

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

DIALECT, LLC,
Patent Owner.

IPR2024-00751
Patent 8,447,607 B2

**PETITIONER'S REQUEST FOR DIRECTOR REVIEW
OF INSTITUTION DECISION¹**

¹ This IPR is one of five involving challenges to related patents that were denied for substantially the same reason (IPR2024-00746, IPR2024-00747, IPR2024-00748, IPR2024-00749, IPR2024-00751). Similar requests are being filed in all five proceedings.

I. INTRODUCTION

Petitioner Google LLC respectfully requests Director Review of the Board’s decision not to institute IPR of U.S. Patent No. 8,447,607 B2 (“’607 patent”). (Paper 11 (“Decision”).) Director Review is warranted because the Decision rejects the un rebutted testimony of Petitioner’s expert and credits Patent Owner’s pure attorney argument. At most, that attorney argument raises a factual dispute that the Board has repeatedly ruled is best resolved on a full record through trial. Moreover, the Decision’s reliance on that attorney argument is based on a clearly erroneous understanding and application of the Petition’s primary reference, Coffman², to the challenged claims. The Decision’s understanding of Coffman is not correct and does not withstand scrutiny. But even if correct, the proper course would have been to institute IPR and resolve any supposed factual dispute on a full record.

II. FACTUAL BACKGROUND

The ’607 patent and Coffman both involve controlling a computer using natural language voice commands. To determine which of several applications the user wants to invoke, the ’607 patent and Coffman both rely on the same mechanism: a “context stack” having context stack entries that are compared to the user’s request to determine a context and associated application that should carry out the request.

² Int’l Patent App. Pub. No. WO 00/20962 A2 (EX1003).

A. The Challenged Patent and Claims

The '607 patent provides a device with a speech interface for natural language human-machine interactions. (EX1001, 1:18-23.) The device allows “users to submit natural language speech and/or non-speech questions or commands in a wide range of domains.” (*Id.*, 1:23-27, 3:31-36.) Using information supplied by “domain agents,” user input is interpreted to determine a command or query to carry out, and an agent is selected to process it. (*Id.*, 6:16-23, 26:18-35, 27:13-23, 27:40-47.) Based on “context stack contents,” a “context” for user input is determined to identify a domain and related “domain agent” to invoke. (*Id.*, 33:34-36, 33:65-34:24.) Claim 1, with the key disputed limitation emphasized below, is illustrative:

1. A device for processing natural language inputs, comprising one or more processors configured to:
 - receive a multi-modal natural language input from a user, the multi-modal natural language input including a natural language utterance and a non-speech input;
 - generate a non-speech transcription from the non-speech input;
 - identify the user who provided the multi-modal natural language input;
 - generate a speech-based transcription based on a cognitive model associated with the user, wherein the cognitive model includes information on one or more prior interactions between the user and the device;
 - generate a merged transcription from the speech-based transcription and the non-speech transcription;
 - identify, from among a plurality of entries that are in a context stack and that are each indicative of context, an entry in the context stack that matches information in the merged transcription;

identify a domain agent associated with the entry in the context stack;

determine a request based on the merged transcription; and

communicate the request to the domain agent, wherein the domain agent is configured to generate a response to the user.

(EX1001, 38:38-62.)

B. The Petition and Institution Decision

The Petition challenged all claims as obvious over Coffman in view of Kennewick and Ross, but relied on Coffman alone for the key disputed limitation, **“identify a domain agent associated with the entry in the context stack.”** (Pet. at 10, 47-50). The Petition was supported by expert testimony (EX1002).

Coffman discloses a multi-modal “conversational” computing system that is remarkably similar to the ’607 patent. (Pet. at 15.) The system includes four key components for purposes of IPR (and this request): a conversational virtual machine (CVM), conversational subsystems, conversationally aware applications, and a context stack. The CVM “provides the capability to have natural dialog with NLU [natural language understanding], NLG [natural language generation], contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains).” (*Id.* at 22-23 (quoting EX1003, 6:6-8); *see also* EX1002 ¶42 (citing EX1003, 10:23-25, 11:10-13 (CVM “allows a computer...to converse with a user”).) The CVM does this by using conversational subsystems “including speech recognition, speaker recognition, text-to-speech, natural language understanding and

natural dialog generation engines to understand and generate dialog between user and machine.” (Pet. at 23 (citing EX1003, 5:1-4); *see also* EX1002 ¶42.)

The CVM system’s conversationally aware applications, which the Petition mapped to the claimed “**domain agent[s]**,” are applications (e.g., spreadsheet or name dialer applications) specifically programmed for operating with a CVM via application program interfaces (APIs). (Pet. at 47-49 (citing EX1003, 12:28-30, 58:30-32), 53 (citing EX1003, 45:2-6 (name dialer)); *see also* EX1002 ¶¶43, 98-100, 107.) A conversationally aware application registers and operates with a CVM by providing its data files (such as active vocabulary and language models) to the CVM for use by the conversational subsystems (such as the speech recognition engine). (Pet. at 48 (citing EX1003, 57:23-27, 36:31-32); *see also* EX1002 ¶43.)

The CVM “allows the user to run multiple tasks” by using a “context stack,” which the Petition (unsurprisingly) mapped to the claimed “**context stack**.” (Pet. at 36-37 (citing EX1003, 29:18-30:3, 30:4-10); *see also* EX1002 ¶85.) Coffman’s context stack “comprises all the information associated with an application” and its structure allows “maintain[ing] non-ambiguous conversational interactions with multiple tasks, threads or processes.” (EX1002 ¶44 (citing EX1003, 11:10-13, 21:23-24, 42:16-17); *see also* Pet. at 67 (citing EX1003, 21:23-26).) Coffman explains that “each new task, process, or thread creates a new stack entry and is associated with a discourse” and that “[e]ach application may be associated with

multiple discourses[.]” (Pet. at 37 (citing EX1003, 30:4-10, 30:13-16, 41:8-9); *see also* EX1002 ¶86.) Context stack entries are searched to determine a context matching user input and to activate the corresponding process. (Pet. at 37-39 (citing EX1003, 40:25-27, 42:1-5); *see also* EX1002 ¶¶86-87.) The search includes context stack entries of *multiple* applications: “The first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other application is stored to activate if the recognized query is rejected by the user.” (Pet. at 49 (quoting EX1003, 37:2-4); EX1002 ¶¶44, 102.)³

Thus, with respect to the key disputed limitation, **“identify a domain agent associated with the entry in the context stack,”** the Petition explained that Coffman’s context stack entries are **“associated with”** conversationally aware applications (**“domain agent[s]”**) because Coffman teaches that the entries are created for and associated with applications. (Pet. at 49 (citing EX1003, 30:4-10, 29:18-30); *see also* EX1002 ¶101.) The Petition explained that Coffman therefore teaches this **“identifying”** limitation because “by identifying a context stack entry matching the user’s input, the associated conversationally-aware application is selected to process a command/query based on the user’s input.” (Pet. at 49 (citing EX1003, 37:2-4, 42:1-5, 41:2-5, 40:12-15; EX1002 ¶102).)

The Decision narrowly focused on this **“identifying”** limitation and accepted

³ All emphasis added to quoted material unless noted.

that the Petition sufficiently demonstrated that the prior art disclosed the claimed “**domain agent**” and “**entry in the context stack.**” (Decision at 15, 18-19.) But then, contrary to unrebutted evidence and without the benefit of a full record, the Decision found that the Petition failed to sufficiently show that the prior art disclosed *identifying* a domain agent *associated with* the entry in the context stack. (*Id.* at 18-19.) The key issue in the Decision appears to be that Coffman’s disclosures that the “tasks,” “processes” or “threads” associated with context stack entries and executed or activated as a result of matching a context stack entry were not, in the Panel’s view, sufficiently tied to corresponding conversationally aware applications. (*Id.* at 19-22.) As explained below, Petitioner respectfully submits that the Petition and supporting evidence amply demonstrated this tie and that Coffman therefore discloses and renders obvious the claimed step of “*identify[ing] a domain agent associated with the entry in the context stack.*” Any factual dispute should have been resolved on a full post-institution record.

III. ARGUMENT

*A. The Petition Sufficiently Demonstrated, For Purposes of Institution, that Coffman Discloses and Renders Obvious “**identify[ing] a domain agent associated with the entry in the context stack.**”*

To start, the Petition sufficiently showed that a task/process/thread associated with a context stack entry in Coffman corresponds to an *application*. As discussed in II.B above, the Petition, supported by a full expert report, specifically pointed out

that a context stack in Coffman “comprises all the information associated with an application.” (Pet. at 67 (citing EX1003, 21:23-26).) The Petition further noted that Coffman discloses that context stack entries are created by “each new task, process, or thread”—which the Petition specifically linked to an application by noting that an “application process” would be an example of an “active task/process.” (*Id.* at 36-37 (citing EX1003, 29:18-30:3, 30:4-10, 30:13-16, 41:8-9).) The Petition further linked a task/process/thread to an application by noting Coffman’s statements that each task/process/thread is “associated with a discourse” and that “[e]ach application can be associated with multiple discourses.” (*Id.* at 37 (citing EX1003, 30:4-10, 30:13-16, 41:8-9).) Of course, a task/process/thread is not *exactly* the same as an application—rather, they describe a *running* application. This would have been common knowledge even to everyday Microsoft Windows users who have typed “CTRL-ALT-DEL” to open “Task Manager” to view information about (and force quit) application processes. This common knowledge is reflected in Coffman’s “BACKGROUND” section, which describes how Microsoft Windows provides “multi-tasking” by “displaying each process as a separate window.” (EX1003, 1:26-28.) Coffman explicitly links a task/process/thread to an application by explaining, as the Petition noted, that “[t]he first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other application is stored to activate if the recognized query is rejected by the

user.” (*Id.* at 49 (quoting EX1003, 37:2-4).) Coffman’s similar statements about having “a natural dialog...across multiple tasks, processes and discourses” and “a natural dialog...across multiple applications” also show this link. (*Compare* EX1003, 6:6-8 (cited in Pet. at 22-23) *with* EX1003, 36:29-31 (cited in Pet. at 49).)

The Petition also sufficiently showed that a task/process/thread associated with a context stack entry in Coffman corresponds not only to an application generally, but to a *conversationally aware application*. As the Petition noted, Coffman explains that the “core of the CVM 401 is the context stack” and that conversationally aware applications are “specifically programmed for operating with a CVM[.]” (Pet. at 48-49 (citing EX1003, 12:28-30, 29:18-30:3).) The Petition noted how conversationally aware applications register with a CVM by providing data files to the CVM, which Coffman elsewhere discusses in connection with a task/process. (*Compare* Pet. at 48 (citing EX1003, 57:23-27 (conversationally aware application “register[s] all its conversational capabilities and requirements” including “[a]ctive vocabulary, finite state grammar and language models to control the application” and “[s]ymbolic mapping if NLU is supported...”)) *with* Pet. at 37 (citing EX1003, 29:18-30:3 (“...the specific characteristics and full query arguments of the corresponding task (e.g, vocabulary file, language model, parsing, tags, voiceprint, TTS rules, grammar, NLU etc. of each active task/process)”)).) The Petition further noted that a context “discourse”—with which an application can be

associated—is a “conversational interaction associated with a given task/process/thread.” (*Id.* at 36 (quoting EX1003, 29:18-30:3).)

These disclosures, all detailed in the Petition, should leave little doubt that Coffman teaches identifying a conversationally aware application by matching a context stack entry associated with the application and selecting the application in order to process a query, thus disclosing and rendering obvious the claimed step of **“identify[ing] a domain agent associated with the entry in the context stack.”**

B. The Decision’s Understanding of Coffman Is Wrong, But Even If Correct, It Should Not Have Resulted in Institution Denial.

Against the Petition’s detailed analysis and expert support, Patent Owner offered only attorney argument, with no opposing expert. The Decision improperly credited Patent Owner’s attorney argument, and engaged in its own unsupported speculation, about the significance of isolated disclosures in Coffman, over the Petition and the considered testimony of Petitioner’s expert. *Icon Health & Fitness, Inc. v. Strava, Inc.*, 849 F.3d 1034, 1043 (Fed. Cir. 2017) (“attorney argument is not evidence”); *Vizio, Inc. v. Maxell, Ltd.*, IPR2022-01463, Paper 7 at 22 (PTAB) (crediting uncontested expert testimony and granting institution); *Tesla, Inc. v. Graphite Charging Co.*, IPR2024-00388, Paper 14 at 15 (PTAB Aug. 27, 2024) (same); *Hilti AG v. Specified Techs. Inc.*, IPR2024-00192, Paper 16 at 26-27, 33-34 (PTAB May 14, 2024) (instituting IPR, rejecting argument that petition and supporting expert with similar verbiage were “conclusory” because prior art

disclosures were sufficiently identified and mapping was reasonable). As explained below, Patent Owner's and the Decision's understanding of Coffman is wrong; but even if correct, it at most raised factual issues to resolve on a more complete post-institution record. *See, e.g., Samsung Elecs. Am., Inc. v. Snik LLC*, IPR2020-01324, Paper 9 at 20 (PTAB Mar. 9, 2021) ("At most, Patent Owner's alternative reading of [prior art] raises a factual dispute best resolved on a full record."); *Permian Global Inc. v. Fuel Automation Station, LLC*, IPR2023-01238, Paper 11 at 17 (PTAB Feb. 28, 2024) (instituting IPR where dispute that petition's "conclusory" statement did not include "much analysis" but was "supported by un rebutted testimony" was "best resolved after further development of the record").

Turning to the points raised in the Decision, it first states that "[w]e cannot discern...how Coffman's disclosure of determining a matching *discourse* in a context stack can correspond to determining a matching *application* such as one of Coffman's applications 11, especially when each application can be associated with multiple discourses." (Decision at 20 (emphasis in original).) The Panel was correct that a context stack entry is associated with a "discourse," but it is not clear how that undercuts Coffman or Petitioner's analysis. As the excerpts cited in the Petition explain, a "discourse" is a "conversational interaction associated with a given task/process/thread." (Pet. at 36 (quoting EX1003, 29:18-30:3).) Therefore, as the Petition explained, by matching a discourse, the associated task/process/thread of an

application is matched to activate/execute. (*Id.* at 49 (quoting EX1003, 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed...”)); *see also* EX1003, 31:24-29 (“A dialog controller 404 manages the dialog...across all the conversational and conventional applications (registered with the task dispatcher/controller 402).... The dialog controller 404 manages and determines the active context and application.”), 65:12-13 (claim 11) (“a dialog manager for managing dialog across the applications and for selecting an active dialog, context and application based on input queries”).)

Nor does Coffman’s statement that an application can be associated with multiple discourses provide any basis for denial. The Decision does not explain why this point matters. There is no requirement in any of the challenged claims that every entry in a context stack be associated with a different domain agent. Even if there are multiple context stack entries associated with the same application, by matching a particular entry, the associated application would be identified—and that should have been carried forward for a decision on a full record.

The Decision’s next assertion, that cited disclosures from Coffman “most pertinently describe Coffman’s context stack(s) as being associated with a ‘discourse’ as discussed above, or perhaps with a non-specific ‘engine’ or ‘network service’ or ‘process’ or ‘task’ or ‘thread’ or ‘subsystem’” (Decision at 20), fares no better. Even if the “tasks,” “processes,” and “threads” that Coffman describes could

be understood as also encompassing an “engine,” “network service” or “subsystem” (or anything else) there is no basis to exclude conversationally aware applications.

In fact, the whole point of Coffman is to enable voice and multi-modal user interaction with those applications—for example, Coffman describes how a “personal vehicle assistant” (PVA) “runs an embedded CVM while the two other applications are conversationally aware, i.e., you can control them through the CVM running on the PVA.” (EX1003, 56:22-25 (cited in Pet. at 16, 49); *see also id.*, 4:13-16 (“It is the conversational computing paradigm that allows a user to seamlessly control multiple Windows applications, for example, running in parallel....”).) There should be no doubt that such user interaction involves invoking those applications specifically. For instance, Coffman provides an illustration of the conversational system, explaining how it allows a user to “dial in your PC to consult or modify some spreadsheet information; the same way that you would 20 have consulted it by voice on your desktop while processing your mail.” (EX1003, 57:17-20 (cited in Pet. at 48, 66).) If a user asks for the value of a particular spreadsheet cell, wouldn’t the CVM select the conversationally aware spreadsheet application to lookup and provide that information? Coffman nowhere discloses such application-specific functionality is moved to or duplicated by the CVM or its subsystems.

And even if the tasks/processes/threads Coffman describes could be understood as encompassing an “engine,” “network service” or “subsystem” (or

anything else), it is enough that tasks/processes/threads would at least encompass conversationally aware applications, such that a conversationally aware application would at least sometimes be selected to process a query. *See Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations of prior art that sometimes meet the claim elements are sufficient to show obviousness.”).

The Decision then states that “Figure 1 of Coffman appears to show that conversationally aware applications 11 are directly connected to conversational user interface 10 to receive a user’s utterances at the ‘frontend’ of Coffman’s system” and that this is “at least some indication that Coffman’s system does not utilize CVM 14...to identify a particular application 11....” (Decision at 21-22.) But again, this interpretation is contradicted by Coffman’s explanation that “[t]he first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other application is stored to activate if the recognized query is rejected by the user.” (EX1003, 37:2-4.) In fact, as noted, Coffman expressly states that the CVM’s dialog manager “manag[es] dialog across the applications and... select[s] an... application based on input queries.” (EX1003, 65:12-13 (claim 11); *see also id.*, 20:19-23 (dialog manager determines “which inputs the user presents, and which application(s) should handle the user inputs”).) More broadly, Coffman touts its system as an improvement over prior art systems that were limited to “giv[ing] commands to the task in focus.” (EX1003, 3:14-16.)

Putting these disclosures aside, Figure 1 itself does not support the Decision's interpretation. The figure shows a single box 11 corresponding to a plurality of "conversationally aware applications," with no insight into the ways that the CVM may interact with those applications.

The Decision lastly asserts that statements in Coffman suggest that each conversationally aware application would have its own context stack, and that "again, this is at least some indication that Coffman's system does not utilize CVM 14 to identify a particular application 11 associated with an entry in a context stack that matches information in the user's merged speech and non-speech input." (Decision at 22.) The Decision relies mainly on Coffman's isolated statement that "[t]ypically, different devices, CVMs or different applications will have their own dialog manager, context stack 217, 217a and global history 218, 218a." (*Id.* at 22 (citing EX1003, 27:1-2).) But that statement relates to a particular embodiment involving a CVM system that is distributed across a network. (See EX1003, 26:28-27:2.) Coffman does **not** say that multiple conversationally aware applications that are all on the same device must each have their own context stack. In fact, Coffman elsewhere states that "dialog controller 404 will control the dialog of all applications and manage the context through the context stack 405." (EX1003, 32:27-28; see also *id.*, Fig. 4 (showing single context stack 405).) And notably, even in an embodiment involving a distributed environment, Coffman further explains that

device and application context stack information can be added to another context stack, such as for a master dialog manager. (*See* Pet. at 69-70 (citing EX1003, 25:10-14).) Coffman does not limit a context stack to a single application.

But even if Coffman only disclosed application-specific context stacks, Coffman’s disclosure of searching context stack entries to identify an application to activate/execute would still satisfy the disputed “**identifying**” limitation. As discussed above, the Petition made clear that Coffman’s CVM system searches context stack entries to determine not only which query should be executed but also to identify *which application* to activate to perform that query. (*See* Pet. at 49 (citing EX1003, 37:2-4.) Because the claims simply recite “a context stack”—i.e., *one or more* context stacks—there is no reason that a search across multiple context stacks would not meet the claim. *See, e.g., Baldwin Graphic Sys., Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1342-43 (Fed. Cir. 2008) (“That ‘a’ or ‘an’ can mean ‘one or more’ is best described as a rule, rather than merely as a presumption or even a convention. The exceptions to this rule are extremely limited....”).

IV. CONCLUSION

For the reasons above and in the Petition, Petitioner respectfully requests Director Review of the Decision, and for an order vacating the Decision and instituting *inter partes* review.

Dated: November 12, 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 SERVICE

I hereby certify, pursuant to 37 C.F.R. Section 42.6, that a complete copy of the attached **PETITIONER'S REQUEST FOR DIRECTOR REVIEW OF INSTITUTION DECISION** is being served via electronic mail on the 12th day of November 2024 upon Patent Owner's counsel of record in this proceeding as follows:

Wayne Helge
James Wilson
BUNSOW DE MORY, LLP
whelge@bdiplaw.com
jwilson@bdiplaw.com
BDIP-DialectIPRs@bdiplaw.com

DATED: November 12, 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