before March 16, 2013, we apply the pre-AIA version of the statutory basis for unpatentability.

[2] International Patent Application Publication No. WO 00/20962 to Daniel Coffman, Stephane H. Maes et al. (filed October 1, 1999, published April 13, 2000) (Ex. 1003).

[3] U.S. Patent No. 5,897,616 to Dimitri Kanevsky and Stephane Herman Maes (filed June 11, 1997, issued April 27, 1999) (Ex. 1004).

Claim(s) Challenged	35 U.S.C. §	Reference(s)
		Kneser,[4] Zadrozny[5]
2	103	Coffman, Kanevsky, Kneser, Zadrozny, Maes[6]
3	103	Coffman, Kanevsky, Kneser, Zadrozny, Maes '101,[7] Franceschi[8]
6	103	Coffman, Kanevsky, Kneser, Zadrozny, Chai[9]
7–9, 12	103	Coffman, Kanevsky, Kneser, Zadrozny, Redfern[10]

Petitioner also cites to the Declaration of Dr. Christopher M. Schmandt In Support of the Petition (Ex. 1002, Schmandt Decl.)

VII. LEVEL OF ORDINARY SKILL IN THE ART

Petitioner states that a person of ordinary skill in the art ("POSITA" or "ordinarily skilled artisan") "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

[4] U.S. Patent No. 6,157,912 to Reinhard Kneser et al. (filed March 2, 1998, issued December 5, 2000) (Ex. 1005).
[5] Wlodek Zadrozny et al., *Natural Language Understanding with a Grammar of Constructions*, Proc. 15th Int'l Conf. on Computational Linguistics (Vol. 2) (1994) (Ex. 1006).
[6] International Patent Application Publication No. WO 00/21075 to Stephane H. Maes et al. (filed October 1, 1999, published April 13, 2000) (Ex. 1007).
[7] U.S. Patent No. 6,073,101 to Stephane Herman Maes (filed January 28, 1997, issued June 6, 2000) (Ex. 1010).
[8] U.S. Patent No. 6,321,196 B1 to Carlos Antonio Franceschi (filed July 2, 1999, issued November 20, 2001) (Ex. 1011).
[9] U.S. Patent No. 6,829,603 B1 to Joyce Yue Chai, Wlodek Zadrozny et al. (filed February 2, 2000, issued December 7, 2004) (Ex. 1012)
[10] U.S. Patent No. 6,078,914 to Darren M. Redfern (filed December 9, 1996, issued June 20, 2000) (Ex. 1013).

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)." Pet. 15 (citing Ex. 1002, Schmandt Decl. ¶¶ 11–15). Patent Owner does not explicitly dispute Petitioner's characterization of a person of ordinary skill in the art.

The level of ordinary skill in the art usually is evidenced by the references themselves. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001); *In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995); *In re Oelrich*, 579 F.2d 86, 91 (CCPA 1978). As Petitioner's description of the level of ordinary skill appears consistent with the technology of the '209 patent and the references, we apply Petitioner's characterization of an ordinarily skilled artisan for purposes of this Decision.

VIII. CLAIM CONSTRUCTION

We interpret claim terms using "the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b)." 37 C.F.R. § 42.100(b) (2019). In this context, claim terms "are generally given their ordinary and customary meaning" as understood by a person of ordinary skill in the art in question at the time of the invention. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (citations omitted) (en banc). "In determining the meaning of the disputed claim limitation, we look principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in evidence." *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1014 (Fed. Cir. 2006) (citing *Phillips*, 415 F.3d at 1312–17). Extrinsic evidence is "less significant than the intrinsic record in determining 'the legally operative meaning of claim language.'" *Phillips*, 415 F.3d at 1317 (citations omitted).

Any special definition for a claim term must be set forth in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). "The Board is required to construe 'only those terms . . . that are in controversy, and only to the extent necessary to resolve the controversy.'" *Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019) (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)).

Neither party has proposed a special definition or explicit construction of any claim term for purpose of this Decision. *See* Pet. 17 ("[e]xpress claim construction is unnecessary"); Prelim. Resp. 13 ("the Board need not engage in any pre-institution claim construction"). For purposes of this Decision, we understand Petitioner to propose that the claims be given their ordinary and customary meaning as understood by a person of ordinary skill under *Phillips*. 37 C.F.R. 42.104(b)(3).

Although Patent Owner does not propose any specific claim constructions for purposes of this Decision, Patent Owner's Preliminary Response includes a list of terms from other apparently related patents construed by the district court in *Dialect, LLC v. Amazon,com Inc.*, Case 1:23-cv-00518, Doc. No. 213 (filed Apr. 29, 2024). Prelim. Resp. 11–13 (citing Ex. 2007). Patent Owner's Preliminary Response also includes a footnote concerning the construction of "domain agent." *Id.* at 12 n.2.

As discussed further in Section IX herein, some of the issues before us turn on whether the references disclose the claimed "domain agent." The '209 patent Specification states:

> According to another aspect of the invention domain specific behavior and information is organized into agents. Agents are autonomous executables that receive, process and respond to user questions, queries and commands. The agents provide

> complete, convenient and re-distributable packages or modules of functionality, typically for a specific domain or application. Agents can be complete packages of executable code, scripts, links to information, and other forms of communication data, required to provide a specific package of functionality, usually in a specific domain. In other words, an agent includes everything that is needed to extend the functionality of the invention to a new domain.

Ex. 1001, 2:48–59; *see also id.* at 11:21–11–34 (stating "domain specific behavior and information may be organized into domain agents").

Although for purposes of the Decision, we do not adopt any specific claim constructions and apply the ordinary and customary meaning to the claim terms, we address the scope of "domain agent" further in our analysis of the challenges to the patentability of the claims.

IX. ANALYSIS

A. Introduction

"In an [inter partes review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable." *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring inter partes review petitions to identify "with particularity . . . the evidence that supports the grounds for the challenge to each claim")). This burden of persuasion never shifts to Patent Owner. *See Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (discussing the burden of proof in inter partes review).

The question of obviousness is resolved on the basis of underlying factual determinations including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) objective evidence of nonobviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). Additionally, the obviousness inquiry typically requires an analysis of "whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006) (requiring "articulated reasoning with some rational underpinning to support the legal conclusion of obviousness")); *see In re Warsaw Orthopedic, Inc.*, 832 F.3d 1327, 1333 (Fed. Cir. 2016) (citing *DyStar Textilfarben GmbH & Co. Deutschland KG v. C. H. Patrick Co.*, 464 F.3d 1356, 1360 (Fed. Cir. 2006)).

An obviousness analysis "need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ." *KSR*, 550 U.S. at 418; *accord In re Translogic Tech., Inc.*, 504 F.3d 1249, 1259 (Fed. Cir. 2007). Petitioner cannot satisfy its burden of proving obviousness by employing "mere conclusory statements." *In re Magnum Oil Tools Int'l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016). Instead, Petitioner must articulate a reason why a person of ordinary skill in the art would have combined the prior art references. *In re NuVasive*, 842 F.3d 1376, 1382 (Fed. Cir. 2016).

A reason to combine or modify the prior art may be found explicitly or implicitly in market forces; design incentives; the "'interrelated teachings of multiple patents'"; "'any need or problem known in the field of endeavor at the time of invention and addressed by the patent'"; and the background knowledge, creativity, and common sense of the person of ordinary skill. *Perfect Web Techs., Inc. v. InfoUSA, Inc.*, 587 F.3d 1324, 1328–29 (Fed. Cir. 2009) (quoting *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 418–21 (2007)).

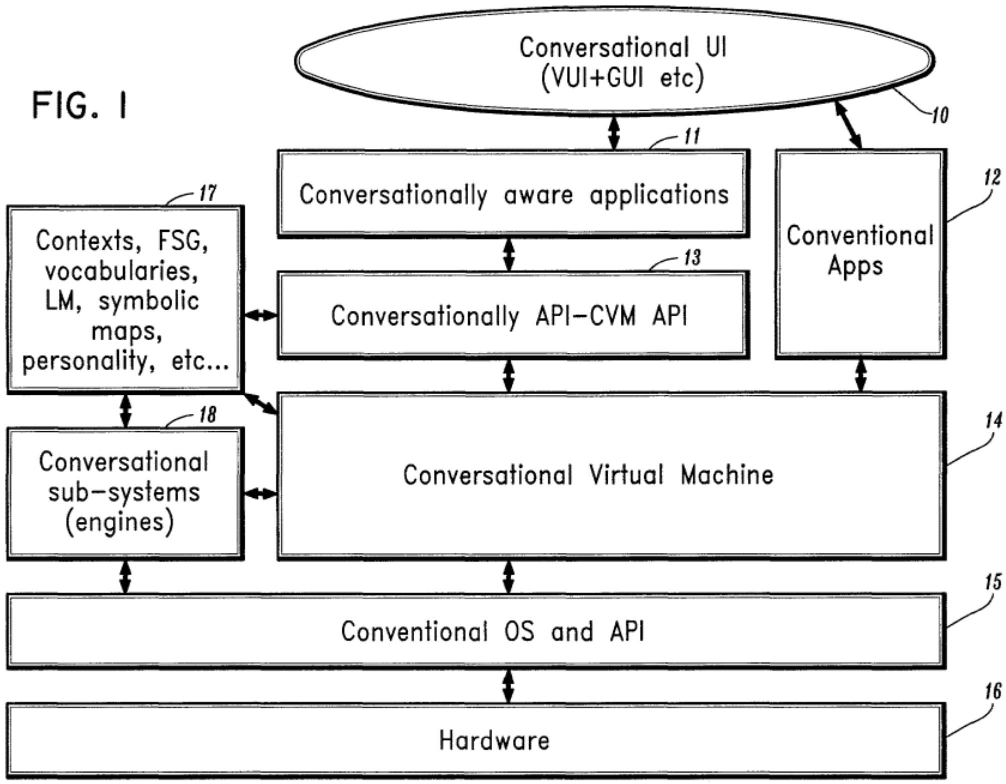
In determining whether a claim is obvious in light of the prior art, when in evidence, we consider any relevant objective evidence of non-obviousness. *See Graham*, 383 U.S. at 17. Notwithstanding what the teachings of the prior art would have suggested to one of ordinary skill in the art at the time of the invention, the totality of the evidence submitted, including objective evidence of non-obviousness, may lead to a conclusion that the challenged claims would not have been obvious to one of ordinary skill. *In re Piasecki*, 745 F.2d 1468, 1471–72 (Fed. Cir. 1984). At this stage of the proceeding Patent Owner does not present evidence of such objective considerations

We analyze the asserted grounds of unpatentability in accordance with these principles to determine whether Petitioner has met its burden to establish a reasonable likelihood of success at trial.

B. Petitioner's Contention That Claims 1–2 and 4–5 Would Have Been Obvious Over Coffman In View of Kanevsky, Kneser, and Zadrozny

1. *Coffman (Ex. 1003)*

As discussed further herein, a primary issue before us is Petitioner's reliance on the teachings of Coffman. Coffman aims to provide conversational computing through a universal conversational user interface (CUI) applicable to speech I/O interfaces, as well as other conversational classes, e.g., keyboards and gesture inputs. Ex. 1003, 1:12–17, 4:4–13, 14:5–9, 13:29–14:6. Figure 1 of Coffman, reproduced below, is a block diagram of a conversational computing system according to one embodiment. *Id.* at 9:27–28.

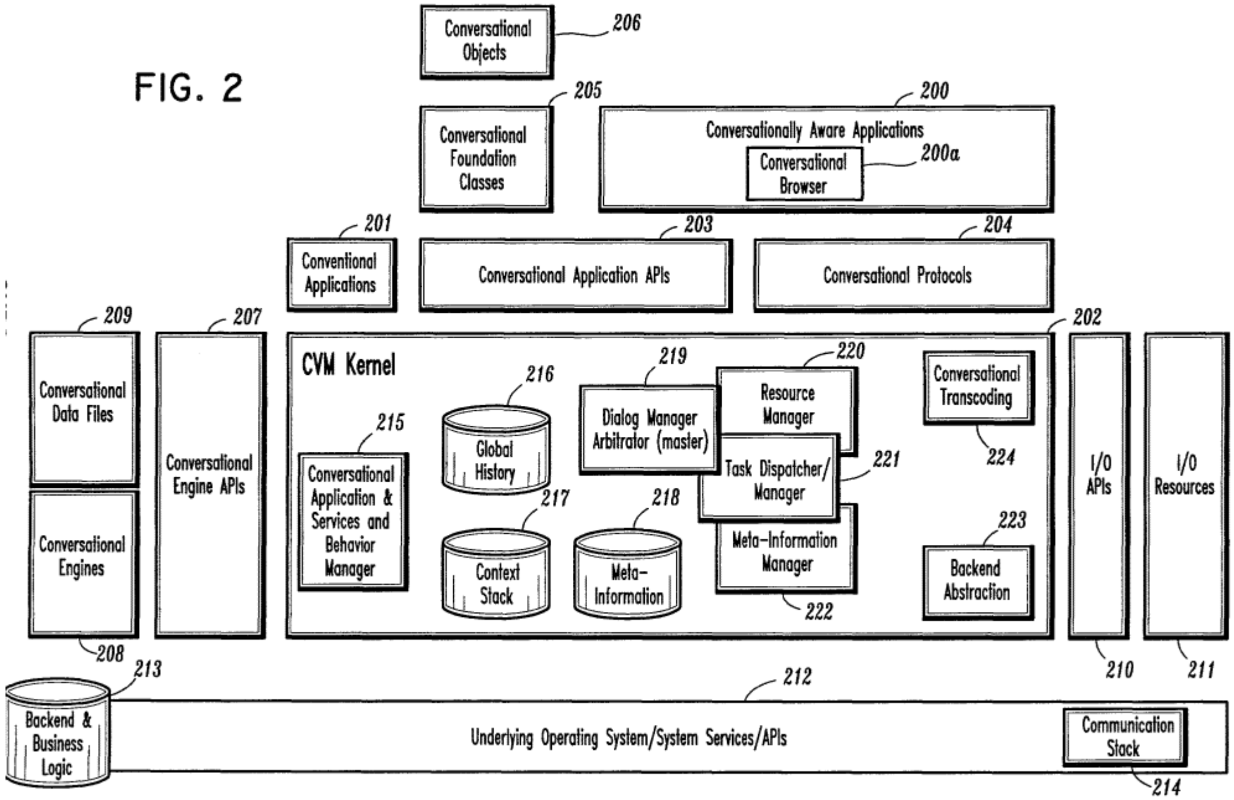


Ex. 1003, Fig. 1. As shown in Figure 1, Coffman discloses conversational virtual machine (CVM) 14 as a kernel that runs on top of conventional operating system (OS) 16 to provide backward compatibility for conventional platforms by managing dialog and context, conversational sub-systems (engines) and resources, and conversational protocols/communications across platforms and devices with different capabilities. *Id.* at 1:12–17, 4:17–26, 13:1–3. "The CVM kernel is the core layer that controls the dialog across applications and devices on the basis of their registered conversational capabilities and requirements." *Id.* at 4:27–28. "[C]onventional applications 12 are managed by CVM kernel layer 14 which is responsible for accessing, via the OS APIs, GUI menus and commands of the conventional applications as well as the underlying OS commands." *Id.* at 13:3–5. Conversational sub-systems 18, e.g., speech recognition, natural language understanding (NLU) and natural dialog generation (NLG) engines, are accessed through the CVM. *Id.* at 5:1–7.

Coffman also discloses conversationally aware applications 11 "that are specifically programmed for operating with a CVM core layer (or kernel) 14 via conversation applications APIs 13, . . . [based on] their registered conversational capabilities and requirements." *Id.* at 12:28–32. CVM kernel 14 controls the dialog across applications and devices based on their registered conversational capabilities and provides a unified conversational user interface (CUI). *Id.* A conversationally aware application 11 provides its data file to the CVM for use by the conversational subsystems 18 by "registering all its conversational capabilities and requirements," including active vocabulary, symbolic mapping (if NLU is supported or in a context state list), associated meta-information to allow categorization of the output, conversational I/O information, and CVM capabilities/state (master, slave, or partner CVM). *Id.* at 57:22–58:2.

Figure 2 of Coffman, reproduced below, illustrates abstract layers of an embodiment of Coffman's conversational computing system. Ex. 1003, 9:29–30.

FIG. 2



Ex. 1003, Fig. 2. Conversationally aware applications 200 interact with CVM kernel layer 202 via conversationally aware application API layer 203. *Id.* at 13:16–19. CVM kernel layer (or CVM controller) 202 includes a meta-information manager 222, resource manager 220, context stack 217, global history 216, dialog manager 219, and task dispatcher 221, for managing dialog and selecting the active dialog, context, and application. *Id.* at 5:12–15. CVM kernel layer 202 tracks resources, transmits inputs to appropriate conversational sub-system, arbitrates between devices, states, and applications, and coordinates output generation and prioritizations. *Id.* at 5:20–24. Within kernel 202, context stack 217 "accumulates the context (full query arguments, and state/mode, i.e., query arguments already introduced, any I/O event and event produced by the application) of each active process with an activated discourse along with any data needed to input understanding (e.g., active FSG, topic, vocabulary or possible queries for a speech input)." *Id.* at 5:15–18.

Conversational application API layer 203 comprises a plurality of conversational foundation classes 205 (or fundamental dialog components), i.e., elementary components or conversational gestures that characterize any dialog independently of the modality, and comprise CUI building blocks and conventional platform libraries, dialog modules, components, and scripts and beans. *Id.* at 13:29–14:6. Noting that data files for the conversational foundation classes are present on the CVM and that data files of objects can be expanded and loaded to act as simultaneous data managers, Coffman lists a number of conversational foundation classes at pages 15–18 of the reference.

Coffman's conversational API layer 203 encompasses conversational programming languages and scripts to provide universal input and output. *Id.* at 18:23–25. Conversational engines 208 and conversational data files 209, accessed by CVM 202 via conversant engine APIs 207, convert each input into binary or ASCII input that can be processed directly by the programming language as built-in objects. *Id.* at 18:27–30. "The input procedure provides a set of multi-mode input streams, each transcribed into an ASCII command, or a list of n-uples." *Id.* at 38:26–27. "Each input entity (command, NLU query field or argument unit (isolated letter, word etc.) are associated with time-marks and appended accordingly to a compounded input stream." *Id.* at 38:27–29.

Figure 4 of Coffman, reproduced below illustrates an embodiment of a conversational system that combines conventional subsystems and conversational subsystems executed and managed by the CVM. Ex. 1003, 29822–24.

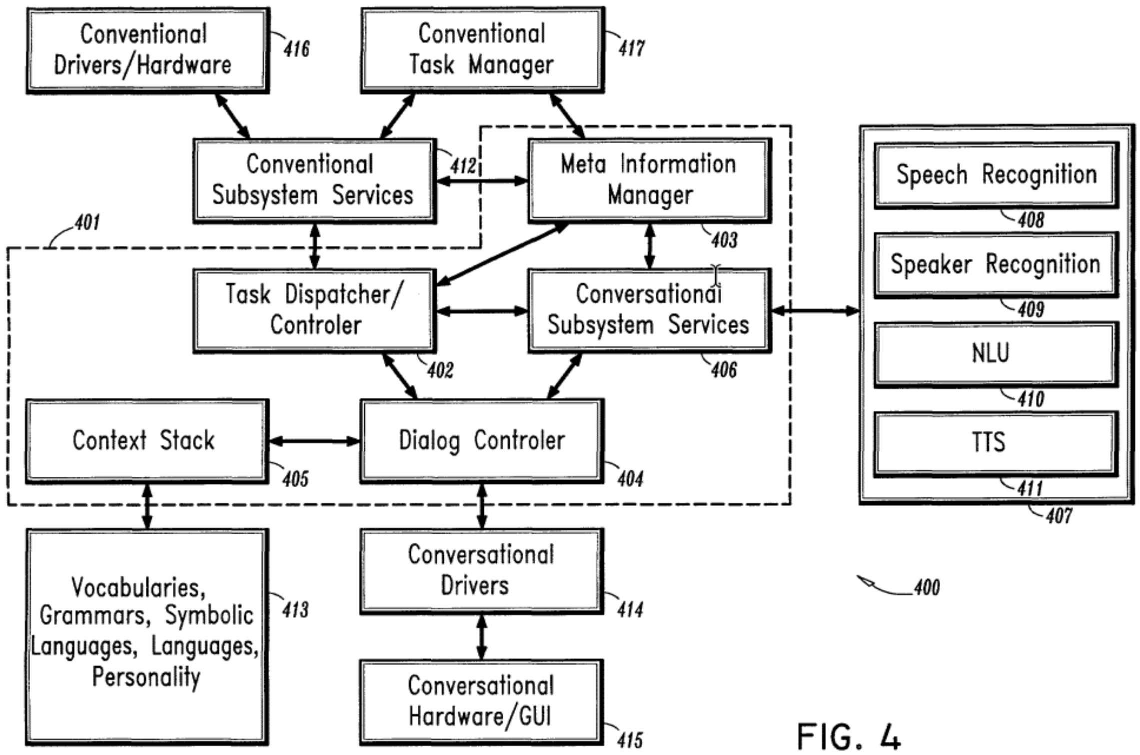


FIG. 4

Ex. 1003, Fig. 4. CVM 401 includes task dispatcher/controller 402, meta information manager 403, dialog controller (or dialog manager) 404, context stack 405 and conversational subsystems services manager 406. *Id.* at 28:22–24. CVM 401 converts conversational and conventional input streams into multiple actions and produces sorted output to a user through conversational and/or conventional output means. *Id.* at 28:27–31. Conversational resource systems (engines) 407, which communicate with CVM 401, include speech recognition system 408, speaker recognition system 409, natural language understanding (NLU) and natural language parsing system 410, and text-to-speech (TTS) system 411. *Id.* at 29:1–4. Conversational subsystems 407 (designated 18 in Figure 1) may be accessed through API calls to CVM 401 via conversational subsystems services manager 406, such that CVM 401 locates requested conversational subsystem 407, drives its execution, and returns the results. *Id.* at 29:5–9.

Conventional subsystem services manager 412 (shown as not within CVM 401) manages the services and UI offered by an underlying operating system. *Id.* at 29:14–17.

CVM 401 includes context stack 405. Ex. 1003, Fig. 4. In Coffman "context" refers to the state of each discourse. *Id.* at 29:28–29. Context stack 405 accumulates context (i.e., a full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., a conversational interaction associated with a given task/process/thread) along with data files 413 (or identifiers for such conversational arguments) for different engines (conversational resource systems 407) that may be needed for understanding input. *Id.* at 29:20–27. Dialog controller 404 manages context stack 405 by creating a new stack entry for each new task, process or thread associated with a discourse. *See id.* at 30:4–25. For reasons discussed below, it is important to note that Coffman's context stack is associated with a discourse rather than an application.

Context stack 405 is associated with the machine state stack so that for any new input from a user, dialog controller 404 may traverse context stack 405 until the input context can be appropriately established. Ex. 1003, 30:22–24. This essentially amounts to finding and selecting the active discourse between the user and machine among the last and past discourses. *Id.* at 30:24–25.

Input/output processing is organized by CVM 401, where system calls can spawn multiple actions, e.g., executing a task, listening for new input, and producing output. *Id.* at 40:11–12. On the basis of context stack 405, task dispatch/controller 402 decides the different statistical parsers that must operate on a query for dialog controller 404 to operate to identify the active context and complete the query. *Id.* at 40:13–15. Prioritized actions can include executing completed queries and updating context stack 405, providing user feedback for incomplete or ambiguous queries, allowing new input to be decoded and run down context stack 405, and returning output or running processes. *Id.* at 40:15–19. CVM 401 produces output to files/streams resources, display, and audio using the context stack and task controller. *Id.* at 39:3–8. Output streams can be formatted according to the needs of the application or user. *Id.* at 18:32–33.

2. *Kanevsky (Ex. 1004)*

As discussed further herein, Petitioner cites Kanevesky as disclosing a speech recognition model may be implemented using a trigram language having probabilities associated with the model. *See* Pet. 21–22 (citing Ex. 1004, 13:48–61, 37:10–13).

3. *Kneser (Ex. 1005)*

As discussed further herein, Petitioner cites Kneser as disclosing the advantages of adaptive n-gram language models. *See* Pet. 26 (citing Ex. 1005, 1:22–29)

4. *Zadrozny (Ex. 1006)*

Petitioner cites Zadrozny as an example of a semantic analyzer illustrating a technique in which a grammar of constructions analyzes a sent utterance and formulates it as an action and series of parameters. Pet. 38–39 (citing Ex. 1006, 19–22).

5. *Claim 1*

The Petition address all the limitations of claim 1. As Patent Owner focuses its response to Petitioner's arguments concerning limitations 1[e] and 1[g], we focus our analysis on those limitations and summarize Petitioner's arguments concerning the remaining limitations of claim 1.

a) Limitations 1[a]–1[d]

As designated by Petitioner, limitation 1[a] recites "receiving the user generated natural language speech utterance, the received user utterance containing at least one request." Pet. 7. Petitioner cites Figure 6 of Coffman as illustrating a system that receives a user's spoken natural language input. *Id.* at 18–20 (citing, e.g., Ex. 1003, Fig. 6, 53:8–10, 37:14–22, 6:32–7:3; Ex. 1002, Schmandt Decl. ¶¶ 88–90).

As designated by Petitioner, limitation 1[b] recites "maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance." Pet. 7. Petitioner notes that Coffman explains that its conversationally aware application, when registering its capabilities and characteristics with the conversational system, provides language models to control the application. *Id.* at 21. Petitioner acknowledges that, although Coffman discloses the use of natural language understanding, Coffman does not provide specific detail about the well-known concepts of maintaining a set of prior probabilities or fuzzy possibilities. *Id.* Petitioner further cites Kanevsky as explaining speech recognition may be implemented using a trigram language model that uses probabilities of the occurrences of sequences of three words. *Id.* at 21–22. Petitioner further contends that an ordinarily skilled artisan would have had reason to combine the teachings of Coffman and Kanevsky, as they are in the same field and are reasonably pertinent to the problem of implementing a system that interacts with a computer using speech and techniques for performing speech recognition. *Id.* 22–23 (citing Ex. 1002, Schmandt Decl. ¶ 97). Petitioner further argues that a "dynamic set of prior probabilities" is disclosed both by Coffman and Kanevsky. *Id.* at 24–26. Noting that Coffman discloses that, over time, the applications registered with the conversational system can change, Petitioner argues that in combination Coffman and Kanevsky teach that, e.g., the set of trigrams and the associated probabilities used by the CVM changes as applications register with the conversational system. *Id.*

Petitioner also states that adaptive n-gram language models that use language model probabilities and their advantages were well-known as taught by Kneser. Pet. 26 (citing Ex. 1005, 1:13–37; Ex. 1002, Schmandt Decl. ¶ 103). Petitioner cites Kneser as providing express motivation to incorporate such well known natural language techniques into Coffman's conversational system to improve speech recognition reliability and performance. *Id.* at 27 (citing Ex. 1002, Schmandt Decl. ¶¶ 105–107).

As designated by Petitioner, limitation 1[c[recites "recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables." Pet. 7. Expanding on its discussion of trigrams under limitation 1[b], Petitioner adds that trigrams are recognized using information in one or more dictionary and phrase tables and argues that an ordinarily skilled artisan would have understood and found it obvious to organize the n-gram language model as a table. *Id.* at 29 (citing Ex. 1002, Schmandt Decl. ¶ 110). Petitioner further argues that a person of ordinary skill would have understood and found it obvious that the words comprising a language model in Coffman would have been based on an active vocabulary alone or in combination with baseforms that provide pronunciation rules, organized as a table. *Id.* at 29–30 (citing Ex.1002, Schmandt Decl. ¶ ¶ 110–112).

As designated by Petitioner, limitation 1[d] recites "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." Pet. 7. Petitioner cites Coffman's conversational system as including "natural language understanding and natural language parsing system 410" and Kanevsky as explaining that natural language understanding can be implemented using "a combination of classical parsing capabilities with [keywords] spotting and speech recognition." *Id.* at 31 (citing Ex. 1003, 29:1–6, 19:16–19, Figs. 4, 6; Ex. 1004, 14:1–7; Ex. 1002, Schmandt Decl. ¶ 114). Petitioner cites Coffman as teaching that parsing the recognized words and phrases is a part of natural language understanding to determining the meaning of an utterance. *Id.* Petitioner cites Coffman as disclosing a "context stack" where the context, including the stated history of each active process, such as an application process is tracked at the context stack is searched to determine a context that matches the user's request based on a natural language understanding process. *Id.* at 32 (citing Ex. 1003, 29:18–30:3, 30:4 –10). According to Petitioner, Coffman discloses and renders obvious that the context for the request is determined based on one or more keywords contained in the recognized word or phrases because the context stack is searched to identify the context that fits the utterance at the level of the natural language understanding symbolic language; i.e., in Coffman recognized speech is run down the context stack to extract the active query and complete it and active grammars, vocabularies and symbolic language map the actual query. *Id.* at 32 (citing Ex. 1003, 42:1–5, 41:4–5, 30:20–22; Ex. 1002, Schmandt Decl. ¶ 118).

b) Limitation 1[e]

As designated by Petitioner, limitation 1[e] recites "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." Pet. 7. As to limitation 1[e], Petitioner's arguments focus on conversationally aware applications disclosed in Coffman. *Id.* at 33–37.

Petitioner notes that Coffman's conversationally aware application receives, processes, and responds to user questions, queries, and commands. Pet. 33 (citing Ex. 1003, 13:8–13 (queries transformed into application calls), 37:2–4 (task executes query). 40:9–19 (execution of completed queries and outputs of executed or running processes). Petitioner cites the executable software of Coffman's conversationally aware application (and its associated data, e.g., vocabulary, grammar, language models, and NLU symbolic mapping) as disclosing or suggesting the claimed domain agent because a conversationally aware application provides functionality for a particular area or realm, such as a particular subject matter or domain (e-mail or spreadsheet) or device (PDA or smartphone). *Id.* at 33–34 (citing Ex. 1003, 16:35–17:7 (listing application specific classes, e.g., manage calendar, manage e-mails, manage address book); 56:22–25 (conversational PDA and smart phone); Ex. 1002, Schmandt Decl. ¶¶ 120–121).

Petitioner further argues that, like the claimed domain agent, Coffman's conversationally aware application is an autonomous executable because without user involvement it registers its capabilities with a conversational system to receive, process, and respond to requests automatically. Pet. 35 (citing Ex. 1003, 7:12–14, 23:16–20, 25:18–24 (automatic discovery of local and network conversationally aware systems), 37:2–4, 42:1–7 (application selections and request carried out by processing of the conversational system); Ex. 1002, Schmandt Decl. ¶ 122). Petitioner further argues that Maes, incorporated into Coffman by reference, confirms the autonomous nature of Coffman's conversationally aware applications. Pet. 35 (citing e.g., Ex. 1007, 3:4–10 (describing techniques for conversationally aware applications to automatically communicate and coordinate with each other); Ex. 1002, Schmandt Decl. ¶ 123).

Petitioner also contends that Coffman's conversationally aware application discloses a domain agent selected based on a determined meaning because activation of an application for processing a user's query is based on the application's association with the determined context. Pet. 36 (citing Ex. 1003, 42:1–7 as disclosing that as soon as a context is found that fits an utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active).

Patent Owner contends that Coffman's conversationally aware application does not teach the claimed domain agent. Prelim. Resp. 27. According to Patent Owner, Petitioner incorrectly characterizes Coffman as disclosing a CVM receives a user request via a conversationally aware application, searches a context stack for the active context, and selects the conversationally aware application to carry out the user's request. *Id.* at 2. Patent Owner contends that "[a] correct understanding of Coffman would recognize that a user initiates interaction with a conversationally aware application in Coffman, and that a conversationally aware application is neither selected nor later invoked based on the determined meaning of an utterance." *Id.* at 35.

Patent Owner argues that Coffman's conversationally aware application is merely a gateway to use the system and only provides an interface (UI) for a user to provide speech-based content for a conversationally aware application to interact with users. Prelim. Resp. 28. According to Patent Owner, Coffman's conversationally aware application is not comparable to the claimed domain agent because the conversationally aware application merely accesses tools provided by Coffman's CVM via an API, but the CVM does not select any conversationally aware application based on determined context to carry out a process or generate a response. *Id.* at 28–29. Patent Owner argues that, in Coffman, the user selects the conversationally aware application and that, in response to a particular conversationally aware application making an API call containing a user's utterance, Coffman's CVM activates a processing thread; then the processing thread accesses a context stack specific to the conversationally aware application, identifies the active discourse from that stack, and carries out a process associated with the active discourse's context. *Id.* at 2, 29.

Patent Owner points to the Petition's acknowledgement that Coffman searches for an active context in a stack of contexts by comparing new queries or arguments to the stack until an acceptable match is obtained. Prelim. Resp. 30 (citing Pet. 36, referencing Ex. 1003, 42:1–7). Patent Owner argues that such determination does not lead to the selection of any conversationally aware application as a domain agent because Coffman's search for an active context follows the user's selection of a conversationally aware application and the identification of that application's corresponding context stack. *Id.* at 30. According to Patent Owner, Petitioner conflates Coffman's process described with respect to Coffman's context stack with Coffman's conversationally aware application. *Id.*

Patent Owner's response to Petitioner's contention that the executable software of Coffman's conversationally aware application and its associated data disclose "selecting at least one domain agent based on the determined meaning," as claimed, is persuasive. Referring to Figure 1, Coffman describes "conversationally aware application 11 (i.e., applications that 'speak conversational protocols')" as "applications that are specifically programmed for operating with a CVM core layer (or kernel) 14 via conversational application APIs 13." Ex. 1003, 12:26–30. Coffman further discloses that "CVM kernel 14 controls the dialog across applications and devices on the basis of their registered conversational capabilities and requirements and provides a unified conversational interface." *Id.* at 12:30–32. Conversational sub-systems 18 (not conversationally aware applications) are responsible for converting voice requests into queries and outputs into spoken messages. *Id.* at 13:8–11. Although Coffman discloses that "[c]onversational application API 13 conveys all the information for the CVM 14 to transform queries into application calls and conversely output into speech," (*id.* at 13:11–13), Petitioner has not demonstrated Coffman discloses a conversationally aware application includes executable software for selecting a domain agent.

In Coffman, task dispatcher/controller 402 associates each task or device with a conversational engine and with conversational arguments. Ex. 1003, 40:20–21. Coffman discloses that, based on the modality (e.g., pointer keyboard, file, speech) task dispatcher 402 in the CVM redirectes the stream to the appropriate conversational subsystem (or conventional subsystem) with speech inputs being transcribed and understood. *Id.* at 41:2–4. The outputs of those subsystems are run down context stack 405 in the CVM to extract a query and complete it. *Id.* at 41:4–5. Context stack 405 is traversed until the input context is established; this means finding and selecting the active discourse between the user and the machine from among the last and past discourses. *Id.* at 41:10–13. The results of complete and unambiguous commands are retuned to conversation engine 407, which includes speech and speaker recognition, NLU, and TTS services. *Id.* at 41:24–25.

The subject matter of Coffman cited by Petitioner states that a new query is compared by a dialog engine (within the CVM) going down the context stack (within the CVM) and, if a context is found that fits the utterance, activating the context and the corresponding process. Pet. 36 (citing Ex. 1003, 42:1–7). Petitioner does not explain how Coffman's disclosure of activating a process after traversing context stack 405 in the CVM teaches a conversationally aware application executing software that selects a domain agent. Coffman's disclosure that "[e]ach action, completed query and conversational argument of an active task as well as each returned value/result on stored on the context stack 405" (*id.* at 40:27–29) suggests that traversing the context stack concerns determining the context based on applications that have already been selected. *See also id.* at 41:8–13 (disclosing that traversing the context stack "essentially amounts to finding and selecting the active discourse between the user and machine among the last and past discourses, possible going back into history."). Petitioner does not explain how selection of the active discourse teaches selecting a domain agent.

Referencing Figure 2, Coffman describes foundation classes 205 as CUI building block and conversation platform libraries, dialog modules, and dialog scripts. Ex. 1003, 14:4–6. Noting that data files for the foundation classes are present on the CVM (loadable for embedded platform), Coffman discloses high-level application specific foundation classes, e.g., classes that manage a bank account, manage a portfolio, request travel reservations, manage e-mail, manage a calendar, or manage an address book. *Id.* at 13:9–11, 16:35–177. Coffman's traversal of a context stack merely concerns which process to execute next; as the foundational data classes are present on the CVM, the selection of which process to execute next occurs after the application has been selected. Patent Owner notes that Coffman associates a process with tasks and threads performed at the CVM. Prelim. Resp. 31 (citing Ex. 1003, 42:16–17).

Patent Owner points out that Coffman's search for an active context follows the user's selection of, and interaction with, a conversationally aware application and identification of the application in the application's corresponding context stack. Prelim. Resp. at 30–31 (citing Ex. 1003, 41:1–7). In Coffman, applications can make calls to the CVM via APIs. Ex. 1003, 32:12–13. Based on its traversal of context stack 405, Coffman's "CVM 401 decides which application is active and how the context is to be modified (. . . with the dialog manager and conversational protocols)." *Id.* at 32: 3–4, 41:8–13. When calls are made through the CVM 401, they are registered through the task dispatcher/controller 402 and the dialog (which can be multi modal and even without any speech input or output) is managed by the dialog controller 404. *Id.* at 32:13–16. Petitioner has not shown that the CVM's decision about which application is already active and has made a call to the CVM constitutes selecting a domain agent based on the determined meaning of an utterance.

c) Limitations 1[f]–1[h]

As designated by Petitioner, limitation 1[g] recites "invoking the selected domain agent to process the formulated request," i.e., a request that results from step 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." Pet. 7. Limitation 1[h] recites "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.” *Id.* Patent Owner does not discuss limitation 1[f] or 1[h]. We focus our analysis on limitation 1[g].

The Petition argues that the conversationally aware application processes the request formulated in step 1[f] because the application carries out processing in accordance with the request. Pet. 41 (citing Ex. 1002, Schmandt Decl. ¶ 136 (as discussing Zadrozny’s conferencing schedule example)). Noting that a domain agent provides a functionality for a particular area or realm, such as a particular subject matter or device, Patent Owner argues that Coffman’s conversationally aware application does not teach limitation 1[g] because Coffman’s conversationally aware application does not process a request. Prelim. Resp. 29, 35–39. Noting that in Coffman, each application has its own context stack, Patent Owner argues that Coffman’s description of the context stack contents for various discourses does not support Petitioner’s theory that a conversationally aware application is invoked to process a user’s request. *Id.* at 37 (citing Ex, 1003, 27:1–2). As Patent Owner points out, calls to conversational sub-systems (*see* Ex. 1003, Fig. 2, 218, Fig. 4, 407) are hidden to the application. Prelim. Resp. 37 (citing Ex. 1003, 29:10–11), *see also* Ex. 1003, 29:7–9 (“CVM 401 will locate the requested conversational subsystem 407 (via the conversational subsystem services manager 406), drive its execution and return appropriately the results.”)). As a conversational subsystem, rather than a conversationally aware application executes the request, we determine that Petitioner has not demonstrated that Coffman discloses limitation 1[g].

d) Claim 1 conclusion

In view of the above, we determine that Petitioner has not demonstrated that the combined teachings of Coffman, Kanevsky, Kneser, and Zadrozny teach imitations 1[e] or 1[g] to an ordinarily skilled artisan and that Petitioner has not shown a reasonable likelihood it would prevail in its challenge to claim 1 based on the combined teachings of these references.

C. All Remaining Claims and Grounds

All remaining grounds concern claims 2–9 and 12, which depend directly or indirectly from claim 1 and rely on combining the references cited in Petitioner's first ground with other references. Pet. 17. Petitioner explicitly states that "[a]ll grounds rely primarily on Coffman." *Id.* In view of our determination that Petitioner has not demonstrated Coffman teaches the limitations of claim 1 for which Coffman is cited, we determine that Petitioner has not demonstrated that the references in any of its asserted grounds teach the limitations of challenged dependent claims 2–9, and 12.

X. CONCLUSION

For the reasons discussed above, we are not persuaded that Petitioner has demonstrated a reasonable likelihood that it will succeed on any of the challenges to patentability.

XI. ORDER

In consideration of the foregoing, we decline to institute *inter partes* review.

FOR PETITIONER:

Heidi L. Keefe
COOLEY LLP
hkeefe@cooley.com

Andrew C. Mace (Reg. No. 63,342)
COOLEY LLP
amace@cooley.com

Mark R. Weinstein
COOLEY LLP
mweinstein@cooley.com

Reuben Chen
COOLEY LLP
rchen@cooley.com

FOR PATENT OWNER:

Wayne M. Helge
James T. Wilson
NOSSAMAN, LLP
whelge@nossaman.com
jwilson@nossaman.com