

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

DIALECT, LLC,
Patent Owner.

Case IPR2024-00751
Patent 8,447,607

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

I.	THE PETITION’S GROUNDS FAIL TO SATISFY EVERY CLAIMED FEATURE, AND RELY UPON ART THAT WAS FULLY CONSIDERED BY THE EXAMINER DURING PROSECUTION	1
II.	THE ’607 PATENT DISCLOSES AND CLAIMS PROCESSING OF MULTIMODAL SIGNALS THAT IS NOT DISCLOSED OR SUGGESTED BY THE ASSERTED ART	4
A.	The ’607 Patent’s Use of Agents.....	6
B.	The ’607 Patent’s Speech Analysis.....	7
III.	THE BOARD NEED NOT REACH ANY CLAIM CONSTRUCTION ISSUES TO DENY THE PETITION.....	15
IV.	PETITIONER’S CHALLENGES FAIL ON THE MERITS.....	17
A.	Coffman’s Conversationally-Aware Applications Do Not Satisfy the Requirements of the Claimed “Domain Agent”	18
1.	Understanding Coffman Conversationally-Aware Applications	18
2.	Petitioner Fails to Establish that Coffman’s Conversationally-Aware Applications Perform the Requirements of Elements 1[g].....	29
3.	Ground 1 Fails, and Petitioner Presents No Alternative Theory for the Claimed “Domain Agent”.....	35
B.	Coffman Processes Multimodal Inputs Individually and Does Not Use a Merged, Natural Language Transcription to Identify a Context.....	36
1.	Coffman Processes Multimodal Inputs Individually	37
2.	Ground 1 Fails to Demonstrate that Coffman, Alone or in Combination with Ross, Identifies an Entry in a Context Stack That Matches Information in the Merged Transcription	42

IPR2024-00751 Patent Owner’s Preliminary Response

C.	Ground 1 Fails to Demonstrate that Coffman, Alone or in Combination with Kennewick, Has a Cognitive Model Associated with the User.....	44
D.	Ground 2 Fails as Lee is Not Shown to Have a Cognitive Model Associated with the User.....	46
V.	The Board Should Deny the Petition Based on 35 U.S.C. § 325(d)	47
VI.	CONCLUSION.....	51

TABLE OF AUTHORITIES

Cases

<i>Advanced Bionics, LLC v. MED-EL Electromedizinische Geräte GmbH,</i> IPR2019-01469, Paper 6 (PTAB Feb. 13, 2020).....	48
<i>Becton, Dickinson & Co. v. B. Braun Melsungen AG,</i> IPR2017-01586, Paper 8 (PTAB Dec. 15, 2017)	47, 48

Statutes

35 U.S.C. § 325(d)	4, 14, 47
--------------------------	-----------

Rules

37 C.F.R. § 42.107	1
37 C.F.R. § 42.120(a).....	1

PATENT OWNER’S EXHIBIT LIST

Exhibit No.	Description
2001	Patent Owner’s Power of Attorney
2002	U.S. Patent 7,693,720 (“the ’720 Patent”)
2003	U.S. Patent 8,015,006 (“the ’006 Patent”)
2004	U.S. Patent 8,195,468 (“the ’468 Patent”)
2005	U.S. Patent 9,263,039 (“the ’039 Patent”)
2006	U.S. Patent 9,495,957 (“the ’957 Patent”)
2007	Order (Construing Claim Terms), Dialect, LLC v. Amazon.com Inc., Civ. No. 1:23cv581, Dkt. 213 (E.D. Va. Apr. 29, 2024)

Pursuant to 37 C.F.R. § 42.107, Patent Owner Dialect, LLC (“Dialect” or “Patent Owner”) files this preliminary response to the Petition, setting forth reasons why the Petition for *inter partes* review (“IPR”) of claims 1-15 of U.S. Patent No. 8,447,607 (the “’607 Patent”), as requested by Google LLC (“Petitioner”), should be denied.¹

I. THE PETITION’S GROUNDS FAIL TO SATISFY EVERY CLAIMED FEATURE, AND RELY UPON ART THAT WAS FULLY CONSIDERED BY THE EXAMINER DURING PROSECUTION

The ’607 Patent relates to enabling natural language human-machine interactions with multimodal inputs, such as from a speech interface and non-speech interfaces. As part of the claimed device, independent claim 1 of the ’607 Patent requires the device to generate a speech-based transcription, identify an entry in a context stack, and “identify a domain agent” associated with an identified entry in a context stack. Independent claims 12 and 14 provide a similar requirement to “identify a domain agent.”

Petitioner pointed to a conversationally-aware application in the Coffman reference when attempting to identify the domain agent as part of this element. Pet., 47-50. But Coffman’s conversationally-aware applications use an application program interface (“API”) to send utterances to the backend of the system to be

¹ By submitting this Preliminary Response, no waiver of any argument is intended by Patent Owner. Patent Owner will have a right to file a complete response to the Petition, should the Board institute *inter partes* review. 37 C.F.R. § 42.120(a).

processed into requests and to receive responses for presentation to the user. As such, they cannot be the claimed “domain agent” because they are not “identified” as associated with an identified context stack entry.

Additionally, claims 1 and 12 each requires a “merged transcription” and identifying “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.” The Petition tries to show these features are present in Coffman, or obvious in view of Coffman and Ross, but fails to do so. Petitioner's error on this element stems from its failure to understand when the “merged transcription” and “identifying” steps occur in the claims. In the claims, the merged transcription is used to identify an entry in a context stack that matches information in the merged transcription. However, in Coffman, inputs are processed separately and placed into entries in the context stack separately, so there is no use of a merged transcription to identify an entry in a context stack that matches information in the merged transcription. Ross does not address this deficiency, particularly because it is not directed to processing multi-modal inputs.

Finally, each independent claim requires generating a speech-based transcription based on a “cognitive model associated with the user.” Petitioner relies on speech recognition techniques in Coffman and Kennewick to demonstrate this feature but neither reference discloses predictive responses for a user's next

actions. Thus, Petitioner fails to show that any speech recognition techniques in Coffman or Kennewick satisfy the requirement of a cognitive model associated with a user.

The Petition’s substantive failings are not the sole basis to deny the Petition. Petitioner acknowledges that the lead references relied upon in Grounds 1 and 2—Coffman and Kennewick—were already considered and discussed by the Examiner during prosecution. Petitioner contends that the Examiner did not recognize Coffman would render the identifying step obvious. This is wrong. The Examiner correctly understood Coffman and did not err because, as noted above, Petitioner’s theory for the identifying step misunderstands the import of Coffman’s disclosure concerning how multimodal inputs are separately processed. Petitioner’s secondary reference to Ross fails to fix this missing aspect of Coffman, particularly because Ross does not involve the processing of multi-modal inputs and “merged transcriptions,” and the addition of Ross does not alter the Examiner’s correct and reasoned decision to allow the claims over Coffman.

While Petitioner’s challenge in Ground 1 fails based on Petitioner’s analysis, the Petition also presents no reason why the Board should double-check the Examiner’s thorough analysis of Coffman. Concerning Ground 2, Petitioner attempts to rely on Lee to disclose a “second cognitive model,” but that does not change the fact that the references considered by the Examiner do not disclose the

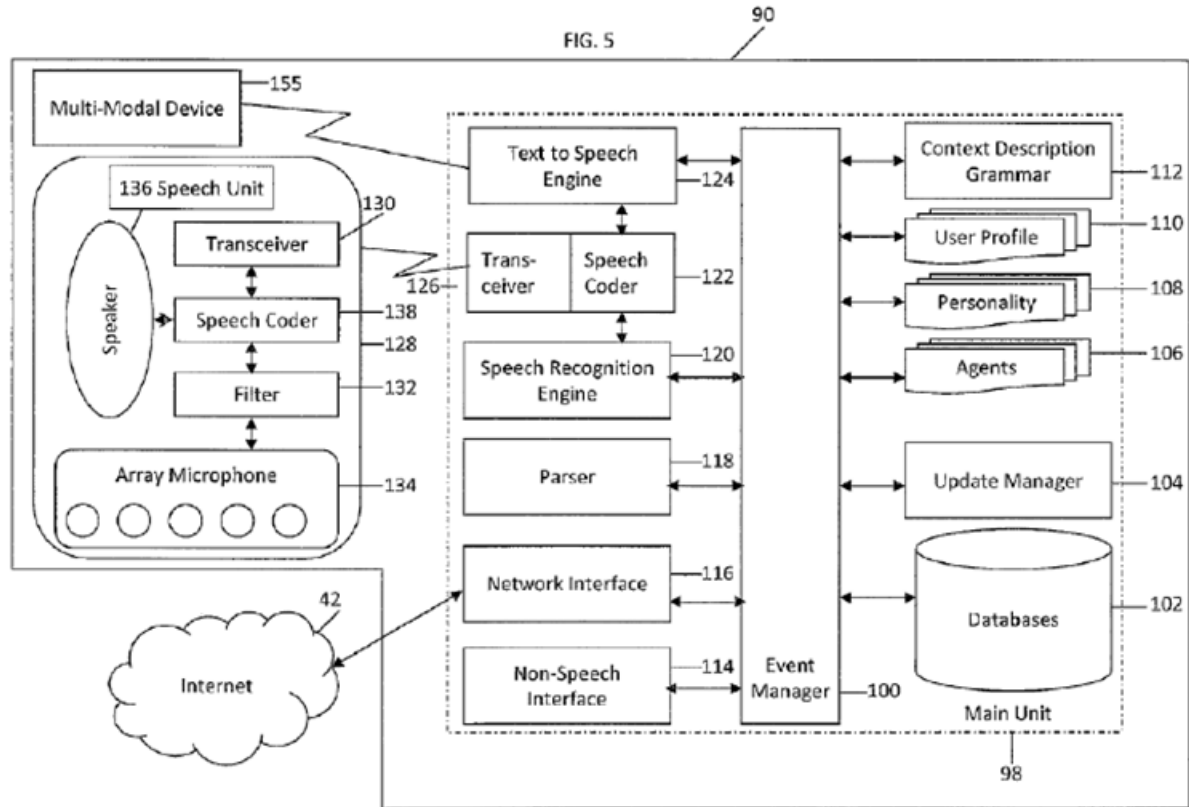
first claimed user-specific cognitive model, for which Lee is not relied upon.

Thus, the Petition should be denied under 35 U.S.C. § 325(d).

II. THE '607 PATENT DISCLOSES AND CLAIMS PROCESSING OF MULTIMODAL SIGNALS THAT IS NOT DISCLOSED OR SUGGESTED BY THE ASSERTED ART

The '607 Patent is entitled, “Mobile Systems and Methods of Supporting Natural Language Human-Machine Interactions,” and relates to enabling natural language human-machine interactions with multimodal inputs (such as from a speech interface and non-speech interfaces). Central to the claimed invention, the '607 Patent uses a cognitive model associated with the user to generate a speech-based transcription, merges natural language speech and non-speech questions or commands, and uses the merged transcription to identify the proper context entry for the multimodal input. This context entry is then used to identify a domain agent associated with the identified context entry.

As recounted in the Petition at 13-15, FIG. 5 of the '607 Patent, reproduced below, is an overall diagrammatic view of an embodiment of the '607 Patent:



'607 Patent, FIG. 5; *see also id.*, 15:38-40. As shown in the figure above, the system includes, among other things, speech unit 128, speech recognition engine 120, context description grammar module 112, parser 118, agents 106, multi-modal device 155 and non-speech interface 114. Speech unit 128 includes a microphone to receive a spoken utterance from a user. '607 Patent, 25:50-52.

After initial processing, the speech is passed to speech recognition engine 120 for processing using the context description grammar module 112. '607 Patent, 26:18-20. Then, “[a]ny recognized information may be processed by the parser 118, which transforms information into complete algorithms and questions using data supplied by knowledge agents” or “domain agents.” '607 Patent, 26:20-23; *see*

also id., 27:41-45. The knowledge agents or domain agents “may then process the commands or questions.” ’607 Patent, 26:27-31; *see also id.*, 36:51-53. Non-speech interface 114 and multi-modal devices 155 may be used as a substitute or complement to the speech interface. ’607 Patent, 26:51-53.

A. The ’607 Patent’s Use of Agents

As noted above, the ’607 Patent’s system 90 includes different types of agents 106. ’607 Patent, 27:13-14. As shown in the exemplary agent architecture of FIG. 6, agents include domain agents 156, a system agent 150, and an agent library 158 included within FIG. 1’s “agents 106.” *Id.*, 27:49-52. “A system agent ... provides default functionality and basic services.” *Id.*, 27:17-18. “[G]eneric and domain specific behavior and information may be organized into domain agents 156.” *Id.*, 27:15-17.

Speech “may be processed by the speech recognition engine 120 using the context description grammar module 112.” ’607 Patent, 26:18-20. “Both system agents 150 and domain agents 156 may receive questions and commands from the parser 118. Based on keywords in the questions and commands and the structures of the questions and commands, the parser may invoke selected agents.” *Id.*, 27:41-45. “[T]he domain agent 156 may continue to make queries and evaluate results until a satisfactory response is constructed.” *Id.*, 36:51-53. Also, “recognized information may be processed by the parser 118, which transforms information

into complete algorithms and questions using data supplied by knowledge agents.

Knowledge agents may be selected from the plurality of agents 106 using a grammar stack, wherein the knowledge agents provide information for generating a response to the question or command.” *Id.*, 26:2-26.

In particular, the parser may invoke a domain agent 156 among the agents 106. Specifically, the “[t]he parser may determine a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile 110, dialog history, and context stack contents. *The context of a question or command may determine the domain and, thereby, the domain agent 156, if any, to be invoked.*” *Id.*, 34:2-7 (emphasis added).

B. The '607 Patent's Speech Analysis

FIG. 8, reproduced below, illustrates aspects of speech analysis of the '607 Patent in more detail.

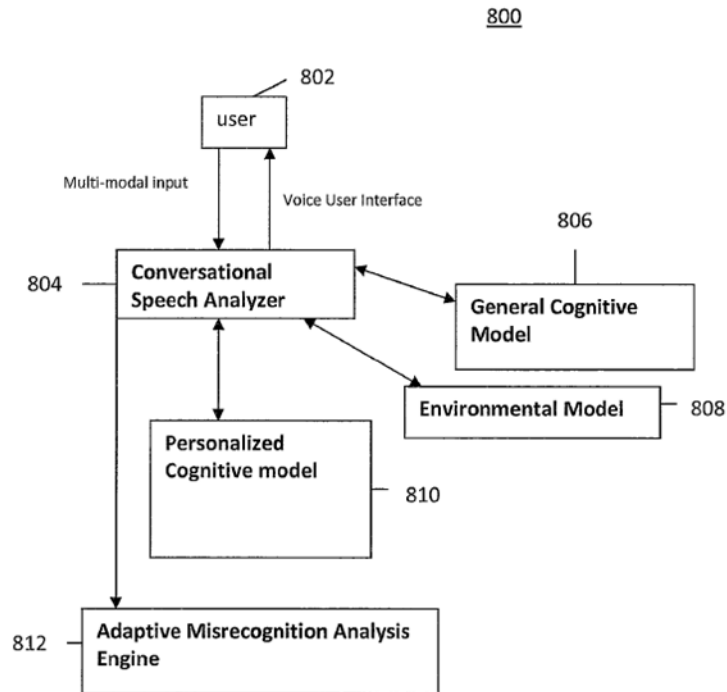


FIGURE 8

In FIG. 8, an enhanced conversational user interface 800 receives multi-modal input from users 802 and communicates with a conversational speech analyzer 804. '607 Patent, 23:48-51. The conversational speech analyzer 804 may be coupled to several components, including a general cognitive model 806, an environmental model 808, a personalized cognitive model 810, and an adaptive misrecognition analysis engine 812 or other components. '607 Patent, 23:51-55.

The conversational speech analyzer 804 may include one or more speech recognition engines that transcribe utterances to textual messages. '607 Patent, 23:56-58. The transcription may be performed using one or more conversational grammars, a context description grammar 112, or other transcription methods.

'607 Patent, 23:58-61. Conversational speech analyzer 804 also may include a semantic knowledge-based model that analyzes the textual message and detects command components. '607 Patent, 23:64-67.

FIG. 8 also illustrates a personalized cognitive model 810 and a general cognitive model 806. A personalized cognitive model 810 is a model derived from a user's interaction pattern with the system and may be used to "predict what actions the user may take next in time," thus assisting with speech recognition and/or question or command recognition. '607 Patent, 24:1-6.

A general cognitive model 806 is a statistical abstract that corresponds to interaction patterns with the system for multiple users. '607 Patent, 24:12-15. Data stored within general cognitive model 806 may be used to "predict a user's next action," thus also assisting with speech recognition and/or question or command recognition. '607 Patent, 24:15-17.

An environmental model 808 may include information associated with the user's environment and surroundings. The information may include the type of environment that a user is in (e.g., quiet or noisy); details of a microphone and/or speaker system; the user's current global position and movement, as may be determined by GPS; current system status, such as what song/movie is playing, is the system in the midst of retrieving something, or other system status; details on all voice-enabled devices in the immediate vicinity, such as a presence of a voice-

enabled TV, stereo, and DVD player in the same room; a user's credit card information, such as numbers and current balances, wherein the user may ask a mobile telephone to download and pay for a video and the system may respond that there are insufficient funds; or other information. '607 Patent, 24:22-36. The information may be accessed to invoke a context, domain knowledge, preferences, and/or other cognitive qualities that enhance the interpretation of questions and/or commands. '607 Patent, 24:37-39.

The conversational speech analyzer 804 may also access the general cognitive model 806 and/or the personalized cognitive model 810 to further refine a context, domain knowledge, preferences, and/or other cognitive qualities to enhance the interpretation of questions and/or commands. '607 Patent, 24:40-44.

As described in the '607 Patent, context information may be determined from a command or request that is presented in a textual format and/or a command or request that is presented as an utterance and processed using a multi-pass automatic speech recognition module that transcribes the utterance to a text message. '607 Patent, 3:66-4:2.

Consistent with the '607 Patent's disclosure, each independent claim (claims 1, 12, and 14) recites use of a cognitive model associated with the user to generate a speech-based transcription, identification of a context stack entry based on the speech-based transcription (claim 14) or a merged transcription (claims 1 and 12),

and identifying a domain agent associated with the identified context stack entry.

Independent claims 1 and 12 of the '607 Patent also require a multi-modal natural language input produced by merging a speech transcription and non-speech transcription, and use of that merged transcription to identify an entry in a context stack. Claim 1 recites:

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.

'607 Patent, claim 1 (emphasis added).

Claim 12 similarly recites:

12. A method for processing natural language inputs, comprising:
receiving, by one or more processors, 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;
generating, by the one or more processors, *a non-speech
transcription* from the non-speech input;
identifying, by the one or more processors, the user who provided
the multi-modal natural language input;
generating, by the one or more processors, *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*;
generating, by the one or more processors, *a merged transcription
from the speech-based transcription and the non-speech
transcription*;
identifying, by the one or more processors, *an entry in a context
stack*, from among a plurality of entries that are in the context
stack and that are each indicative of context, wherein the
*identified entry matches information in the merged
transcription*;

identifying, by the one or more processors, a *domain agent* associated with the entry in the context stack;
determining, by the one or more processors, a request based on the merged transcription; and
communicating, by the one or more processors, the request to the domain agent, wherein the domain agent is configured to generate a response to the user.
'607 Patent, claim 12 (emphasis added).

Claim 14 receives a natural language utterance from a user and employs a combination of a personal cognitive model and a general cognitive model to generate a speech-based transcription:

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

identify a domain agent associated with the entry in the context stack;
determine a request based on the speech-based transcription; and
communicate the request to the domain agent, wherein the domain agent is configured to generate a response to the user.
'607 Patent, claim 14 (emphasis added).

As acknowledged in the Petition, the Examiner expressly found claims 1 and 12 allowable over Kennewick and Coffman, noting that the references do not disclose “where an entry in a context stack (which includes a plurality of entries that are each indicative of context) that matches information *in the merged transcription* is identified (merged from the speech and non-speech-based transcription where the speech-based transcription is generated based on the cognitive model including information on one or more prior interaction).” Pet., 11 (citing Ex. 1012, 152) (emphasis added). Regarding claim 14, the Examiner noted that the claim “similarly recites, in combination with the remaining limitations of claim [14], the identification, from among a plurality of entries that are in a context stack and that are each indicative of context, of an entry that matches information in the speech-based transcription.” Ex. 1012, 152. For reasons discussed in more detail in Section V, the Examiner’s prior consideration of this asserted art warrants denial under 35 U.S.C. § 325(d).

III. THE BOARD NEED NOT REACH ANY CLAIM CONSTRUCTION ISSUES TO DENY THE PETITION

The Petitioner identifies no terms as requiring constructions. The United States District Court for the Eastern District of Virginia has issued constructions for the following terms (Ex. 2007):

Term	Patent(s)	Claim(s)	Construction
context	'720 Patent	1, 31	"the subject matter area to which a particular user input is directed and which is used to determine the meaning of the user input"
	'006 Patent	1, 2, 4, 5, 10	
	'468 Patent	19, 28	
	'039 Patent	17, 18	
context stack	'006 Patent	2	"a data structure that stores or references contexts in a manner that can be ordered"
	'468 Patent	19, 28	
	'039 Patent	17	
	'957 Patent	1, 3, 4, 7	
entries in a context stack	'468 Patent	19	"elements of a context stack that store or reference contexts"
context entries	'957 Patent	1, 3, 4, 5, 7	"elements of a context stack that store or reference contexts"

IPR2024-00751 Patent Owner's Preliminary Response

grammar	'720 Patent	1, 31	"a set of rules governing how inputs and requests to a domain agent should be structured"
	'006 Patent	1, 5, 10	
context description grammar	'039 Patent	13	"a data structure containing entries constituting or referencing sets of rules, wherein each of those sets describes the structure of natural language in a particular context"
domain agent	'720 Patent	1, 31	"executables associated with a specific domain that receive, process and respond to user questions, queries and commands"
	'006 Patent	1, 2, 5, 10	
	'468 Patent	19, 30, 32	"software with domain-specific behavior and information"
	'039 Patent	13, 14	
procedures sensitive to the determined context	'006 Patent	5	"procedures that differ based on which context has been determined"
parser	'720 Patent	1, 35	"software that analyzes a string of words"
	'006 Patent	1, 5, 10	

In the table above, U.S. Patent 8,195,468 (“the ’468 Patent”) (Ex. 2004) is the parent of the ’607 Patent and U.S. Patent 9,495,957 (“the ’957 Patent”) (Ex. 2006) is related, through other continuations, to the ’607 Patent. ’607 Patent, 1; Ex. 2006, 1-2. U.S. Patent 7,693,720 (“the ’720 Patent”) (Ex. 2002), U.S. Patent 8,015,006 (“the ’006 Patent”) (Ex. 2003), and U.S. Patent 9,263,039 (“the ’039 Patent”) (Ex. 2005) are each assigned to Patent Owner and share at least one inventor in common with the ’607 Patent.

Without waiving its right to submit constructions in a Patent Owner Response, Patent Owner submits that the Board need not engage in any pre-institution claim construction. For the reasons presented in Section IV below, Petitioner’s grounds include fundamental flaws and fail to establish the disclosure of every claimed element in each independent claim. These findings do not require the Board to conduct any construction of the above-identified terms.

IV. PETITIONER’S CHALLENGES FAIL ON THE MERITS

Petitioner asserts two Grounds in its Petition. Ground 1 asserts that claims 1-15 are obvious in view of Coffman,² Kennewick,³ and Ross⁴. But throughout its discussion, Petitioner makes statements suggesting it is relying on Coffman alone to demonstrate obviousness, in absence of Ross, for certain limitations in the independent claims. *See, e.g.*, Pet., 39 (“To the extent there is any question whether Coffman alone discloses or renders obvious ... it would have been obvious in view of Ross”) (emphasis omitted). As noted above, Coffman was known to the Examiner and relied upon by the Examiner, and Petitioner’s attempt to second-guess the Examiner based on its incorrect interpretation of Coffman

² Ex. 1003, International Patent Application Publication No. WO 00/20962 (“Coffman”).

³ Ex. 1004, U.S. Patent App. Pub. No. 2004/0044516 (“Kennewick”).

⁴ Ex. 1005, U.S. Patent App. Pub. No. 2002/0133354 (“Ross”).

should fail. Ground 2 contends claims 5-7 and 13-15 are obvious in view of Coffman, Kennewick, Ross, further in view of Lee⁵.

As summarized in Section I, Grounds 1 and 2 are both facially deficient in establishing the unpatentability of any challenged claim. The Petition should be denied without institution for the reasons set forth below.

A. Coffman’s Conversationally-Aware Applications Do Not Satisfy the Requirements of the Claimed “Domain Agent”

Element 1[g] of the ’607 patent requires a device to “identify a domain agent associated with” an identified entry in a context stack. Petitioner points to a “conversationally-aware application” in Coffman when attempting to disclose this element. Pet., 47-50. A correct understanding of Coffman’s conversationally-aware applications, as recounted below, demonstrates that they cannot satisfy the requirements for the claimed “domain agent” in element 1[g].

1. Understanding Coffman Conversationally-Aware Applications

- a) Coffman’s non-proprietary speech engine and application program interfaces (APIs) receive speech inputs from and send responses to conversationally-aware applications but are not identified based on identified contexts**

As a background to Coffman’s disclosure, Coffman was seeking to provide a backend system that would more easily allow a developer to program a voice-

⁵ Ex. 1006, U.S. Patent App. Pub. No. 2002/0087315 (“Lee”).

based interactive telephone application to converse with a user, without having to build an entire backend speech engine or API. Coffman notes that “voice command and control (voice C&C) UIs [user interfaces] are emerging everywhere computers are used.” Coffman, 3:1-2. “For instance, telephone companies, call centers and IVR have implemented speech interfaces to automate certain tasks, reduce their operator requirements and operating costs and speed-up call processing.” Coffman, 3:5-7. However, Coffman also notes that interactive voice response (“IVR”) “application developers offer their own proprietary speech engines and APIs (application program interface),” and that “proprietary applications are typically not portable from vendor to vendor.” Coffman, 3:7-8.

Coffman also states that “speech interfaces for GUI based OSs have been implemented using commercially available continuous speech recognition applications for dictation and command and control.” Coffman, 3:12-14. Coffman describes these “applications” as “add-ons to the GUI based OSs in the sense that such applications allow for the replacement of keyboard and mouse and allows a user to change the focus, launch new tasks, and give voice commands to the task in focus.” Coffman, 3:14-16.

As explained, Coffman’s purpose is to provide a “conversational virtual machine” with a speech engine and application program interfaces (APIs) usable by application developers “for implementing conversational computing” with

conventional applications and conversationally-aware applications. Coffman, 1:7-17. The point of Coffman, therefore, is to provide a “universal conversational user interface.” Coffman, 3:30.

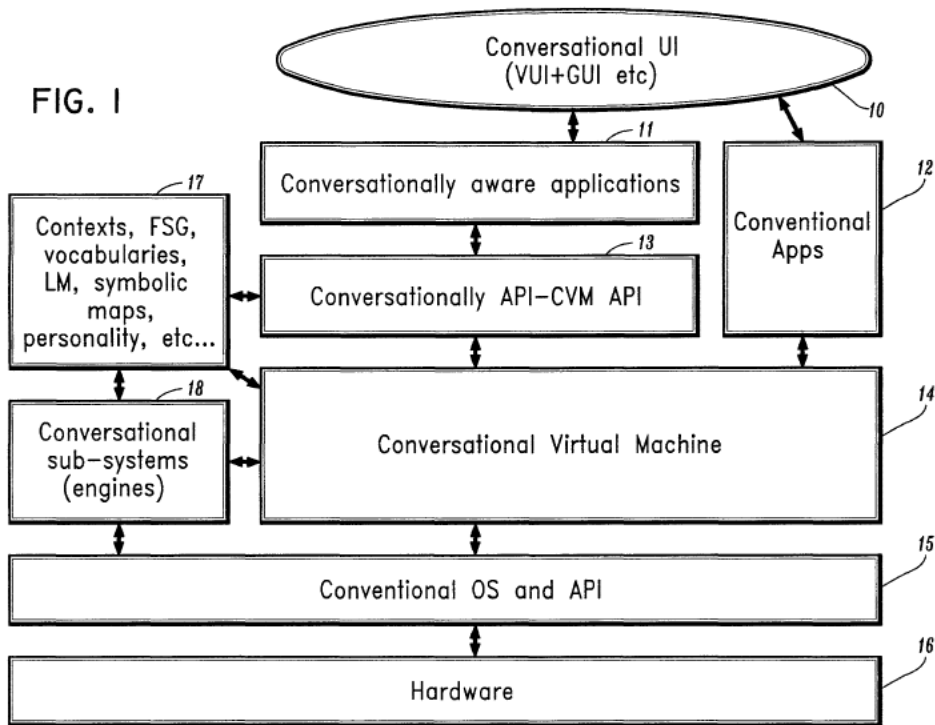
Coffman's Conversational Virtual Machine, or “CVM,” “utilizes conversational subsystems (which may be local or distributed) including speech recognition, speaker recognition, text-to-speech, natural language understanding and natural dialog generation engines to understand and generate dialog between ... user and machine.” Coffman, 5:1-4. Coffman refers to these aspects of the CVM as a “subsystem,” and notes that the “subsystem [is] accessed through the CVM.” Coffman, 5:4-5. As an aid to software developers, Coffman explains that the “development environment offered by the CVM ... allows a developer to build, simulate and debug conversational aware application [*sic*] for CVM.” Coffman, 18:16-18. An application developer does not need to build a speech engine or APIs, and instead accesses the CVM through the system's established APIs. The system's speech “engines are hidden to the application through the conversational application APIs.” Coffman, 5:5-6.

The relationship of Coffman's CVM to conventional applications, conversationally-aware applications, and the APIs is illustrated in Coffman's FIG.

1. Coffman's FIG. 1 is described as a “block diagram of a conversational

computing system according to an embodiment of the present invention.”

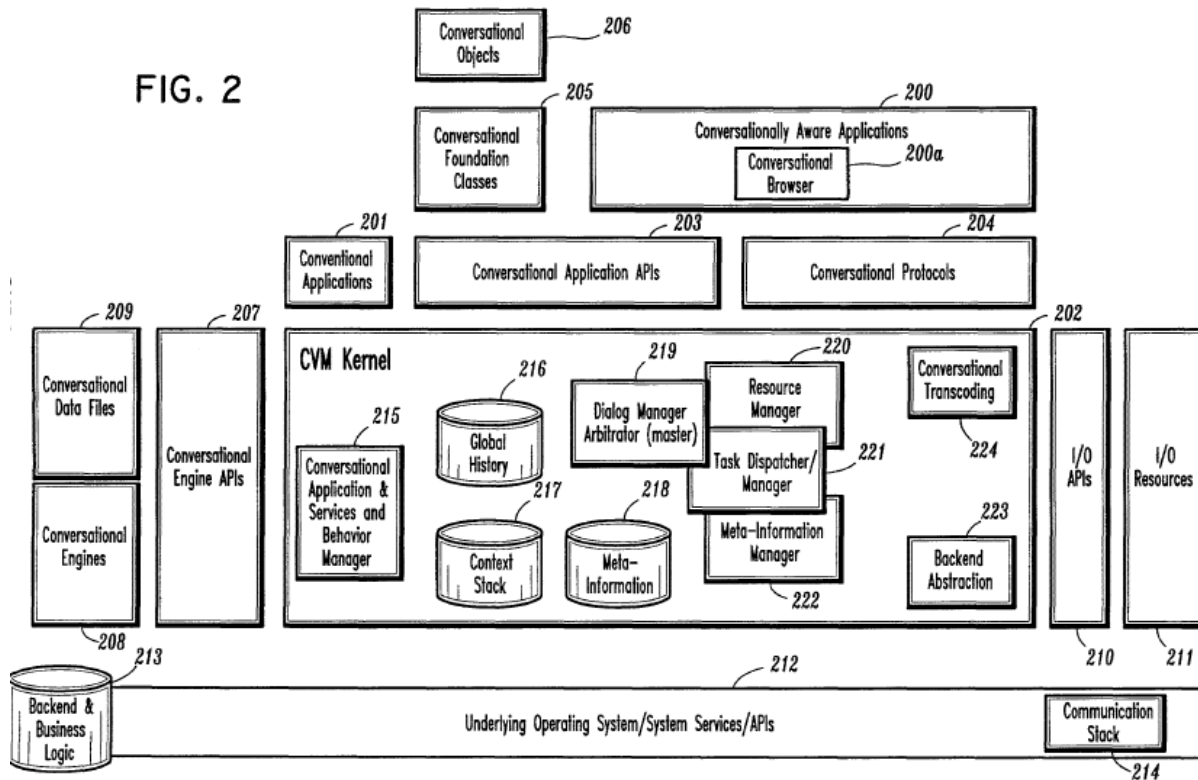
Coffman, 9:27-28.



Per Coffman, the CVM system “comprises a plurality of applications including conversationally aware applications 11 (i.e., applications that ‘speak’ conversational protocols) and conventional applications 12.” Coffman, 12:26-28. The difference between these types of applications, per Coffman, is that “conversationally aware applications 11 are applications that are specifically programmed for operating with a CVM core layer (or kernel) 14 via conversational application APIs 13.” *Id.*, 12:28-30. Conventional applications interact with the CVM, but do not need to do so through the conversational application APIs 13.

Thus, for both conventional applications and conversationally-aware applications, the “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 user interface ... to provide conversational system behaviors.” *Id.*, 12:30-13:1. In particular, Coffman’s conversationally-aware applications 11 are directly connected to the User Interface 10 to receive a user’s utterance, and interface with the CVM 14 via conversational application API.

Indeed, the conversational application API serves as an intermediary between the conversationally-aware applications on the frontend, and the backend system including CVM 14, conversational sub-systems 18, and backend logic. The CVM and its components are illustrated and described in more detail with respect to Coffman’s FIG. 2, reproduced below. The CVM Kernel 202 in FIG. 2 corresponds to the CVM Kernal 14 illustrated in FIG. 1.



Relevant to the current Petition, the CVM Kernel 202 includes a Context Stack 217, a Dialog Manager Arbitrator (master) 219 (which is frequently referred to in Coffman as the Dialog Manager 219), and a Backend Abstraction 223. There is also a “conversational application and behavior/service manager layer 215,” which “comprises functions for managing the conventional and conversationally aware applications 200 and 201.” *Id.*, 19:30-31. Coffman describes these management functions as including “for example, keeping track of which applications are registered (both local and network-distributed), what are the dialog interfaces (if any) of the applications, and what is the state of each application.” *Id.*, 19:31-20:2.

Describing the context stack 405, Coffman discloses that “[t]he core of the CVM 401 is the context stack 405 which operates and is managed under the control of the dialog controller 404.” *Id.*, 29:19-20. Coffman defines the term “context” to refer “to the state of each discourse (whether active or nonnative), which keeps track of the past history of the discourse, its current state, and the specific characteristics and full query arguments of the corresponding task (e.g, vocabulary file, language model, parsing, tags, voiceprint, TTS rules, grammar, NLU etc. of each active task/process) along with any data needed for input understanding.” *Id.*, 29:27-32.

The application-specific “context stack 217 is managed by the dialog manager 219.” *Id.*, 21:23; *see also id.*, 27:1-2 (disclosing that each application will have its “own ... context stack ...”). “More specifically, each new task, process, or thread creates a new stack entry and is associated with a discourse.” *Id.*, 30:4-5. “Each context associated with a given discourse comprises the latest requests made to the corresponding process/task/thread as well as the latest output.” *Id.*, 30:6-8.

The dialog manager 219 (or dialog controller 404) “manages the context stack 405 by creating a new stack entry in the context stack 405 for each new task/process/thread that is spawned either local or remotely from a networked device.” *Id.*, 30:13-15. “The context stack 405 is associated with the machine state stack so that for any new input from a user, the dialog controller 404 may

traverse the context stack 405 until the input context can be appropriately established. This essentially amounts to finding and selecting the active discourse between the user and machine among the last and past discourses.” *Id.*, 30:22-25; *see also id.*, 41:9-13.

“Each application may be associated with multiple discourses (e.g. the application management discourse and the application content navigation discourses).” *Id.*, 29:4-8. The context stack 217 “comprises all the information associated with an application.” *Id.*, 21:23-24. “Such information includes all the variable, states, input, output and queries *to the backend* that are performed in the context of the dialog and any extraneous event that occurs during the dialog.” *Id.*, 21:24-26 (emphasis added). The CVM Kernel 202 also includes a “global history 216,” which “includes information that is stored beyond the context of each application,” such as “the information that is associated with all the applications and actions taking during a conversational session (i.e., the history of the dialog between user and machine for a current session (or from when the machine was activated).” *Id.*, 21:28-32.

Importantly, Coffman’s mention in this passage of “queries to the backend” is explained further by the inclusion of Coffman’s “backend abstraction layer 223” in the CVM Kernel 202. The “backend abstraction layer 223 ... allows access to backend business logic 213 via the dialog manager 219 (rather than bypassing the

dialog manager 219).” Coffman, 22:1-3. The “backend abstraction layer 223 can translate input and output to and from the dialog manager 219 to database queries. This layer 223 will convert standardized attribute value n-uples into database queries and translate the result of such queries into tables or sets of attribute value n-uples back to the dialog manager 219.” *Id.*, 22:4-7. As such, Coffman’s backend can process queries formulated at the CVM, and provide the results back to the CVM for communicating them to a conversationally-aware application, which presents the results of the processed query to the user.

This understanding of Coffman is also supported by FIG. 4. FIG. 4 of Coffman, reproduced below, provides a block diagram of Coffman’s system 400 and CVM 401 (Coffman, FIG. 4 and at 28:10-12):

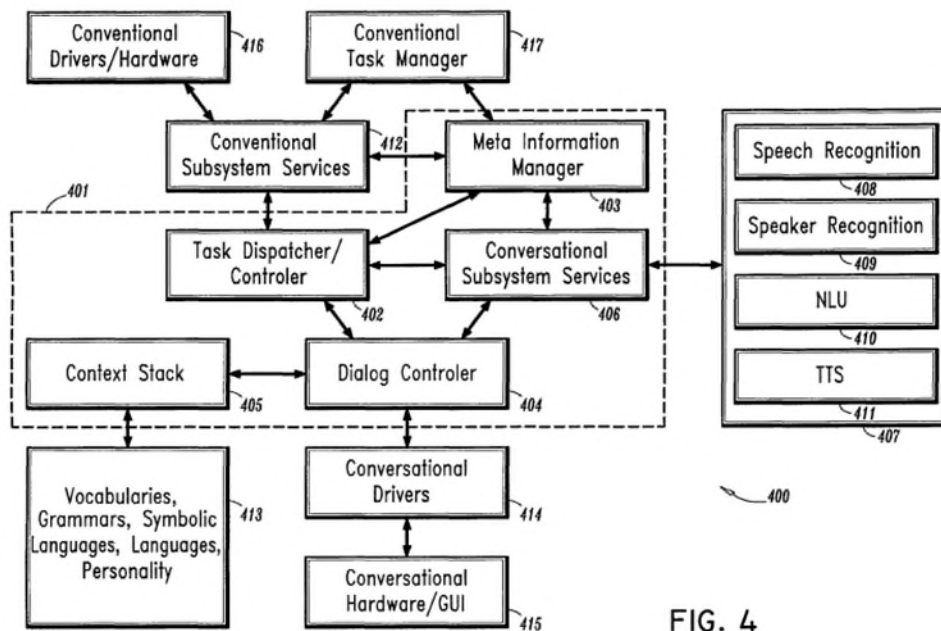


FIG. 4

The conversational system 400 comprises a combination of conventional subsystems and conversational subsystems which are executed and managed by a CVM 401. The CVM 401 comprises a task dispatcher/controller 402, a meta information manager 403, a dialog controller/manager 404, a context stack 405, and a conversational subsystem services manager 406. The “CVM controller” of Coffman is understood to refer collectively to the task dispatcher/controller 402 and the dialog controller 404. In CVM 401, “instructions/commands that are initiated by the user are explicitly *sent to the background* of the system.” *Id.*, 43:1-2 (emphasis added).

This disclosure is critical to a proper understanding of Coffman's CVM, and how it relates to the conversationally-aware applications disclosed in Coffman. In particular, a user who interacts with a conversationally-aware application on the front end of Coffman's system will communicate with the conversationally-aware application via speech, and the conversationally-aware application will access the CVM layer (or kernel) via conversational application APIs so that the CVM can perform the backend operations. And, as noted, Coffman's CVM and backend operations are hidden from the conversationally-aware applications. *See* Coffman, 5:5-6, 29:10-11. As disclosed by Coffman: “conversational sub-systems 18 are responsible for converting voice requests into queries and converting outputs and results into spoken messages using the appropriate data files 17 (e.g., contexts,

finite state grammars, vocabularies, language models, symbolic query maps etc.).”

Coffman, 13:8-10.

b) Coffman's Context Stacks are Unique to Each Application

Coffman discloses that a CVM may interact with multiple applications. The CVM provides applications and devices with “conversational services and behaviors as well as conversational protocols for interaction with *multiple applications* and devices also equipped with a CVM layer, or at least, conversationally aware.” Coffman, 11:2-4 (emphasis added). However, Coffman discloses that where there are different applications, “different devices, CVMs or different applications will have their own dialog manager, context stack 217, 217a and global history 218, 218a.” Coffman, 27:1-2.

A context stack for an application is further described with respect to FIG. 4's context stack 405, “which operates and is managed under control of the dialog controller 404.” *Id.*, 29:18-19. The context stack 405 “accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 (or at least identifiers of such conversational arguments) for the different engines that may be needed for input understanding (e.g., files or arguments that the engines use for performing their respective tasks such as active FSG, topic, vocabulary,

HMM (hidden markov models), voiceprints, language models or possible queries for a speech input).” *Id.*, 29:21-27. As such, a search of an application’s own context stack 217a may identify an active context and active process for that application. But this search does not support Petitioner’s assertion that the CVM would then identify a conversationally-aware application based on the active context.

2. Petitioner Fails to Establish that Coffman’s Conversationally-Aware Applications Perform the Requirements of Elements 1[g]

As introduced above, element 1[g] of claim 1 recites, in relevant part, “identify a domain agent associated with the entry in the context stack.” Petitioner alleges that “the executable software of a conversationally-aware application (and associated data ...)” discloses the claimed “domain agent” of claim 1. Pet., 47 (citing Schmandt, ¶99); *see also* Pet., 49 (alleging that “context stack entries are created for and associated with applications” and “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”).

However, this characterization fails because Coffman fails to disclose any “identify[ing]” or selecting a conversationally-aware application, much less identifying such a conversationally-aware application based on an identified “entry in the context stack.”

As Petitioner concedes, a context stack entry in Coffman is “identified when an acceptable match between the user’s input and context stack is found.” Pet., 38 (citing Coffman 42:1-5); *see also* Coffman, 30:22-25 & 41:9-13. But Coffman’s determination of an active context does not lead to identifying or selecting of any conversationally-aware application as a “domain agent.”

Petitioner’s argument for element 1[g] of claim 1 appears to be founded on Petitioner’s unsupported supposition that identifying a context stack entry identifies the domain of the user input. This fundamentally misunderstands Coffman. Understood correctly, Coffman discloses that a user initiates interaction with a conversationally-aware application, which passes user information to the backend system for identifying an entry in a context stack. Coffman does not disclose a conversationally-aware application being identified based on the user information or based on the identified context stack entry.

Moreover, Coffman discloses that “all conversational subsystem calls are hidden to the application” Coffman, 29:10-11; *see also id.*, 5:5-6. The CVM 401, not any application, “will locate the requested conversational subsystem 407 (via the conversational subsystem services manager 406), drive its execution and return appropriately the **results**” to the application. *Id.*, 29:7-9 (emphasis added). Thus, Petitioner is incorrect that identification of a context in a context stack yields the activation, identification, or selection of a conversationally-aware application.

Instead, the subsystems are invisible to the conversationally-aware application, which receives the final results to be reported to the user.

In attempting to argue that Coffman discloses the step of “identify[ing] a domain agent associated with the entry in the context stack,” Petitioner relies on Coffman’s disclosure at 37:2-4 to assert that, in Coffman’s CVM, an “associated *conversationally-aware application is selected* to process a command/query based on the user’s input.” Pet., 49 (emphasis added). Coffman’s disclosure does not support this assertion, however. And rather than providing any analysis, Petitioner merely quotes this passage in Coffman, stating “[t]he first task to have its dialog completed by producing a non-ambiguous query is executed.” Pet., 49 (citing Coffman, 37:2-4). No link exists between that passage and selecting or identifying a conversationally-aware application.

Neither the Petition nor paragraph 102 of the Schmandt Declaration even attempts to explain what the sentence at 37:2-4—which is directed to queries, not identifying applications—is attempting to disclose, or why it is relevant to the purported identification or selection of a conversationally-aware application. It is a mere conclusion without any linking analysis, and should not be relied upon.

Moreover, Petitioner’s “see also” citations to Coffman at “42:1-5, 41:2-5, 40:12-15” (*see* Pet. 50; Schmandt, ¶102) also fail to advance Petitioner’s argument as these passages do not discuss conversationally-aware applications and instead

confirm that Petitioner's theories are inconsistent with Coffman's actual operation.

Coffman's disclosure at 42:1-5 is reproduced below:

Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the *context becomes active and the corresponding process becomes active*.

Coffman, 42:1-5 (emphasis added). Coffman's disclosure of a "process becom[ing] active" is not the same as identifying or selecting a conversationally-aware application. Indeed, Coffman associates a "process" with tasks and threads performed *at the CVM*. See Coffman, 42:16-17 ("This stack structure allows to maintain non-ambiguous conversational interactions with multiple tasks, threads or processes."); 21:1-3 ("The task dispatcher/manager 221 dispatches and coordinates different tasks and processes that are spawned (by the user and machine) on local and networked conventional and conversational resources"); 30:1-2 ("context stack (as well as other CVM components)"); 30:4-5 ("[E]ach new task, process, or thread creates a new stack entry and is associated with a discourse."); FIG. 2 (illustrating CVM components including task dispatcher/manager 221).

Coffman's disclosure at 41:2-5 also does not identify any conversationally-aware application, as it describes how the different input stream modes are redirected "to the appropriate conversational subsystems or conventional subsystem with speech inputs being transcribed/understood," with the "output of these subsystems [being] run down the context stack 405 to extract the active query and complete it." Coffman, 41:2-5. Of course, for any new input from a user, the dialog controller or manager in the CVM "may traverse the context stack 405 until the input context can be appropriately established." Coffman, 30:22-25; *see also* Coffman, 41:9-13. But Coffman's process for identifying context stack entries occurs at the CVM, and not within a conversationally-aware application and not to identify a conversationally-aware application.

Similarly, Coffman's disclosure at 40:12-15 is also not directed to identifying or selection a conversationally-aware application. The elements identified in Coffman's disclosure at 40:12-15 are elements within Coffman's CVM, not in Coffman's conversationally-aware application, and are not explained as elements that identify or select a conversationally-aware application. *See* Coffman, 40:12-15 ("task dispatcher/controller 402," "context stack 405," "statistical parsers," "dialog controller 404").

Accordingly, nothing about these disclosures suggests that the CVM identifies or selects a conversationally-aware application, as Petitioner contends.

Rather, Coffman expressly discloses that the task dispatcher/controller 402 of the CVM 401 “dispatches and coordinates different tasks and processes that are spawned” Coffman, 30:26-28. Nothing in Coffman’s discussion of the task dispatcher/controller 402 suggests or discloses identifying a conversationally-aware application based on a match in the context stack, or even that the task dispatcher/controller 402 is capable of selecting a conversationally-aware application. Instead, Coffman selects a context stack based on the user’s interaction with a conversationally-aware application.

Finally, confirming that Coffman’s determination of an active context in the context stack does *not* lead to an identification of a conversationally-aware application based on an identified context entry, Coffman discloses that where there are different applications, “different devices, CVMs or different applications will have their own dialog manager, context stack 217, 217a and global history 218, 218a.” Coffman, 27:1-2. Similarly, Coffman explains that a “context stack 217 comprises all the information associated with ***an application.***” Coffman, 21:23-24 (emphasis added). As such, a search of an application’s own context stack 217 may identify an active context and active process for that application, but does not support Petitioner’s expansive reading that Coffman’s CVM will then identify a conversationally-aware application among many applications to process a user’s request based on the context stack search. Rather, a user interacting with a

conversationally-aware application in Coffman results in, at most, an identification of an active context for that user’s current discourse with that application. *See* Coffman, 30:4-5 (“[E]ach new task, process, or thread creates a new stack entry and is associated with a discourse.”); 29:4-5 (“Each application may be associated with multiple discourses”); 30:23-24 (“[F]or any new input from a user, the dialog controller 404 may traverse the context stack 405 until the input context can be appropriately established. This essentially amounts to finding and *selecting the active discourse between the user and machine* among the last and past discourses.”) (emphasis added). This does not disclose or suggest an identification or selection of that conversationally-aware application by Coffman’s CVM, contrary to Petitioner’s theory.

Thus, Coffman fails to disclose the claimed “identifying” step of element 1[g].

3. Ground 1 Fails, and Petitioner Presents No Alternative Theory for the Claimed “Domain Agent”

For the foregoing reasons, Petitioner’s reliance on Coffman’s conversationally-aware application to disclose the claimed “domain agent” fails to satisfy every requirement of claim 1, including element 1[g] as discussed above. Further, Petitioner advances no alternative theory in any asserted Ground to shore up the deficiencies concerning element 1[g] in this regard. Thus, the Petition fails with regard to Ground 1 and independent claim 1. Additionally, Petitioner

incorporates its erroneous analysis against the other independent claims 12 and 14 (Pet. 72-73), and since all other challenged claims depend directly or indirectly from independent claims 1, 12 and 14, this defect is fatal to the Petition.

B. Coffman Processes Multimodal Inputs Individually and Does Not Use a Merged, Natural Language Transcription to Identify a Context

Petitioner also fails to establish the disclosure or suggestion of elements 1[e] and 1[f]. Element 1[e] of the ’607 Patent requires “generat[ing] a merged transcription” from a speech-based transcription and a non-speech transcription and element 1[f] of the ’607 patent requires a device to ““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.*”

Petitioner points to multimodal inputs in Coffman and its “context stack” when attempting to disclose these features (Pet., 34-39) and, alternatively, relies on Ross to disclose element 1[f]. A correct understanding of how Coffman uses its context stack, as recounted below, demonstrates that it cannot satisfy the requirements of elements 1[e] and 1[f], whether considered alone or in combination with Ross.

1. Coffman Processes Multimodal Inputs Individually

Coffman provides a multimodal system that allows a user to interact with its system using speech and non-speech interfaces, but Petitioner's theories do not identify a merged transcription in Coffman that is used to identify a context, as required by claims 1 and 12. Instead, as explained below, as an input is received in Coffman, its values are placed into a "n-uple" in a context entry. The result of Coffman may produce "n-uples" in context entries with both speech and non-speech content, and those "n-uples" may be used to produce queries and commands, but those "n-uples" are not used to perform the predicate step of identifying an entry in a context stack that matches merged transcript information.

In FIG. 6, reproduced below, Coffman discloses a conversational input/output interface (Coffman, FIG. 6, 37:14-29):

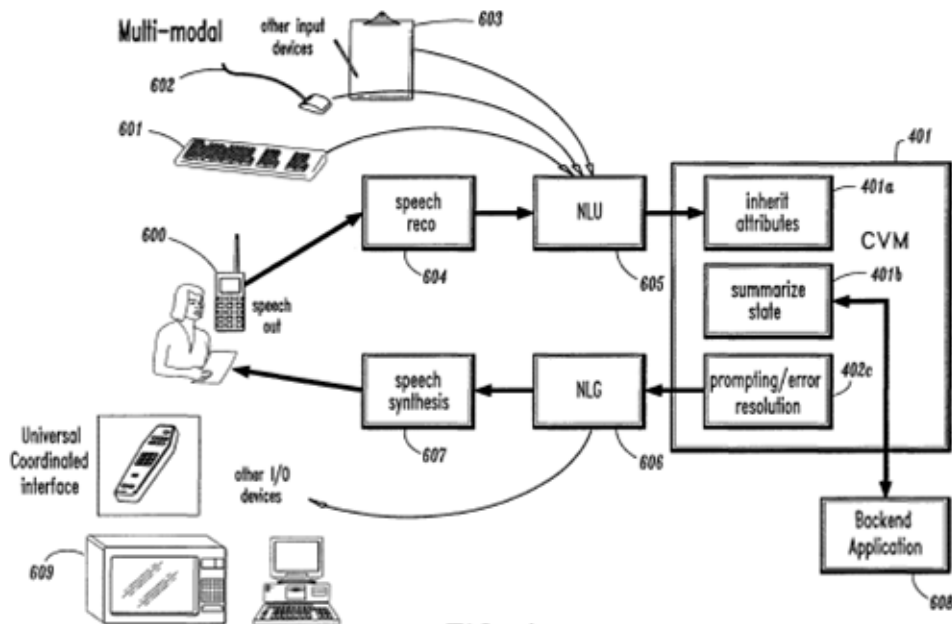


FIG. 6

The conversational input interface according to Coffman can process multi-modal inputs, such as files/streams/resources, speech via a phone 600, keyboard 601, pointing devices 602, and handwriting devices 603. The modalities are caught and transferred to a dialog manager that stores the inputs in the context stack. Spoken input from a speech client is subject to a speech recognition process 604 and other input are subject to NLU processing 605.

Coffman explains that each input is subject to attribute acquisition (401a) and “attribute value n-uples are acquired from the input.” Coffman, 37:22-23. A summarization process 401b is performed whereby “the attribute value n-uples are added to the context.” Coffman, 37:23-24. A verification can take place to determine whether the query is complete, incomplete, or ambiguous. Coffman, 37:24-26.

Coffman explains that “each input stream behaves ... conventionally,” (Coffman, 37:29), and notes a problem where simultaneous input streams to be merged can create ambiguity. Coffman, 38:2-3. Coffman addresses the input problem by treating it as “a merge of the output of multiple decoders, ASCII transcription or a list of attribute value n-uples.” Coffman, 38:8-9. Coffman’s speech recognition process 604 converts each input stream into an ASCII transcription and aligns the stream with input time-marks. Coffman, 38:9-11.

Notably, Coffman states that “[w]hen different input streams are associated to the same task,” “transcripts are merged.” Coffman, 38:11-12. Thus, concerning the timing aspect of when the transcripts are merged, Coffman’s process merges content only *after* it is determined that the streams are associated to the same task. Confirming that timing, Coffman explains that its process allows for checking the commands and queries in the inputs against rules and grammar for the task. Coffman, 38:13-19.

Accordingly, Coffman’s check to determine if different input streams are associated to the same task indicates that Coffman determines the context for the input streams *prior to* the decision to merge the information in the input streams. Indeed, Coffman’s further detail of its “input/output processing by CVM 401” confirms this understanding. Coffman, 40:9-19.

FIG. 4 of Coffman, reproduced below, provides a block diagram of Coffman’s system 400 and CVM 401 (Coffman, FIG. 4, 28:10-12):

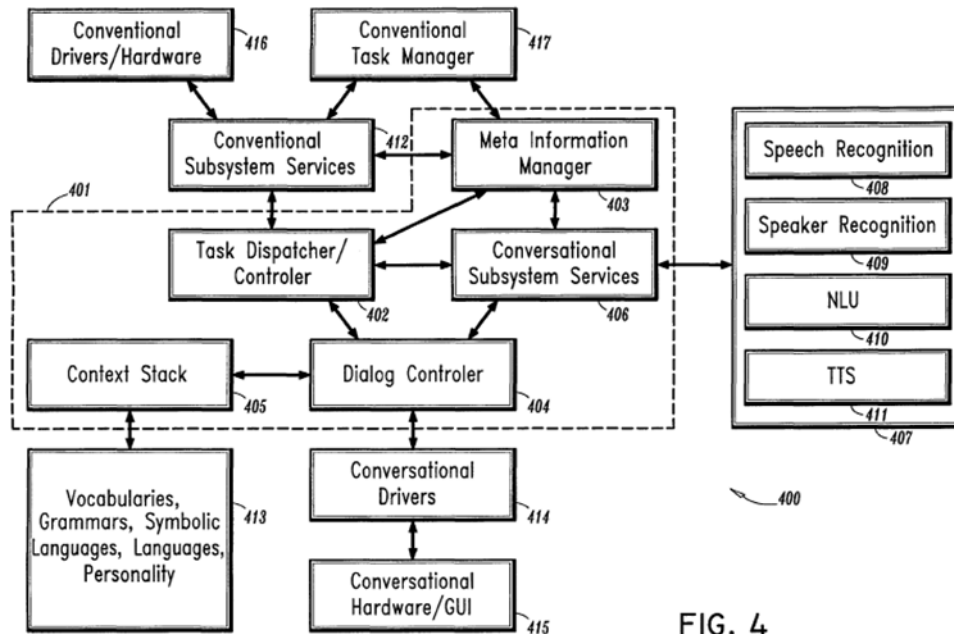


FIG. 4

The conversational system 400 comprises a combination of conventional subsystems and conversational subsystems which are executed and managed by a CVM 401. The CVM 401 comprises a task dispatcher/controller 402, a meta information manager 403, a dialog controller/manager 404, a context stack 405, and a conversational subsystem services manager 406. Coffman, 28:22-27. The “CVM controller” of Coffman is understood to refer collectively to the task dispatcher/controller 402 and the dialog controller 404. *See* Coffman, 28:27-29.

Coffman explains that the CVM 401 operates by converting conversational and conventional input streams into multiple actions and produces sorted output to a user through conversational and/or conventional output streams. Coffman, 28:29-31. Coffman describes that “[b]ased on the modality (pointer, keyboard, file, speech), the task dispatcher 402 redirects the stream to the appropriate

conversational subsystems or conventional subsystem with speech inputs being transcribed/understood.” Coffman, 41:2-4. Importantly, and confirming that the inputs are merged within, and not prior to identifying, an entry in a context stack, Coffman expressly states, “[t]he output of these subsystems is run down the context stack 405 to extract the active query and complete it.” Coffman, 41:4-5. Further, Coffman explains that the “context stack 405 is associated with the machine state stack so that *for any new input from a user*, the context stack 405 can be traversed until the input context can be appropriately established.” Coffman, 41:9-11 (emphasis added).

In this way, Coffman eliminates ambiguity that may be caused by different inputs and executes commands and queries with confidence. As Coffman explains, “new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained ... As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active.” Coffman, 42:2-5.

Accordingly, “the context stack 405 accumulates the context (i.e., full query arguments *list of attribute value n-uples*, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 ... for the different engines that may be needed for input understanding (e.g., files or arguments that the engines

use for performing their respective tasks such as active FSG, topic, vocabulary, HMM (hidden markov models), voiceprints, language models or possible queries for a speech input).” Coffman, 29:21-27 (emphasis added).

Petitioner’s failure to establish identify a context using a merged transcription is fatal to its Ground 1 challenges, as will be explained further below.

2. Ground 1 Fails to Demonstrate that Coffman, Alone or in Combination with Ross, Identifies an Entry in a Context Stack That Matches Information in the Merged Transcription

Addressing element 1[e], Petitioner contends that merging of speech and nonspeech transcriptions occurs in a “compounded input stream.” Pet., 34-35. But when addressing element 1[f], Petitioner loses sight of how Coffman teaches that “compounded input stream” is created and where it is stored.

As recounted in the Petition, and above in Section IV.B.1, in Coffman, “a set of multi-mode input streams are each transcribed into an ASCII command or query (i.e., lists of attribute-value pairs or n-uples)” (Pet., 35 (emphasis in original) (citing Coffman, 6:6-12)), and “the context stack 405 accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread).” Pet., 36 (emphasis in original) (citing Coffman, 29:18-30:3). Thus, in Coffman, it is the context stack that accumulates commands with the attribute value n-uples.

However, Petitioner errs in suggesting that Coffman identifies an entry in a context stack that matches information in the “merged transcription” because of the attribute value n-uples. Petitioner alleges that by “taking Coffman’s example involving the utterance ‘I would like to open this’ and mouse click input, Coffman teaches that the system matches the recognized word ‘open’ in the user’s utterance and the mouse click information to a context entry for an application’s ‘open’ command.” Pet., 38-39 (citing Coffman, 6:32-7:3). Petitioner’s assertion that Coffman combines inputs prior to matching information in a context stack is unsupported and wrong.

Unlike the recitation in element 1[f], Coffman’s inputs are not merged before identifying an entry in a context stack. Indeed, in the sentence immediately preceding Petitioner’s citation for “Coffman’s example,” Coffman itself explains that each input is *individually* “added” to the associated context or stack, not that inputs are added to each other before identifying an entry in a context stack. Coffman, 6:30-32 (“any input in any modality is captured by the dialog manager of the CVM kernel layer as *an event that is added to the associated context or context stack.*”) (emphasis added). Petitioner’s theory attempts to combine inputs before traversing the context stack, but does not account for the fact that Coffman explains that the inputs in Coffman are placed into the context stack individually and separately. Thus, Petitioner’s theory fails to disclose or suggest the claimed

features found in independent claims 1 and 12 of the ’607 Patent relating to merging of transcriptions (element 1[e]) to be used for identifying entries in a context stack (element 1[f]).

Petitioner does not address this deficiency with respect to Ross, either. Seemingly in an attempt to try to avoid 35 U.S.C. § 325(d), Petitioner relies on Ross to provide a combination that identifies “a context stack entry that matches information in a transcribed user’s utterance.” *See, e.g.,* Pet., 45. But Petitioner does not rely on Ross to show multimodal inputs or creating a merged transcription from multimodal inputs. Accordingly, Petitioner fails to show that its proposed combination of Coffman and Ross discloses or suggests these claim requirements in independent claims 1 and 12 of the ’607 Patent.⁶ Thus, Ground 1 fails with regard to independent claims 1 and 12 and corresponding dependent claims 2-11 and 13.

C. Ground 1 Fails to Demonstrate that Coffman, Alone or in Combination with Kennewick, Has a Cognitive Model Associated with the User

The Petition also contends that Coffman discloses the user-specific “cognitive model” aspects of each independent claim (claims 1, 12, and 14) (*see,*

⁶ Ground 1 fails to disclose or suggest other features of independent claims 1 and 12 and corresponding dependent claims 2-11 and 13, but Patent Owner does not include those deficiencies in the Preliminary Response in the interests of brevity and focus.

e.g., element 1[d]) because “Coffman teaches that the collected information can be used ‘to perform more robust recognition’ and to ‘recognize the user.’” Pet., 29 (citing Coffman, 37:5-10); Pet., 72-74 (incorporating analysis of claim 1 with regard to claim 14).

But Petitioner fails to address the fact that the cognitive models in the ’607 Patent include prediction of a user’s future actions. *See* ’607 Patent, 24:2-6 (“a model derived from a user’s interaction pattern with the system and may be used to predict what actions the user may take next in time, thus assisting with speech recognition and/or question or command recognition.”); Pet., 29 (“With respect to a ‘**cognitive model**,’ the ’607 patent describes it as information derived from a user’s interactions with the system that may be used in determining a user’s future actions.”) (emphasis in original). Thus, even Petitioner appears to acknowledge that the claimed “cognitive models” include prediction of a user’s future actions to assist with speech recognition and/or question or command recognition.

“[M]ore robust recognition” and recognizing a user, as relied upon in Coffman, is not shown in any way in the Petition to predict a user’s future action. Neither the Petition nor Petitioner’s declarant explain any theory as to how “robust” or “user” recognition should be equated to prediction of a user’s future actions. Similarly, the Petition looks to Kennewick and appears to rely on Kennewick’s “speech recognition engine” that “learns with time and improves

accuracy,” Pet., 32 (citing Kennewick, ¶¶0149-0151), to provide a combination with “collected profile information and using the profile information to *provide improved speech recognition*.” Pet., 33. But once again, there is no showing by Petitioner of any prediction of a user’s future actions in Kennewick or the asserted combination.

For all the foregoing reasons, Ground 1 fails concerning all challenged claims 1-15.

D. Ground 2 Fails as Lee is Not Shown to Have a Cognitive Model Associated with the User

The Petition presents Ground 2 and introduces Lee in order to contend “ it would have been obvious to use *other users’ profile information* involving users’ interactions with the device, such as prior questions and commands, in connection with performing speech recognition.” Pet., 75 (emphasis added). However, Petitioner does not contend or demonstrate that Lee discloses or suggests use of a cognitive model associated with the specific user, as recited in claims 1, 12, and 14, or remedy the deficiencies of Coffman and Kennewick noted above in that regard.

Thus, Ground 2 fails concerning challenged claims 5-7, which find dependence from claim 1, claim 13, which is dependent on claim 12, and claims 14-15.

V. The Board Should Deny the Petition Based on 35 U.S.C. § 325(d)

Section 325(d) allows the Board to elect not to institute a proceeding if the challenge to the patent is based on matters previously presented to the Office. The statute states, in pertinent part, “[i]n determining whether to institute . . . , the Director may take into account whether, and reject the petition . . . because, the same or substantially the same prior art or arguments previously were presented to the Office.” 35 U.S.C. § 325(d). The question of whether a petition presents art or arguments that are “the same or substantially the same” as art or arguments previously presented to the Office is a factual inquiry, which may be resolved by reference to the factors set forth in *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8 (PTAB Dec. 15, 2017) (precedential as to § III.C.5, first paragraph) (“*Becton, Dickinson*”).

The precedential section of that decision sets forth the following non-exclusive factors for consideration:

- (a) the similarities and material differences between the asserted art and the prior art involved during examination;
- (b) the cumulative nature of the asserted art and the prior art evaluated during examination;
- (c) the extent to which the asserted art was evaluated during examination, including whether the prior art was the basis for rejection;

- (d) the extent of the overlap between the arguments made during examination and the manner in which Petitioner relies on the prior art or Patent Owner distinguishes the prior art;
- (e) whether Petitioner has pointed out sufficiently how the Examiner erred in its evaluation of the asserted prior art; and
- (f) the extent to which additional evidence and facts presented in the Petition warrant reconsideration of the prior art or arguments.

Becton, Dickinson, 17–18.

Advanced Bionics, LLC v. MED-EL Electromedizinische Geräte GmbH, IPR2019-01469, Paper 6 at 8, 10, 17-18 (PTAB Feb. 13, 2020) (precedential), sets out a two-part framework for analyzing these factors. In the first part, factors (a), (b), and (d) are considered to determine whether the art or arguments presented in the petition are the same or substantially the same as those previously presented to the Office. *Advanced Bionics*, 8-10. “If, after review of factors (a), (b), and (d), it is determined that the same or substantially the same art or arguments previously were presented to the Office,” then the second part of the analysis determines “whether the petitioner has demonstrated a material error by the Office” in view of factors (c), (e), and (f). *Id.* *Advanced Bionics* instructs that “[i]f reasonable minds can disagree regarding the purported treatment of the art or arguments, it cannot be said that the Office erred in a manner material to patentability.” *Id.* at 9.

With regard to the first part of the *Advanced Bionics* framework, Coffman and Kennewick were cited during examination and discussed by the Examiner. *See*

Ex. 1012, 152. In particular, the Examiner noted that these references do “not teach where an entry in a context stack (which includes a plurality of entries that are each indicative of context) that matches information in the merged transcription is identified (merged from the speech and non-speech-based transcription where the speech-based transcription is generated based on the cognitive model including information on one or more prior interactions).” *Id.* Despite the Examiner’s clear reasoning here, the Petition (incorrectly) relies on Coffman to disclose these features. Pet., 28-39.

Although the Petition attempts to rely on Ross in its obviousness combination (Pet., 39-47), Petitioner cites to Ross in the alternative (“To the extent there is any question whether Coffman alone discloses or renders obvious 1[f], it would have been obvious in view of Ross.”) Pet., 39 (emphasis omitted). In making this statement, Petitioner effectively admits that Ross’s substantive teachings are cumulative to what is found in Coffman and would not give rise to a material difference in the art.

Petitioner further suggests that “Ross makes clear, with specific examples, that it would have been obvious for Coffman’s system to identify a context entry that matches information in a transcription of user input.” Pet., 11. But as noted above, Ross (and Lee) do not cure the deficiency of Coffman and Kennewick with regard to the recitations concerning the “merged transcription” and other

deficiencies. Thus, even when Coffman and Kennewick are evaluated in view of Ross or Lee, there is no substantive challenge to what the Examiner found missing from the prior art. Similarly, with regard to Lee and Ground 2, Petitioner does not rely on Lee to satisfy the features absent from Coffman and Kennewick. Thus, the Board should proceed to step 2 of the *Advanced Bionics* framework.

For step 2 of the *Advanced Bionics* framework, Petitioner acknowledges that Coffman and Kennewick were evaluated during examination and discussed by the Examiner prior to allowing the claims of the '607 Patent. But Petitioner's citations of support for the "merged transcription" (Pet., 11) of Coffman do not demonstrate error by the Examiner. Rather, as noted above, the citations to 6:6-12 and 38:8-16 of Coffman are consistent with the inputs being separately added to entries in the context stack and not merged prior to identifying an entry in the context stack. It was this feature that the Examiner expressly found absent in Coffman and Kennewick. Therefore, there is no room for Petitioner to argue that the Examiner materially erred. Instead of pointing out sufficiently how the Examiner erred in its evaluation of the asserted prior art, the Petition asks the Board to second-guess the Examiner's reasoned analysis. The Board should decline Petitioner's request to reexamine Coffman and Kennewick against the '607 Patent and deny the Petition under § 325(d).

VI. CONCLUSION

For the reasons presented above, the Petition should be denied, and no *inter partes* review should be instituted.

Dated: July 16, 2024

/Wayne M. Helge/

Wayne M. Helge (Reg. No. 56,905)

James T. Wilson (Reg. No. 41,439)

NOSSAMAN, LLP

1401 New York Ave, NW

Suite 800

Washington, DC 20005

Telephone: (202) 887-1400

Email: whelge@nossaman.com

Email: jwilson@nossaman.com

Counsel for Patent Owner

CERTIFICATE OF WORD COUNT

The undersigned certifies that the foregoing PATENT OWNER'S PRELIMINARY RESPONSE complies with the type-volume limitation in 37 C.F.R. § 42.24(b)(1). According to the word-processing system's word count, the brief contains 9,918 words, excluding the parts of the brief exempted by 37 C.F.R. § 42.24(a).

By: /Wayne M. Helge/
Wayne M. Helge (Reg. No. 56,905)
Attorney for Patent Owner

CERTIFICATE OF SERVICE

I hereby certify that on July 16, 2024, a true and correct copy of the foregoing document was served via email, by consent, to Petitioner by serving the correspondence email addresses of record as follows:

Heidi L. Keefe
hkeefe@cooley.com

Andrew C. Mace
amace@cooley.com

Mark R. Weinstein
mweinstein@cooley.com

Reuben Chen
rchen@cooley.com

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Ave. NW, Suite 700
Washington D.C. 20004
Tel: (650) 842-7800

/Wayne M. Helge/
Wayne M. Helge (Reg. No. 56,905)
Attorney for Patent Owner