

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

DIALECT, LLC,
Patent Owner.

IPR2024-00748
Patent 8,015,006 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. 8,015,006 B2 (“’006 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 ’006 patent and Coffman both involve controlling a computer using natural language voice commands, for example, in a vehicle. To determine which of several applications the user wants to invoke, the ’006 patent and Coffman both rely on the same mechanism: evaluating the user’s request in order to determine a matching “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 challenged '006 patent relates to retrieving online information or processing commands through a speech interface. (EX1001, 1:20-22.) It purports to “provide[] a complete speech-based information query, retrieval, processing and presentation environment.” (*Id.*, 9:61-63.) Using information supplied by agents, recognized words and phrases are transformed into complete commands and questions, which the agents process. (*Id.*, 10:61-64.) To select the correct agent to invoke, a “context” for the input is determined. (*Id.*, 17:49-61, 18:12-14.)

Claim 1, with the key disputed limitation emphasized below, is illustrative:

1. A method for processing natural language speech utterances with context-specific domain agents, comprising:
receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;
recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables, wherein recognizing the one or more words or phrases contained in the utterance includes:
dynamically updating the information in the one or more dictionary and phrase tables based on a dynamic set of prior probabilities or fuzzy possibilities;
determining an identity associated with a user that spoke the utterance based on voice characteristics associated with the utterance; and
associating the one or more recognized words or phrases and a pronunciation associated with the one or more recognized words or phrases with the determined identity and the request contained in the utterance in response to the one or more recognized words or phrases satisfying a predetermined confidence level;
parsing, at a parser coupled to the processing device, the one or more

recognized words or phrases to determine a meaning associated with the utterance and a context associated with the request contained in the utterance, wherein the one or more recognized words or phrases are further associated with the determined context in response to the one or more recognized words or phrases satisfying the predetermined confidence level;

formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context;

processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance; and

presenting the generated response to the utterance via the speech unit.

(EX1001, 25:49-26:21.)

B. The Petition and Institution Decision

The Petition challenged claims 1-6 and 9 as obvious over Coffman in view of various references, but relied primarily on Coffman for the key disputed limitation, **“processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance.”** (Pet. at 18, 49-50.) The Petition was supported by unrebutted expert testimony (EX1002).

Coffman discloses a multi-modal “conversational” computing system that is remarkably similar to the ’006 patent. (Pet. at 21.) 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 25 (quoting EX1003, 6:6-8); *see also* EX1002 ¶¶44 (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 25 (citing EX1003, 5:1-4, 29:1-4); EX1002 ¶¶44, 87, 89.)

The CVM system’s conversationally aware applications, which the Petition mapped to the claimed “**domain agent**[s],” are applications (e.g., spreadsheet application) specifically programmed for operating with a CVM via application program interfaces (APIs). (Pet. at 42-44 (citing EX1003, 7:27-8:9, 13:8-13, 58:12-20, 58:30-32 (spreadsheet application)); *see also* EX1002 ¶¶45, 123-125.) 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 45-46 (citing EX1003, 57:22-27, 36:31-32); *see also* EX1002 ¶¶45, 125.)

The CVM allows the user to run multiple tasks by using a “context stack” comprising entries for “contexts” that the Petition (unsurprisingly) mapped to the claimed “**context**.” (Pet. at 35 (citing EX1003, 29:18-30:10), 40-41 (citing EX1003, 41:2-5, 42:1-5); *see also* EX1002 ¶¶46, 113, 120.) 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 ¶46 (quoting EX1003, 21:23-24, 42:16-17); *see also* Pet. at 29 (citing EX1003, 29:18-30:10).) 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 35 (citing EX1003, 30:4-10); *see also id.* at 53 (citing EX1003, 30:13-16, 41:8-9); EX1002 ¶¶113, 145.) Context stack entries are searched to determine a context matching user input and to activate the corresponding process. (Pet. at 35 (citing EX1003, 41:2-5, 42:1-5), 49-50 (citing EX1003, 37:2-4); *see also* EX1002 ¶¶114, 139, 145.) 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 49-50 (citing EX1003, 37:2-4); EX1002 ¶¶46, 114, 139).³

Thus, with respect to the key disputed limitation, “**processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance,**” because it largely refers to elements introduced earlier in the claim, the Petition likewise referred to earlier analysis and

³ All emphasis added to quoted material unless noted.

pointed again to Coffman’s teachings that the conversationally aware application (“**domain agent**”) associated with the determined context is invoked to process the user’s request. (Pet. at 49-50 (citing EX1003, 37:2-4, 30:4-10, 58:9-11, 40:9-19; “as explained for 1[d]...”), 42-45 (1[d] analysis, and referring to 1[b][iii] analysis), 35-36 (1[b][iii] analysis); *see also* EX1002 ¶¶114, 120, 139-140.)

The Decision narrowly focused on two supposed issues related to this “**processing**” limitation. (Decision at 15-20.) Contrary to un rebutted evidence and without the benefit of a full record, the Decision found that the Petition failed to sufficiently show: (1) that it was the conversationally aware application (“**domain agent**”) that *processed* the formulated request; and (2) that the conversationally aware application was *associated with* the determined context. (*Id.*) 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 16-19.) 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 “**processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance.**” 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 “**processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance.**”*

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, noted that Coffman discloses that context stack entries are created by “each new task, process, or thread”—which the Petition repeatedly specifically linked to an application by noting that an “application process” would be an example of an “active process.” (Pet. at 35, 40, 45 (citing EX1003, 29:18-30:10; EX1002 ¶127).) 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.* (citing EX1003, 30:4-10); *see also id.* at 53 (citing EX1003, 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 the “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....” (Pet. at 49-50 (citing EX1003, 37:2-4).) Coffman’s similar statements about having “a natural dialog...across multiple tasks, processes and discourses” and having “a natural dialog...across multiple applications” likewise show this link. (*Compare* EX1003, 6:6-8 (cited in Pet. at 25) *with* EX1003, 36:29-31 (cited in Pet. at 43).)

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 35 (citing EX1003, 29:18-30:3), 42-43 (citing EX1003, 13:26-29, 7:29-8:9, 58:12-20).) 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 27 (citing EX1003, 57:22-58:11 (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 id.* at 35 (citing EX1003, 29:18-30:3 (“...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 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 35 (citing EX1003, 29:18-30:3); EX1002 ¶113.)

These disclosures, all detailed in the Petition, should leave little doubt that Coffman teaches that it is a conversationally aware application that is invoked by matching a context stack entry associated with the application and selecting the application in order to process a query—and that Coffman thus discloses and renders obvious the claimed “**processing**” step.

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); *Hum Indus. Tech., Inc. v. Amsted Rail Co.*, IPR2023-00540, Paper 10 at 37 (PTAB Aug. 11, 2023) (expert testimony not conclusory because testimony supported by citations to record evidence). 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 focuses on two isolated

citations to Coffman (37:2-4 and 42:1-7) and asserts that they “do not teach clearly that the system selects or invokes a conversationally aware application (i.e., the alleged domain agent) associated with a determined context.” (Decision at 17.) But as discussed above, the Petition provided pages of analysis, with numerous other citations and expert support, to show that the applications/tasks/processes/threads in Coffman correspond to conversationally aware applications.

The Decision then points to two passages from Coffman that, in its view, “seem to suggest that user requests are processed by conversational sub-systems” rather than conversationally aware applications. (Decision at 18.) The Panel was correct that Coffman’s conversational sub-systems process user input (e.g., by performing speech recognition and natural language understanding). But conversationally aware applications are ultimately invoked to actually carry out the action requested by the user—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 43); *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; *see also* Pet. at 43 (citing EX1003, 58:30-32 (describing conversationally aware spreadsheet application))).) 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.

The Decision then faults another passage from Coffman for not using the word “application,” instead describing, for example, “executed or running processes.” (Decision at 18 (citing EX1003, 40:9-19).) But as discussed above, the totality of the Petition’s analysis and collected citations to Coffman showed that the “processes” associated with context stack entries correspond to conversationally aware applications.

The Decision next takes issue with “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 19 (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 35 (citing EX1003, 29:18-30:3).) Therefore, as the Petition explained, by matching a discourse, the associated task/process/thread of a conversationally aware application is matched to activate/execute. (*Id.* at 44 (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 last 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 19), 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. As explained, the whole point of Coffman is to enable voice and multi-modal user interaction with conversationally aware applications.

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.”).

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