

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

META PLATFORMS, INC.,  
Petitioner,

v.

DIALECT, LLC,  
Patent Owner.

---

IPR2025-01333  
U.S. Patent No. 9,263,039

---

**PETITION FOR *INTER PARTES* REVIEW OF  
U.S. PATENT NO. 9,263,039**

**TABLE OF CONTENTS**

	<b>Page</b>
I. INTRODUCTION .....	1
II. MANDATORY NOTICES .....	1
A. Real Party-in-Interest .....	1
B. Related Matters.....	1
C. Lead and Back-Up Counsel and Service Information .....	2
III. PAYMENT OF FEES .....	3
IV. REQUIREMENTS FOR <i>INTER PARTES</i> REVIEW .....	3
A. Grounds for Standing .....	3
B. Identification of Challenged Claims .....	4
V. THE 039 PATENT .....	4
A. Effective Filing Date .....	4
B. Person of Ordinary Skill in the Art .....	4
C. The 039 Patent.....	4
VI. CLAIM CONSTRUCTION .....	6
VII. PRINCIPAL PRIOR ART.....	7
A. Summary of Maes .....	7
B. Summary of Ross .....	8
C. The Combination of Maes and Ross .....	9
VIII. THE CHALLENGED CLAIMS ARE UNPATENTABLE .....	21
A. Ground 1: Maes and Ross Render Obvious Claims 13-15, 17, and 18 .....	21
1. Claim 13 .....	21
a. [13.0] A method of processing speech and non- speech communications, comprising:.....	21
b. [13.1] receiving the speech and non-speech communications;.....	26

- c. [13.2] transcribing the speech and non-speech communications to create a speech-based textual message and a non-speech-based textual message;.....29
  - i. Transcribing the speech communications .....29
  - ii. Transcribing the non-speech communications.....36
- d. [13.3] merging the speech-based textual message and the non-speech-based textual message to generate a query;.....40
  - i. Merging the speech-based and non-speech-based textual messages.....40
  - ii. Merging generates a query .....45
- e. [13.4] searching the query for text combinations;.....47
- f. [13.5] comparing the text combinations to entries in a context description grammar; .....49
- g. [13.6] accessing a plurality of domain agents that are associated with the context description grammar;.....54
- h. [13.7] generating a relevance score based on results from comparing the text combinations to entries in the context description grammar; .....56
- i. [13.8] selecting one or more domain agents based on results from the relevance score; .....59
- j. [13.9] obtaining content that is gathered by the selected domain agents; and .....61
- k. [13.10] generating a response from the content, wherein the content is arranged in a selected order based on results from the relevance score. ....63
- 2. Claim 14.....65
  - a. [14.0] The method according to claim 13, further comprising generating an aggregate response that includes the content that is gathered by the selected domain agents. ....65
- 3. Claim 15.....67

- a. [15.0] The method according to claim 13, further comprising: receiving a follow-up speech and non-speech communications;.....67
- b. [15.1] transcribing the follow-up speech and non-speech communications to create a follow-up speech-based textual message and a follow-up non-speech-based textual message; and .....68
- c. [15.2] merging the follow-up speech-based textual message and the follow-up non-speech-based textual message to generate a follow-up query.....68
- 4. Claim 17.....69
  - a. [17.0] The method according to claim 13, further comprising generating a context stack that includes one or more contexts that are selected based on the query.....69
    - i. Generating a context stack including contexts....69
    - ii. Contexts selected based on the query.....71
- 5. Claim 18.....72
  - a. [18.0] The method according to claim 17, wherein the one or more contexts are generated based on applying prior probabilities or fuzzy possibilities to (i) keyword matching, (ii) user profiles, (iii) a dialog history, or any combination of (i) to (iii).....72
- IX. THE BOARD SHOULD NOT EXERCISE ITS DISCRETION AND DENY INSTITUTION .....74
- X. CONCLUSION.....74
- CERTIFICATE OF COMPLIANCE.....76
- CERTIFICATE OF SERVICE .....77
- CLAIM LISTING .....79

**TABLE OF AUTHORITIES**

	<b>Page(s)</b>
 <b>Cases</b>	
<i>Dialect, LLC v. Meta Platforms, Inc.</i> , 7:25-cv-00060 (W.D. Tex.) .....	1
<i>Dialect, LLC v. Microsoft Corporation</i> , 2-24-cv-01067 (EDTX) .....	2
<i>Google LLC v. Dialect, LLC</i> , IPR2024-00750 .....	2
<i>Google LLC v. Dialect, LLC</i> , IPR2024-00752 .....	2
<i>Google LLC v. Dialect, LLC</i> , IPR2024-00753 .....	2
<i>Microsoft Corporation v. Dialect, LLC</i> , IPR2025-00657 .....	1
 <b>Statutes</b>	
35 U.S.C. § 102(b) .....	8
35 U.S.C. § 102(e) .....	7
35 U.S.C. §103 .....	1, 4
 <b>Other Authorities</b>	
37 C.F.R. § 42.6(e).....	77
37 C.F.R. § 42.24 .....	76
37 C.F.R. § 42.100(b) .....	6

**EXHIBIT LIST**

<b>No.</b>	<b>Exhibit Description</b>
1001	U.S. Patent No. 9,263,039
1002	File History of U.S. Patent No. 9,263,039
1003	Declaration of Dr. Henry Houh
1004	CV of Dr. Henry Houh
1005	U.S. Patent No. 6,964,023 (“Maes”)
1006	RESERVED
1007	<i>Dialect LLC v. Bank of America, N.A.</i> , Case No. 2:24-cv-00207 (E.D. Tex.), Dkt. 66 (“Second Amended Complaint”)
1008	RESERVED
1009	RESERVED
1010	RESERVED
1011	RESERVED
1012	RESERVED
1013	RESERVED
1014	RESERVED
1015	RESERVED
1016	RESERVED
1017	RESERVED
1018	RESERVED
1019	RESERVED

1020	RESERVED
1021	D. Walters “Deterministic Context-Sensitive Languages: Part I*” (“Walters”), INFORMATION AND CONTROL 17, 14-40 (1970)
1022	U.S. Patent Application Publication No. 2002/0133354 (“ <u>Ross</u> ”)
1023	RESERVED
1024	RESERVED
1025	RESERVED
1026	Excerpts from Microsoft Computer Dictionary, 5 <sup>th</sup> edition (2002)
1027	Cattaneo, Marco EGV. “Fuzzy probabilities based on the likelihood function.” <i>Soft Methods for Handling Variability and Imprecision</i> . Springer Berlin Heidelberg, 2008.
1028	Shdaifat, I., Grigat, R.R. and Lütgert, S., 2001. Viseme recognition using multiple feature matching. In <i>INTERSPEECH</i> (pp. 2431-2434).
1029	EDTX Calendar, Judge Gilstrap
1030	RESERVED
1031	RESERVED
1032	RESERVED

**TABLE OF ABBREVIATIONS AND CONVENTIONS**

<b>Abbreviation</b>	<b>Meaning</b>
039 Patent	Ex.1001: U.S. Patent No. 9,263,039
IPR	<i>inter partes</i> review
Petitioner	Meta Platforms, Inc. (“Meta”)
Patent Owner or PO	Dialect, LLC (“Dialect”)
Second Amended Complaint	Compl.: Ex.1007
<i>xx:yy–zz</i>	column <i>xx</i> , lines <i>yy</i> to <i>zz</i>

## I. INTRODUCTION

Petitioner submits this Petition for *Inter Partes* Review of claims 13-15 and 17-18 (the “Challenged Claims”) of U.S. Patent No. 9,263,039 (the “039 Patent” (Ex.1000)), assigned to Patent Owner (“PO”). Petitioner respectfully submits that the Challenged Claims of the 039 Patent are unpatentable under 35 U.S.C. §103 in view of the prior art references discussed herein.

## II. MANDATORY NOTICES

### A. Real Party-in-Interest

The real party-in-interest is Meta Platforms, Inc.

### B. Related Matters

This is a copycat of the petition from *Microsoft Corporation v. Dialect, LLC*, IPR2025-00657 on the 039 Patent.

U.S. Patent Office records indicate that the 039 Patent is assigned to Dialect, LLC, which is currently asserting the 039 Patent in the following concurrent litigation filed on February 7, 2025: *Dialect, LLC v. Meta Platforms, Inc.*, 7:25-cv-00060 (W.D. Tex.).

Petitioner has filed, at substantially the same time that this Petition was filed, petitions for *inter partes* review against other patents asserted in the concurrent litigation: U.S. Patent No. 8,447,607, U.S. Patent No. 7,398,209, U.S. Patent No. 8,015,006, and U.S. Patent No. 9,734,825.

The 039 Patent is also asserted by PO in *Dialect, LLC v Bank of America, N.A.*, Eastern District of Texas No. 2:24-cv-00207-JRG (EDTX). In addition, the following cases involve PO and related patents: *Dialect, LLC v. Salesforce, Inc.*, 7:25-cv-00061, *Dialect, LLC v. Microsoft Corporation*, 2-24-cv-01067 (EDTX), *Google LLC v. Dialect, LLC*, IPR2024-00750 (PTAB), *Google LLC v. Dialect, LLC*, IPR2024-00752 (PTAB), *Google LLC v. Dialect, LLC*, IPR2024-00753 (PTAB), *Dialect, LLC v Bank of America, N.A.*, Eastern District of Texas No. 2:24-cv-00207-JRG (EDTX).

### C. Lead and Back-Up Counsel and Service Information

Lead Counsel	Lisa K. Nguyen (Reg. No. 58,018) Paul Hastings LLP 1117 S. California Avenue Palo Alto, CA 94303 Telephone: 650.320.1800 Fax: 650.320.1900 Email: PH-META-DialectIPR@paulhastings.com
Back-Up Counsel	David Tennant (Reg. No. 48,362) Paul Hastings LLP 2050 M St., N.W. Washington, DC, 20036 Telephone: 202.551.1700 Fax: 202.551.1705 Email: PH-META-DialectIPR@paulhastings.com
Back-Up Counsel	Diane Ghrist ( <i>pro hac vice</i> to be filed) Paul Hastings LLP 2050 M St., N.W. Washington, DC, 20036

	Telephone: 202.551.1700 Fax: 202.551.1705 Email: PH-META-DialectIPR@paulhastings.com
Back-Up Counsel	Rachel Wu Hankinson ( <i>pro hac vice</i> to be filed) Paul Hastings LLP 525 South Flower Street Twenty-Fifth Floor Los Angeles, CA 90071 Telephone: 213.683.6112 Fax: 213.627.0705 Email: PH-META-DialectIPR@paulhastings.com

Petitioner consents to electronic service by email at PH-META-DialectIPR@paulhastings.com and the e-mail addresses listed above.

### III. PAYMENT OF FEES

Petitioner authorizes the Office to charge the filing fee and any other necessary fee to Deposit Account No. 50-2613.

### IV. REQUIREMENTS FOR *INTER PARTES* REVIEW

#### A. Grounds for Standing

Petitioner certifies that the 039 Patent is available for inter partes review. Petitioner is not barred or estopped from requesting an inter partes review challenging the claims on the identified grounds herein. Petitioner has not filed a civil action challenging the validity of a claim of the 039 Patent. This petition is being filed no more than 1 year after the date on which Petitioner was served with a complaint alleging infringement of the 039 Patent.

**B. Identification of Challenged Claims**

**Ground 1:** Maes (Ex.1005) and Ross (Ex.1022) render obvious claims 13-15 and 17-18 under 35 U.S.C. § 103.

**V. THE 039 PATENT****A. Effective Filing Date**

Petitioner assumes for the purposes of this Petition that August 5, 2005 is the effective filing date.

**B. Person of Ordinary Skill in the Art**

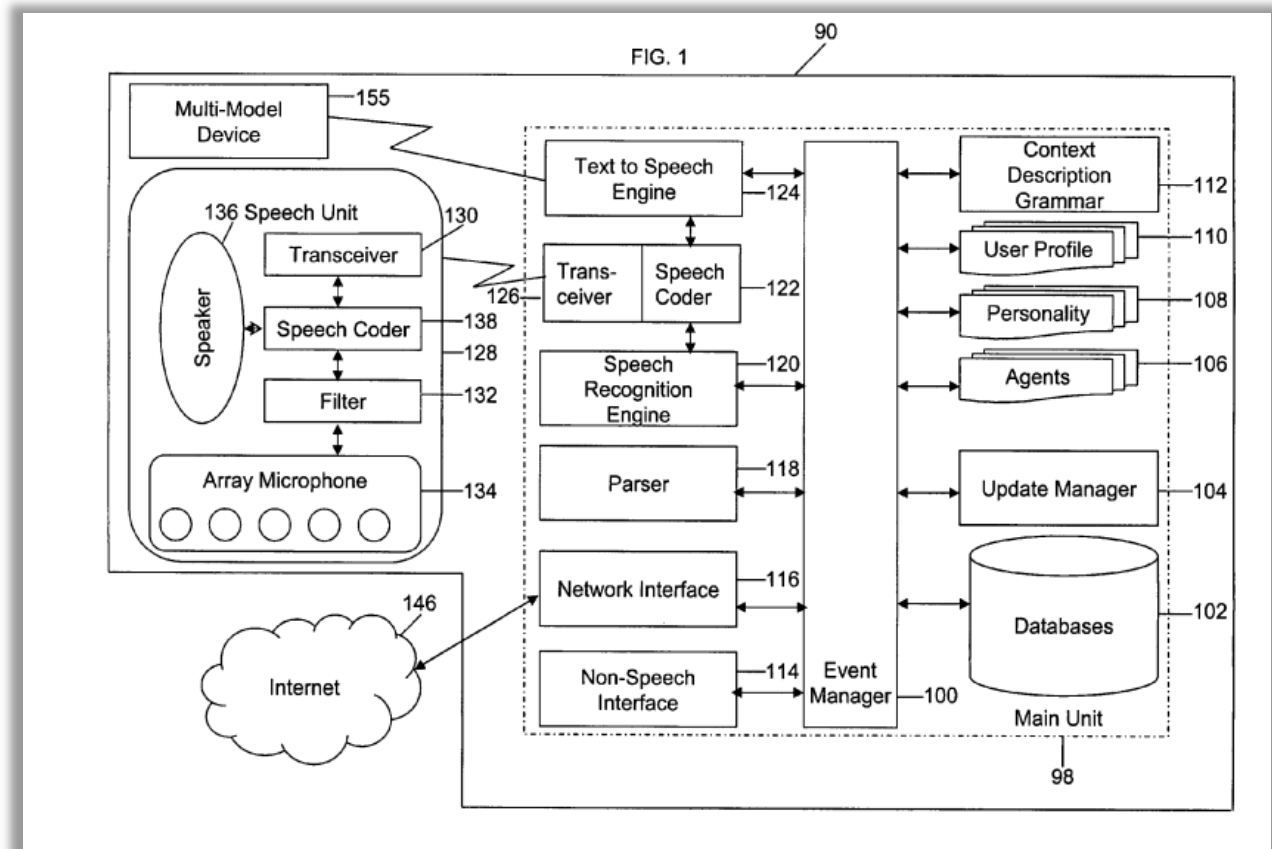
A POSITA with respect to the Asserted Patents as of the time of the invention, i.e., the earliest possible priority date of the 039 Patent, August 5, 2005, would have had a bachelor's degree in electrical engineering, computer science, computer engineering, or an equivalent, and two years of relevant experience involving computer science fundamentals, including natural language processing, speech recognition and transcription, non-speech recognition and transcription that is pertinent to the 039 Patent. Lack of professional experience can be remedied by additional education, and vice versa. Ex.1003, ¶25.

**C. The 039 Patent**

The 039 Patent explains that it “relates to retrieval of information or processing of commands through a speech interface and/or a combination of a speech interface and a non-speech interface.” Ex.1001, 1:25–35. The 039 Patent

states that it “creates, stores and uses extensive personal profile information for each user, thereby improving the reliability of determining the context of the speech and non-speech communications and presenting the expected results for a particular question or command.” Ex.1001, Abstract.

Figure 1 of the 039 Patent (below) “is an overall diagrammatic view according to one embodiment of the invention” (Ex.1001, 10:14–15):



Ex.1001, Figure 1.

The 039 Patent describes a system 90 that includes speech unit 128, speech recognition engine 120, context description grammar module 112, parser 118, and

agents 106, Ex.1001, 12:4–7, and after initial processing, the speech is passed to speech recognition engine 120 for processing using the context description grammar module 112. Ex.1001, 12:40–42. 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.” Ex.1001, 12:42–45. “The knowledge agents may then process the commands or questions.” Ex.1001, 12:49–50. To select the correct agent to process the command or question, “[t]he parser 118 uses a scoring system to determine the most likely context or domain for a user’s question and/or command. The score is determined from weighing a number of factors including, the user profile 110, the domain agent’s data content and previous context. Based on this scoring, the system 90 invokes the correct agent,” which returns results to the user. Ex.1001, 21:28–33. “The domain agent 156 scores the relevance of the results based on results already received, the context, the criteria, the history of the dialog, the user profile 110 and domain specific information using probabilistic or fuzzy scoring techniques. Part of the dialog history is maintained in a context stack.” Ex.1001, 23:19–25.

## **VI. CLAIM CONSTRUCTION**

Claims are given their “ordinary and customary meaning” as understood by a POSITA and the prosecution history pertaining to the patent. 37 C.F.R. § 42.100(b). Because a POSITA would find the challenged claims unpatentable under any

interpretation consistent with their plain and ordinary meaning in the context of the 039 Patent, the Board need not expressly construe the claim terms. *See Vivid Techs., Inc. v. Am. Sci. & Eng. 'g. Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

## VII. PRINCIPAL PRIOR ART

### A. Summary of Maes

U.S. Patent No. 6,964,023 to Maes et al. (Ex.1005) was filed on February 5, 2001, and issued on November 8, 2005. Therefore, Maes qualifies as prior under at least as 35 U.S.C. § 102(e).

Maes describes “systems and methods are provided for performing focus detection, referential ambiguity resolution and mood classification in accordance with multi-modal input data, in varying operating conditions, in order to provide an effective conversational computing environment for one or more users.” Ex.1005, Abstract.

Specifically, Maes explains that its system “receives multi-modal input in the form of audio input data, video input data, as well as other types of input data ..., processes the multi-modal data ..., and performs various recognition tasks (e.g., speech recognition, speaker recognition, gesture recognition, lip reading, face recognition ... in accordance with the recognition engines ...) ... using this processed data. The results of the recognition tasks and/or the processed data, itself, is then used to perform one or more conversational computing tasks, e.g., focus

detection, referential ambiguity resolution, and mood classification ... ." Ex.1005, 4:7–22.

### **B. Summary of Ross**

U.S. Pub. No. 2002/0133354 to Ross et al. (Ex.1022) was filed on August 16, 2001 and published on September 19, 2002. Therefore, Ross qualifies as prior under at least as 35 U.S.C. § 102(b).

Ross describes techniques for determining a context associated with a user's spoken command or question to determine an application to invoke to process the command or question. Ex.1022, [0010], [0013]. In Ross, determining a context involves evaluating the user's recognized spoken command against grammars for applications in which the grammars describe potential contexts (e.g., keywords or phrases) related to the utterance. Ex.1022, [0033]–[0034]. Ross explains that “a grammar is defined for each application 26” and describes two example grammars, one for an electronic mail application and another for a calendar application. Ex.1022, [0035], [0040], and [0046].

Ross describes how identifying context(s) involves matching transcribed text to the grammar expressions in the grammars: “If the speech center 20 hears a phrase such as ‘print the first message’ or ‘print the first appointment,’ the context manager 50 can readily figure out the intended target application 26 for the uttered sentence. Only one grammar will accept the phrase, which thus indicates the selected context

72 for that phrase and that associated application 26 is the one that should be targeted to receive the corresponding command.” Ex.1022, [0052]; *see also* [0053].

### **C. The Combination of Maes and Ross**

It would have been obvious to include Ross's teachings of context grammars (including data/entries in such grammars), maintaining a prioritized list of context grammars in a context list, and comparing a spoken utterance<sup>1</sup> (or, equivalently a representation of a spoken utterance) against data in context grammars to identify matches between keywords/phrases of a context grammar and a spoken utterance, generating relevance scores/weights based on such comparisons, identifying context grammars (and associated applications/application programs) that are capable of accepting a given utterance, determining whether a given context grammar is capable of accepting a given spoken utterance, using context grammars to formulate/construct requests sent to applications (Ex.1022, Abstract, [0002], [0013], [0021], [0033]-[0038], [0052]-[0053], [0059]-[0060], Figures 4, 6) in Maes's system that receives and processes spoken utterances for the reasons explained below. Ex.1003, ¶52.

---

<sup>1</sup> As used herein, it will be understood that a “spoken utterance” would encompass both an audio portion (e.g., speech) and a video (e.g., non-speech) portion (such as the accompanying lip movement) of the utterance. Ex.1005, 6:43–47.

Dr. Houh explains that the use of grammars and specifically, grammars that describe context for applications (i.e., context grammars) and used for comparing/matching decoded text from a user's spoken utterance was well-known in the art. For example, Ross explains that the grammars can be specified using the well-known "Backus Naur Form (BNF)." Ex.1022, [0013], [0060]; *see also* Ex.1026, 49 (describing dictionary definition for "BNF"); Ex.1021, 15–16 (providing definitions of context-sensitive and context-free grammars). Further, Ross's teachings of testing/comparing a recognized utterance against data included in a context list storing context grammars for applications (Ex.1022, [0013], [0034]-[0037], [0053], Figure 4) is strikingly similar to Maes's technique of associating the results of the recognized events included in the decoded text/script against data stored in an organized/sorted context stack. Ex.1005, 7:60–8:4, Figures 1, 8. Ex.1003, ¶53.

Further, Maes's "context stack" is similar to Ross's "context list." For example, Ross's description of maintaining a priority order in its context list in a manner such that the most-recently accessed application moves to the top of its context list (Ex.1022, [0037]) is consistent with the well-known concept of a "stack," in which the most recent item added to the stack will be the first item considered from the stack in a later operation and termed in the art as "last in, first out" (LIFO)

methodology of processing a queue. Ex.1026, 305 (dictionary definition for “last in, first out”); *compare with* 215 (dictionary definition for “first in, first out”). In view of the similarity of the techniques in Maes and Ross related to performing comparisons of spoken utterances against data stored in a context list/ stack, which were known in the prior art, Dr. Houh states that a POSITA would have understood that combining the system of Maes and with the context grammar techniques of Ross would improve the context identification abilities of Maes’s system (*see e.g.*, Ex.1005, 7:60–8:4, 8:37–42, and 37:53–55), and therefore provide at least one motivation to combine these references. Ex.1003, ¶53.

Both Maes and Ross are analogous art to the 039 Patent for at least three reasons. *First*, both references are directed toward the same field of endeavor as the 039 Patent—computer-implemented systems interpreting user utterances. Ex.1001, Abstract, 3:17–37; Ex.1005, Abstract, 4:7–17; and Ex.1022, Abstract, [0021]. *See also* Ex.1005, 7:60–8:4; Ex.1022, [0053]. *Second*, both Maes and Ross are reasonably pertinent to a problem described by the 039 Patent: an environment for reliably processing a user’s language queries, which is a problem of conventional systems identified in the 911 Patent. Ex.1001, 1:55-61. Maes describes its invention as “provid[ing] an effective conversational computing environment for one or more users” based on processing a user’s multi-modal input in the form of audio input data

(e.g., spoken utterance), video input data (e.g., lip movement and/or visual gesture), as well as other types of input data. Ex.1005, Abstract, 4:7–17, 6:39–55. Ross describes a technique for determining a context associated with a user’s recognized utterance that involves evaluating grammars against the user’s recognized utterance. Ex.1022, [0033]-[0034], Figure 5. *Third*, similar to the 039 Patent, both Maes and Ross teach the use of *grammars* and *vocabularies* in recognizing speech inputs. Ex.1001, 13:52–55; Ex.1005, 31:15–18, 34:61–67, 33:11–21, 39:59, 41:13–20, 41:35–38; Ex.1022, [0033]-[0035], Figure 4, [0028]. Ex.1003, ¶54.

A POSITA would have been motivated to combine the teachings of Maes and Ross as described above because, as Dr. Houh describes, Maes solves the problems specifically identified by Ross. Specifically, Ross describes the problem associated with conventional systems stating that different speech-enabled applications “may not share the same speech enabling interfaces” and “cannot make a determination which application should receive a particular speech utterance.” Ex.1022, [0004].. For example, if a user “speaks a calendar command to set up an appointment that is received by a word processing application, then the user can experience an unexpected or undesirable result,” which can lead to “wasted time and effort, as well as frustration.” Ex.1022, [0004]. Maes solves these problems because Maes’s system includes use of “[an] I/O subsystem 12 ... compris[ing] one or more

microphones for capturing audio input data from the environment in which the system is deployed” and based on processing the captured audio input data, Maes’s system “determines ... which application(s) [*e.g.*, speech-enabled applications] should handle the user inputs.” Ex.1005, 6:5–8 and 36:54–57. Thus, by employing a single I/O subsystem, for instance, “across all registered applications,” Maes’s system addresses Ross’s concern that in the conventional practice, “individual, independent speech-enabled applications ... each contend[] for control of the microphone, and each oblivious to the other.” Ex.1005, 36:49–54; Ex.1022, [0005]. As such, a POSITA would have understood Maes’s teachings to expand or otherwise improve the capabilities of Ross, and vice-versa. Ex.1003, ¶55.

Dr. Houh further states that a POSITA would have recognized that Ross’s teachings related to “resolv[ing] ... ambiguous phrase[s]” such as “print this” or “print it” in spoken utterances would be complimentary to Maes’s system, which expressly seeks to provide ambiguity resolution, if needed, by “seek[ing] confirmation, disambiguation, correction, more details ... until the intent is unambiguous and fully determined.” Ex.1022, [0059]; Ex.1005, 36:59–63, 5:52–56 and Figure 2; *see also* 8:30–42 (describing an example of an in-vehicle system performing ambiguity resolution in connection with the potentially ambiguous spoken utterance “turn it on” because it identifies that there are likely other devices

in the vehicle that could be turned on). Furthermore, Ross describes maintaining a priority list of context grammars used in resolving ambiguities in a spoken utterance and identifying a target application for the spoken utterance. Ex.1022, [0035], [0053], [0059]. “When a successful match is found [based on testing the recognition messages against the active grammars], **the corresponding translation 74 is dispatched to the script engine 38 for execution, and the priority of the matching grammar (i.e., selected context 72) is raised [within a prioritized list of grammars in context list 62].**” Ex.1022, [0034]-[0035]; *see also* Ex.1022, [0037]. Thus, a POSITA would have been motivated to combine Maes and Ross because Ross’s teachings of priority list of context grammars when combined with Maes’s system would benefit Maes’s system by providing it with ambiguity resolution functionality. Ex.1022, [0059]; Ex.1005, 36:59–63, 5:52–56. Ex.1003, ¶56.

Maes also provides the function of determining a target application or appliance addressed by the user’s spoken utterance. Ex.1005, 2:54–56. Maes explains “determin[ing] and execut[ing] one or more application programs that effectuate the user’s intention and/or react to the user activity. The application depends on the environment that the system is deployed in.” Ex.1005, 7:40–46. Accordingly, both Maes and Ross are directed at determining the target application

to handle the task corresponding to the user's intention and/or react to the user activity based on processing/decoding a user's spoken utterance. Ex.1005, 2:54–56 and Ex.1022, [0013]. It would have been obvious to a POSITA to modify Maes's teachings of generalized representations of data (or, simply queries) generated as a result of merging the transcriptions (Ex.1005, 6:32–50) to additionally include Ross's teachings of context grammars so that the queries are compared against data in context grammars for that target application, per Ross's teachings. Ex.1022, [0034]-[0037]. Ex.1003, ¶57.

Indeed, Ross expressly teaches that “a grammar is defined for each application.” Ex.1022, [0035]. For example, in the combined Maes/Ross system, applications (“*domain agents*”) that are considered as target application candidates would be accessed, as taught by Ross, to ascertain whether their associated context grammars are capable of accepting a recognized representation of the processed/decoded spoken utterance. Ex.1022, [0052]-[0053], [0013]. One benefit (among others) of incorporating Ross's teachings of context grammars in Maes's system is that it would expand the vocabulary (i.e., more number of keywords and phrases) of Maes's system, which therefore would result in interpreting spoken utterances in a manner that is “closer in meaning” to the user inputs. A POSITA would have thus recognized that Ross's teachings of context grammars (and their

use in testing against decoded text/script corresponding to the user's spoken utterance) are compatible with Maes's system in many ways, and therefore would have had a reasonable expectation of success for combining them without undue experimentation. For instance, the "applications" in Ross and "grammar[s] defined for each application" are "speech-enabled" and Maes is directed at speech processing methods and systems. Ex.1022, [0035]-[0038]; Ex.1005, 4:7-16, Figures 1, 4. Ex.1003, ¶58.

Additionally, Dr. Houh states that a POSITA would have had a reasonable expectation of success in implementing Ross's teachings with Maes's system, because doing so would involve minimal changes to Maes's system architecture and the combined system uses the hardware and techniques already described in each reference, and each system would perform the same in the combined system. For instance, a POSITA would recognize that Ross's teachings related to use of context grammars for applications, e.g., maintaining a list of context grammars according to a certain priority would merely be a few additional process steps for "grammars" or "grammar database" used in Maes's system. Ex.1005, 31:15-18, 34:61-67, 33:11-21, 39:59, 41:13-20, and 41:35-38; and Ex.1022, [0033]-[0035], Figure 4, and [0028]. And, therefore, in one implementation, a POSITA would have been motivated to modify Maes's "grammar" / "grammar database" to additionally

include Ross's teachings of context grammars (e.g., each context grammar comprising entries such as keywords, phrases, and operators used in an application) yielding Maes's modified "grammar" / "grammar database." And in another implementation, for instance, a POSITA would have understood to modify Maes's context stack (Ex.1005, 37:53–55, 7:60–8:4, 8:37–42, Figure 1) to additionally include Ross's context list (Ex.1022, [0013], [0034]-[0035], [0052]-[0059], Figure 4) comprising context grammars (such as 70-1, 70-2, 70-3). Thus, Ross's context grammars (each including appropriate keywords and phrases used in an application) would be implemented within Maes's context stack yielding Maes's modified context stack having improved context identification functionality. Ex.1003, ¶59.

Moreover, in yet another example implementation, a POSITA would have understood to incorporate Ross's teachings related to context grammars for applications as an additional stand-alone database/module within Maes's system. A POSITA would have further understood that any of the above implementations would have yielded a Maes/Ross system in which the results of the recognized input/output (I/O) events produced as a result of processing the user's spoken utterance using Maes's system would be compared, per Ross, against data (e.g., keywords/phrases used in applications) included in context grammars (for instance, stored within Maes's modified grammar database, or within Maes's modified context

stack) and the outcome of prior comparisons would be used to generate a recency of access characteristic (e.g., “*relevance score*”) used for maintaining the priority order of Ross’s context list. Ex.1002, [0034]-[0035], [0053]-[0054]. Such access characteristics would allow tracking (“*selecting*”) recently accessed applications and obtaining relevant content from the recently accessed applications, in the Maes/Ross system. Ex.1003, ¶60.

A POSITA would have also understood that the Maes/Ross system would have predictably resulted in Maes’s dialog manager additionally performing the step of testing the decoded text/script processed from the spoken utterance against entries in context grammars, per Ross. Ex.1022, [0013], [0034]-[0035]. Ross explains a spoken phrase (e.g., a voice command or other speech input) from a user sent to a microphone connected to “the computer system,” such as for example, Maes’s system (which includes the dialog manager). Ex.1022, [0021]; Ex.1005, Figure 1. Maes already describes its dialog manager associating the results of the recognized events included in the decoded text/script (produced as a result of the transcription) against data stored on the context stack. Ex.1005, 37:53–55, 7:60–8:4, 8:37–42, Figure 1. Adding the extra step of testing the decoded text/script against entries in Maes’s and/or Ross’s grammar would improve the dialog manager’s ability to identify contexts. A POSITA would have the ability to make the combination cited

above and would have further understood these options as obvious-to-try and therefore had good reason to pursue them. Ex.1003, ¶61.

A POSITA would have been motivated to combine Maes and Ross because it would have involved the application of known techniques (e.g., identifying contexts) to improve a similar system in the same way. Maes already describes using a context stack “organized/sorted context corresponding to each active dialog” for storing “all the information associated with an application.” Ex.1005, 37:55–56, 37:60-61. Similar to Maes’s context stack (Ex.1005, context stack 817 in Figure 8 or context stack 20 in Figure 1), Ross uses a “context list” (Ex.1022, context list 62 in Figure 4). For example, Ross teaches “maintaining an ordered list of applications” in context list 62 such that “[w]henever an application 26 gains window focus, it will move to the head of the list 62. Likewise, whenever an application 26 which is not the top priority application 26 is chosen as the target for a speech command, the application 26 indicated by the selected context 72 will move to the head of the list 62. In this way, the last application 26 that the user touched or talked to will get the first opportunity at interpreting the next user utterance, and the other applications 26 will be ordered in a most-recently-accessed way.” Ex.1022, [0035]. A POSITA would have recognized that Ross’s technique of maintaining a priority list of context grammars for applications in a manner that results in the most-recently accessed

application to move to the top of the list would greatly benefit Maes's system because it would allow efficient identification of context by retrieving the most-recently accessed application in the modified context stack of the combined Maes and Ross system. Ex.1003, ¶62.

Furthermore, a POSITA would have been motivated to combine Maes and Ross because the references make clear that their systems require no specialized hardware or software. In fact, these references teach using conventional, commercially-available systems. Ex.1005, 2:38–53 (describing Maes's system comprises at least one processor and memory operatively coupled to the at least one processor), 45:50–60 (describing that “the elements illustrated in FIGS. 1 through 9C may be implemented in various forms of hardware, software, or combinations thereof”); 12:43–49 (disclosing that “the processing performed in blocks 414 and 416 may be accomplished via any **conventional acoustic information recognition system** capable of extracting and labeling acoustic feature vectors”); Ex.1022, [0002] (describing commercially-available speech recognition products that convert speech into text strings that can be utilized by software applications on a computer system), [0025] (describing use of Microsoft® Active Accessibility® (MSAA) from Microsoft Corporation, Redmond, Wash). Ex.1003, ¶63.

A POSITA would be motivated to combine Maes with Ross because to do so would have been the arrangement of old elements (a speech transcription and recognition system comprising modules for speech transcription and recognition, speech-enabled applications handling a user's spoken utterance input, context grammars comprising keywords/phrases pertaining to applications, context stack storing current and historical data related to user interactions) with each performing the same function it has been known to perform (e.g., recognizing spoken utterances, converting spoken utterances into computer-readable format based on decoding the utterances, matching the decoded text of the utterance against grammars describing potential contexts such as keywords or phrases included in the utterance) and yielding no more than what one would expect from such an arrangement (an improved computer-implemented system interpreting user utterances), as Maes demonstrates. Ex.1005, 2:54–67. Ex.1003, ¶64.

## VIII. THE CHALLENGED CLAIMS ARE UNPATENTABLE

### A. Ground 1: Maes and Ross Render Obvious Claims 13-15, 17, and 18

#### 1. Claim 13

##### a. [13.0] A method of processing speech and non-speech communications, comprising:

Maes describes “[s]ystems and methods are provided for performing focus detection, referential ambiguity resolution and mood classification **in accordance**

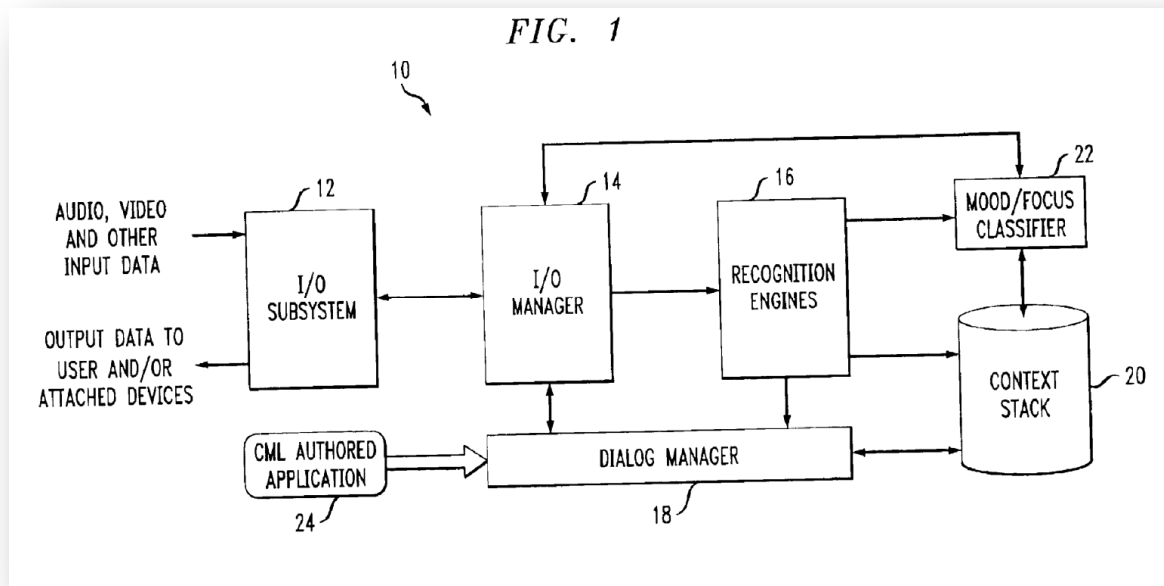
**with multi-modal input data**” and, more particularly, to “systems and methods for performing focus detection, referential ambiguity resolution and mood classification in accordance **with multi-modal input data**. ... Systems that employ such ‘multi-modal’ input techniques have inherent advantages over systems that use only one data input mode.” Ex.1005, Abstract and 1:9–23; Ex.1003, ¶66.

Further, Maes’s systems and methods are directed at “*processing speech and non-speech communications*.” Referencing Figure 1, Maes discloses:

**Generally, the multi-modal conversational computing system ... of the present invention receives multi-modal input in the form of audio input data, video input data, as well as other types of input data (in accordance with the I/O subsystem 12), processes the multi-modal data (in accordance with the I/O manager 14), and performs various recognition tasks (e.g., speech recognition, speaker recognition, gesture recognition, lip reading, face recognition, etc., in accordance with the recognition engines 16), if necessary, using this processed data. The results of the recognition tasks and/or the processed data, itself, is then used to perform one or more conversational computing tasks, e.g., focus detection, referential ambiguity resolution, and mood classification (in**

accordance with the dialog manager 18, the context stack 20 and/or the classifier 22).

Ex.1005, 3:66–4:21.



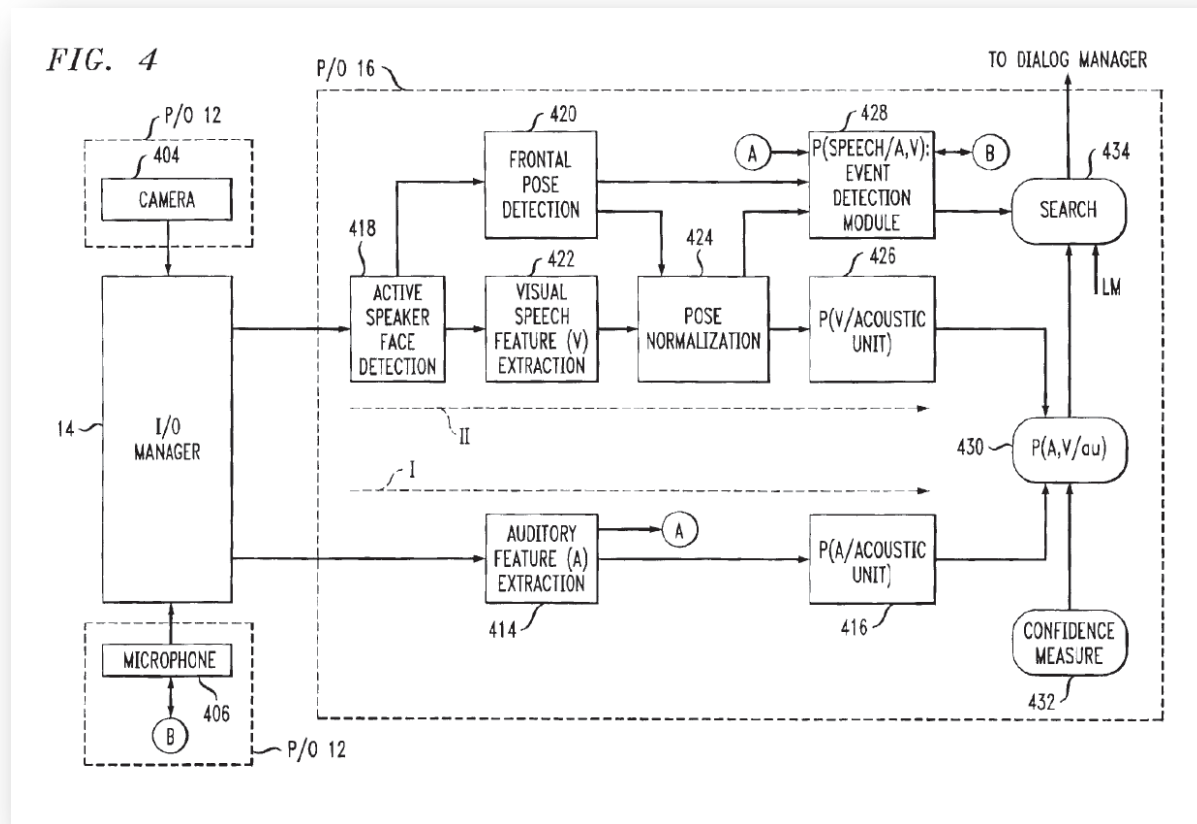
Ex.1005, Figure 1; Ex.1003, ¶67.

Maes describes Figure 4 as “an audio-visual speech recognition module<sup>2</sup> that may be employed as one of the recognition modules<sup>3</sup> of FIG. 1 **to perform speech recognition using multi-modal input data received** in accordance with the invention.” Ex.1005, 10:50–52; Ex.1003, ¶68.

<sup>2</sup> As used herein, the terms “Maes’s system” and “audio-visual speech recognition module” are used interchangeably.

<sup>3</sup> Maes uses the term “block” and “module” interchangeably. For example, see 12:43-38 and 17:46-50 referring to “414” as “block[] 414” and “module 414.”

Specifically, Figure 4 shows that the audio-visual speech recognition module receives multi-modal input data including an audio signal, e.g., speech provided by the speaker and/or background noise) (“*speech*”) and a video signal (e.g., the speaker’s face including lip movement and/or background objects in the environment) (“*non-speech communications*”) and processes the multi-modal input data (e.g., “*speech and non-speech communications*”). Video signals are processed by blocks/modules 418, 422, 424, 426 and audio signals are processed by blocks/modules 414, 416.



Ex.1005, Figure 4; Ex.1003, ¶68.

Therefore, Maes discloses “[a] method of processing speech and non-speech communications.” Ex.1003, ¶69.

Maes describes examples in which its methods are applicable to a broadcast news system, which further confirms that its method applies to “speech and non-speech communications.” For example, Maes states that “the audio-visual speech recognition module ... process[es] arbitrary content video [such as] in the context of broadcast news ... [which] contain a newsperson speaking at a location where there

is arbitrary activity and noise in the background.” Ex.1005, 10:63–11:15; Ex.1003, ¶72

As another example, Maes states that “the multi-modal conversational computing system ... may be employed within a vehicle” in which a user may say the “spoken utterance ‘turn it on’” and Maes’s system processes inputs relating to an I/O event representative of the user’s spoken utterance (audio portion) and the accompanying lip movement of the spoken utterance (video portion). Ex.1005, 4:30–32, 6:43–50, and 7:63–8:29; Ex.1003, ¶73.

**b. [13.1] receiving the speech and non-speech communications;**

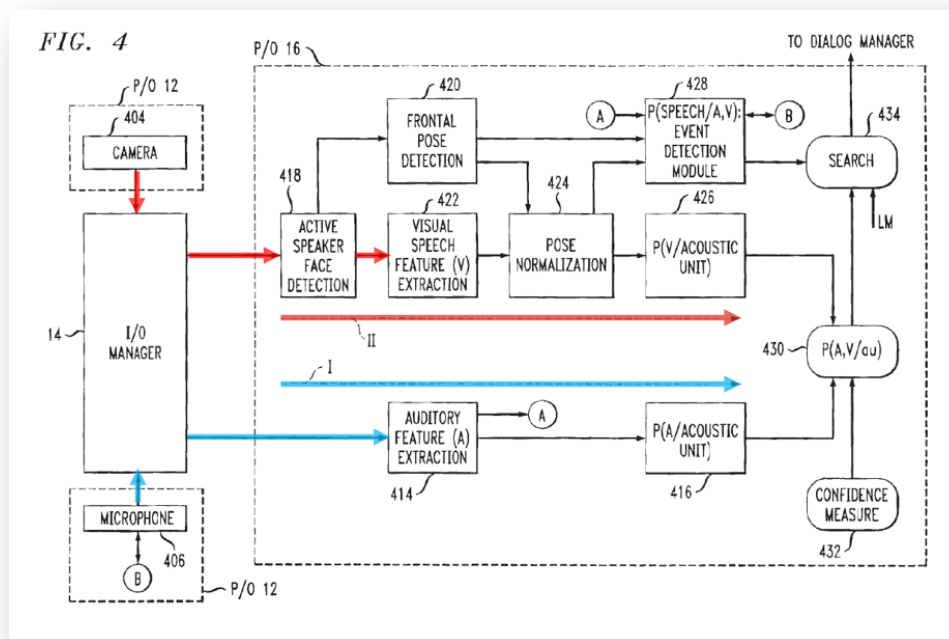
For example, Maes describes Figure 4 as “an audio-visual speech recognition module that may be employed as one of the recognition modules of FIG. 1 **to perform speech recognition using multi-modal input data received** in accordance with the invention.” Ex.1005, 10:50–52; Ex.1003, ¶76.

Specifically, Maes’s Figure 4 discloses “an audio-visual speech recognition module” that “receive[s]” (“*receiving*”) multi-modal input data—such as an audio signal and a video signal (“*the speech and non-speech communications*”) at an “input/output (I/O) subsystem ... compris[ing] one or more microphones for capturing audio input data,” which “one of ordinary skill in the art will realize that

other user interfaces and devices ... may be included **for capturing user activity.**”

Ex.1005, 10:47–11:23, 6:21–29, and 6:5–8; Ex.1003, ¶76.

Maes explains that the “multi-modal input data” includes “an audio signal” (e.g., speech provided by the speaker and/or background noise) (“*speech*”) “and” “a video signal” (e.g., the speaker’s face including lip movement and/or background objects in the environment) (“*non-speech communications*”). Ex.1005, 11:8–63. Maes’s system “receives multi-modal input in the form of **audio input data, video input data, as well as other types of input data ....**” Ex.1005, 4:7–15; Ex.1005, 4:67–5:5 (“system 10 ... **taking both audio input data and image data input and processing it**”).



Ex.1005, Figure 4 (annotated to show processing paths for audio (path I) and video (path II); Ex.1005, 11:55–63; Ex.1003, ¶77.

Maes explains the feature extractor 414 of the audio-visual speech recognition module “**receives an audio or speech signal**” and the active speaker face detection module 418 “**receives video input [from] camera 404**” (e.g., “the visual portion (e.g., lip movement) of the utterance,” (Ex.1005, 6:46-47)). Ex.1005, 11:64–12:2, 12:49–61; Ex.1003, ¶¶78–79.

Dr. Houh explains that Maes describes the “audio signal” (“*speech*”) can include “silence,” “speech,” and “noise” and the “video signal” (“*non-speech communications*”) comprising “the visual portion (e.g., lip movement) of the utterance.” Ex.1005, 13:3–15 and 6:46–47; Ex.1003, ¶80.

For example, the audio signal (“*speech*”) is received using microphones for capturing audio input data and the video signal (“*non-speech communications*”) is received using cameras or sensors for capturing video input data. Ex.1003, ¶¶80, 83; *see also* ¶81 (explaining multi-modality).

Additionally, with reference to the Figure 1<sup>4</sup>, Maes discloses that “the multi-modal conversational computing system 10 of the present invention **receives multi-modal input in the form of audio input data, video input data, as well as other types of input data (in accordance with the I/O subsystem 12).**” Ex.1003, ¶82.

Referencing the in-vehicle system example, Maes discloses the “*receiving ...*” step: Maes’s system receives multi-modal inputs relating to an I/O event representative of the user’s spoken utterance “turn it on” (“*speech*”) and lip movement corresponding to the spoken utterance (“*non-speech communications*”) such that “the microphone picks up the audible portion of the utterance and a camera picks up the visual portion (e.g., lip movement) of the utterance.” Ex.1005, 6:43–50. *See generally* Ex.1005, 6:15–28, 6:35–38, 7:63–67; Ex.1003, ¶84.

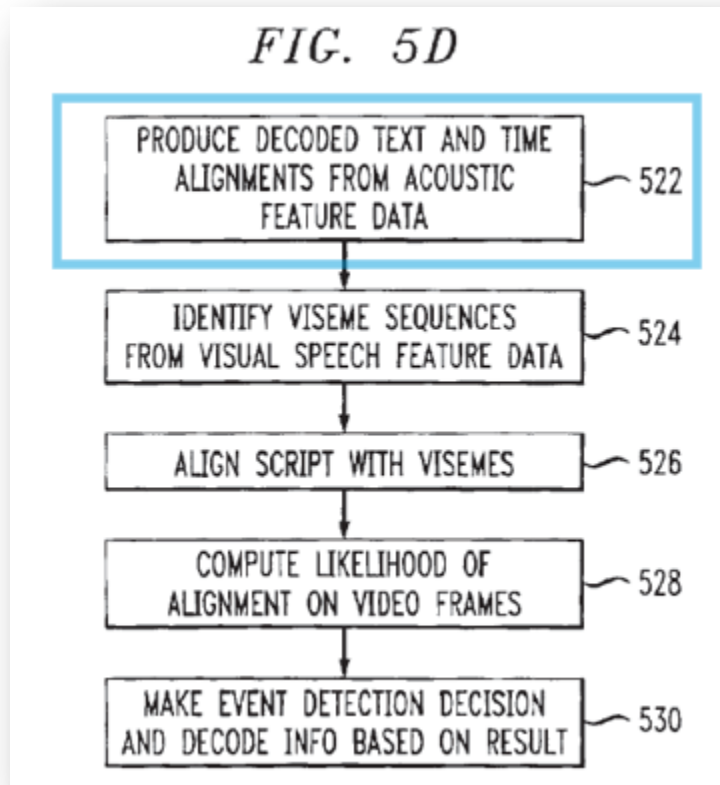
- c. **[13.2] transcribing the speech and non-speech communications to create a speech-based textual message and a non-speech-based textual message;**
  - i. **Transcribing the speech communications**

With reference to Figure 5D (reproduced below), Maes discloses classical speech recognition techniques processing the audio signal including the spoken

---

<sup>4</sup> Maes identifies Figure 4 as “a preferred embodiment of an audio-visual speech recognition module” that may be employed as one of the recognition modules of the Figure 1 embodiment. Ex.1005, 10:47–52.

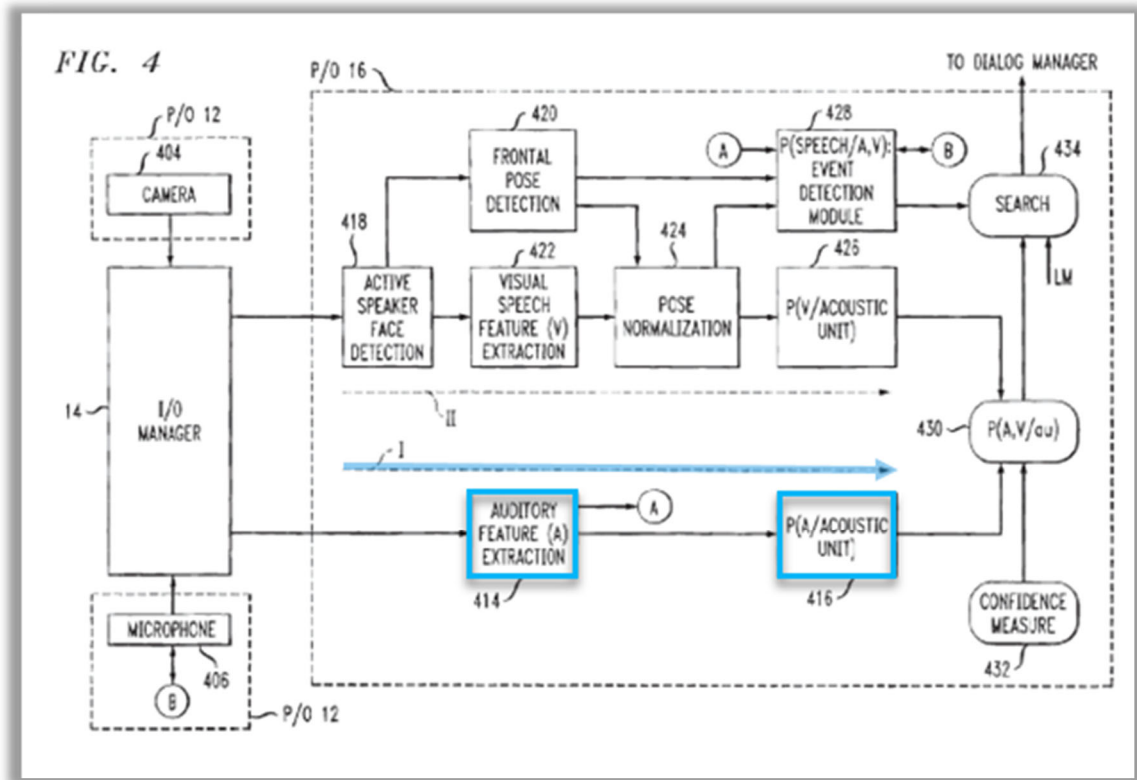
utterance (“*transcribing the speech ... communications*”) “to” “produce” (“*create*”) “decoded text” (or, script) (“*a speech-based textual message*”) “using the feature data from the acoustic feature extractor 414.” See generally Ex.1005, 21:43–47.



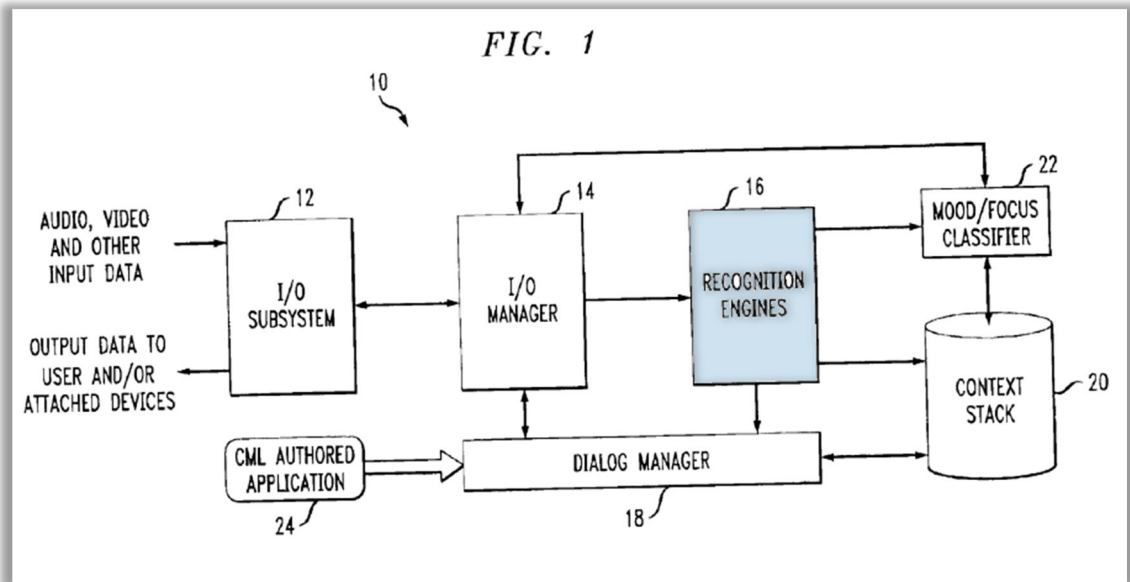
Ex.1005, Figure 5D (block 522). “[I]n step 522, the uttered speech to be verified may be decoded by classical speech recognition techniques so that a decoded script and associated time alignments are available. This is accomplished using the feature data from the acoustic feature extractor 414.” Ex.1005, 21:43–47. Dr. Houh explains that a POSITA would understand that the “decoded script” described in Maes is essentially the tangible result of a speech recognition system’s interpretation

of spoken language and provides a text representation (“*a speech-based textual message*”) of the multi-modal content including the audio signal further comprising the spoken utterance / speech. E.g., Ex.1005, 20:61-64 (“If the buffered data is tagged as speech, in step 518, the buffered data is sent on through the acoustic path so that the buffered data may be recognized, in step 520, so as to yield a **decoded output.**”); *see also* 20:23-33; Ex.1003, ¶88.

Maes provides additional details explaining the “*transcribing ...*” step: Maes describes Figure 4 as “an audio-visual speech recognition module that may be employed as one of the recognition modules of FIG. 1 to perform speech recognition using [received] multi-modal input data.” Ex.1005, 10:50–52. For example, the auditory feature extractor 414 in Figure 4 “receives an audio or speech signal and ... extracts spectral features from the signal at regular intervals[,] ... in the form of acoustic feature vectors (signals) which are then passed on to a probability module 416.” Ex.1005, 11:64–12:2; Ex.1003, ¶89.



Ex.1005, Figure 4 (showing audio information processing path (Path I) annotated in blue and acoustic feature vectors denoted by the letter “A”) and 11:55–64.



Ex.1005, Figure 1 (annotated); Ex.1003, ¶90.

Maes explains how acoustic features are extracted:

magnitudes of discrete Fourier transforms of samples of speech data in a frame are considered in a logarithmically warped frequency scale. Next, these amplitude values themselves are transformed to a logarithmic scale. The latter two steps are motivated by a logarithmic sensitivity of human hearing to frequency and amplitude. Subsequently, a rotation in the form of discrete cosine transform is applied. One way to capture the dynamics is to use the delta (first-difference) and the delta-delta (second-order differences) information.

Ex.1005, 12:12–28; ; Ex.1003, ¶91.

As Dr. Houh explains, Maes describes sampling the signal prior to extracting the acoustic feature vectors. Ex.1003, ¶92.

Maes provides several examples of acoustic feature vectors, such as [linear prediction coefficients] LPC cepstra, [Perceptual Linear Prediction] PLP, and MEL cepstra. Ex.1003, ¶93.

Maes states that “the processing performed in blocks 414 and 416 may be accomplished via any **conventional acoustic information recognition system** capable of extracting and labeling acoustic feature vectors.” Ex.1005, 12:43–49. One such “acoustic information recognition system” that provides additional details on how speech is processed in Maes is described with reference to Figure 9A, explained below. Ex.1003, ¶94.

Referencing Figures 1 and 9A, Maes provides details of the “*transcribing ...*” step explaining that “via the I/O manager 14 of FIG. 1” “user-provided input data events are ... provided to” “apparatus 900”—an “apparatus for collecting data associated with a voice of a user,” including “a dialog management unit 902 for conduct[ing] a conversation with the user,” that provides functionality including natural language understanding (NLU), natural language generation (NLG), finite state grammar (FSG), and/or text-to-speech Syntheses (TTS) for machine-prompting the user....” Ex.1005, 39:22–29, 41:13–20, and Figure 9A; Ex.1003, ¶95.

Maes continues: “Apparatus 900 ... includes a processing module 910” further including “**a speech recognizer 926** which ... include[s] **a speech recognition module 928**,” and “a speech prototype, language model and grammar database”. Ex.1005, 39:59, 41:35–38. “Apparatus 900 can further include **a post processor 938 ... configured to transcribe user utterances and ... perform keyword spotting thereon.** ... Post processor 938 can employ speech recognizer 926” and “can also include a semantic module (not shown) to interpret meaning of phrases. The semantic module could be used by speech recognizer 926 to indicate that some decoding candidates in a list are meaningless and should be discarded/replaced with meaningful candidates.” Ex.1005, 41:48-64; Ex.1003, ¶96.

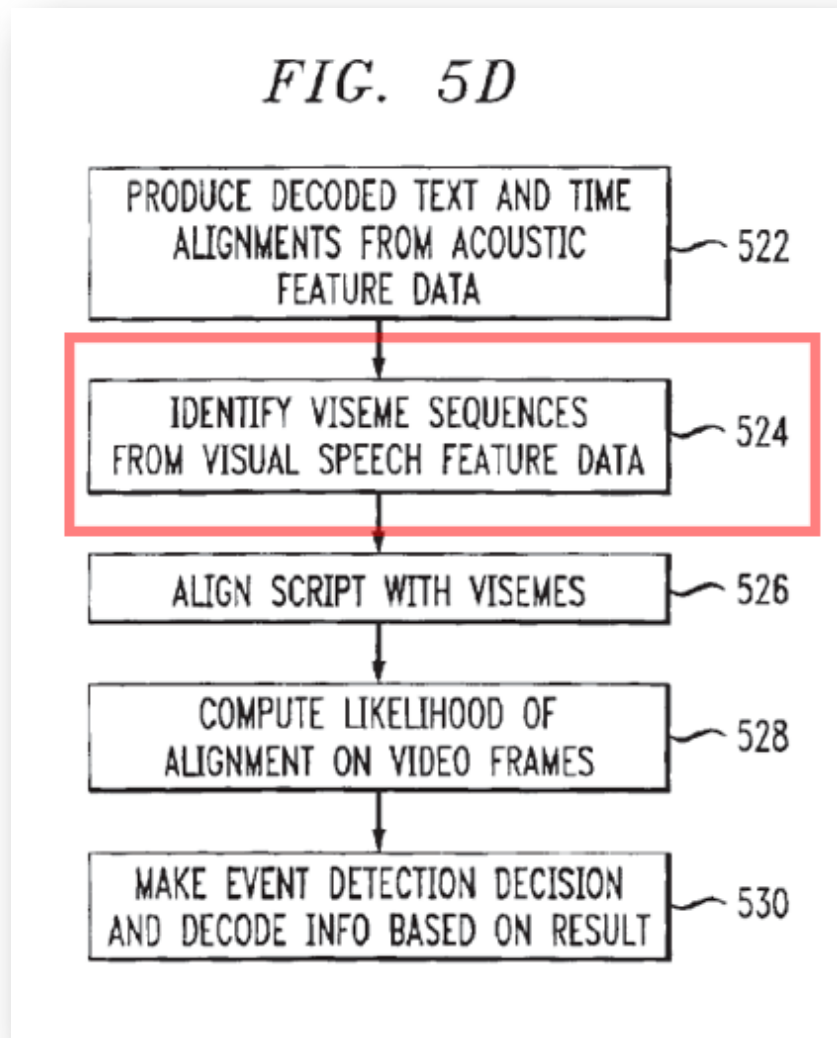
Maes explains that “the processing performed in blocks 414 and 416” in Figure 4 “produces” (“*creat[es]*”), for instance, “**decoded text**” (“*a speech-based textual message*”) (Ex.1005, 21:43–47), and further, referencing Figure 9, Maes explains that the “semantic module” functions to “indicate that some **decoding candidates** in a list are meaningless and should be discarded/replaced with meaningful candidates” (Ex.1005, 41:61–64). Thus, Maes explains that the “**decoded text**” produced by the processing performed in blocks 414 and 416 is the same “decoding candidates” Maes describes in connection with Figure 4, further confirming Figure 9A and its corresponding disclosure provide additional details to,

among other things, the system described in Figure 4. For example, in Figure 9A, Maes identifies “semantic module,” “speech recognizer 926,” “post processor 938,” “speech recognition module 928” and other components/functionalities. Ex.1003, ¶97.

Referencing the in-vehicle system example, Maes explains the “*transcribing ...*” step: the I/O manager abstracts the data into a form that represents ... a spoken utterance ... [A] data abstraction operation may involve generalizing details associated with all or portions of the input data so as to yield a more generalized representation of the data for use in further operations.” Ex.1005, 6:21–38. Thus, as a result of processing by several modules in Maes’s system, the system produces decoded text/script corresponding to the user’s spoken utterance, for example, of saying “turn it on.” Ex.1003, ¶99.

## ii. Transcribing the non-speech communications

For example, Maes Figure 4 discloses a visual speech feature extractor 422 that “extracts visual speech feature vectors (e.g., mouth or lip-related parameters) by processing the video signal (“*transcribing the ... non-speech communications*”), e.g., the face detected in the video frame including the user’s lip movement “*to*” “produce” (“*create*”) “a visual phonemes (visemes) sequence” (“*a non-speech-based textual message*”). Ex.1005, 17:51–55, 21:47–50. Maes discloses the “*transcribing ...*” step as a processing step in block 524 in Figure 5D.



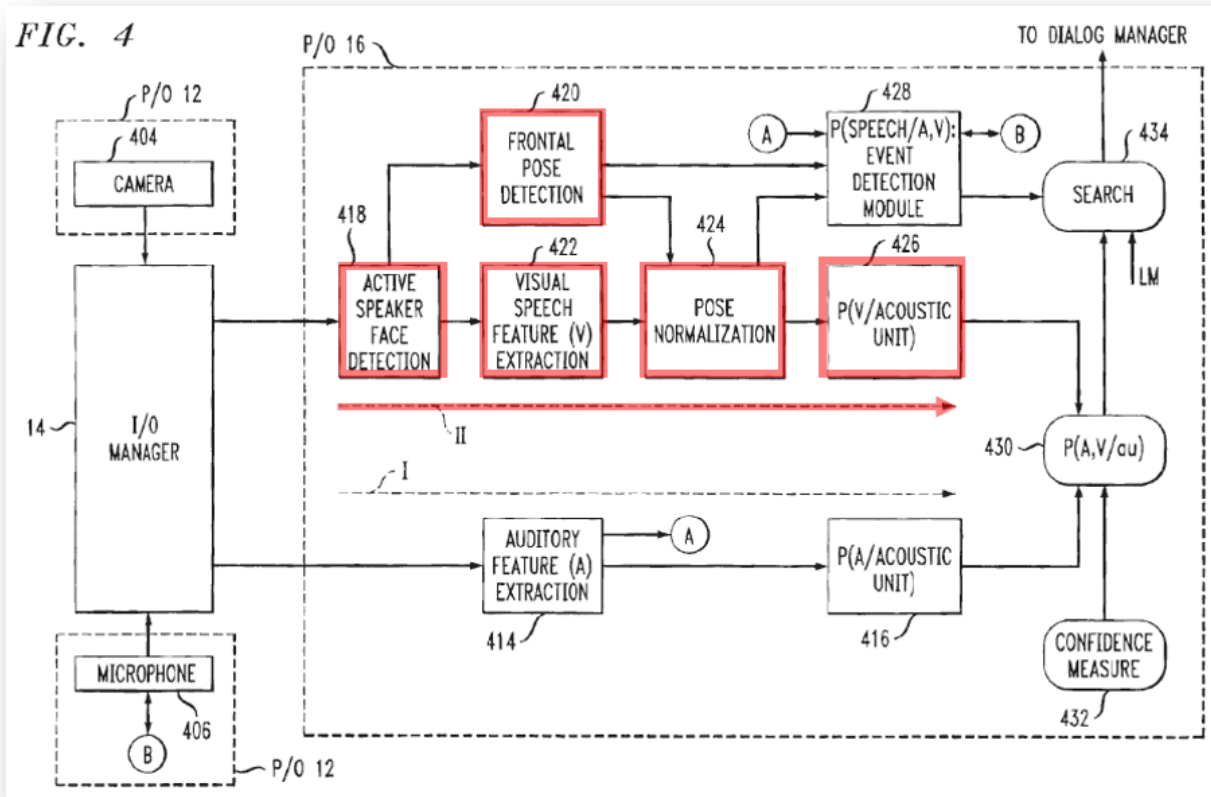
Ex.1005, Figure 5D (annotated); Ex.1003, ¶101.

Maes further explains modules 422, 424 and 426 in Figure 4 extracting, normalizing, and labeling (“*transcribing*”) to create visemes, e.g., in a visemes sequence (“*a non-speech-based textual message*”) using the extracted visual speech vectors.

**The extracted visual speech feature vectors [produced by visual speech feature extractor 422] are then normalized in block 424** with respect to the frontal pose estimates generated by the detection module 420. The normalized visual speech feature vectors are then provided to a probability module 426. ... [Additionally], **the probability module 426 labels the extracted visual speech vectors** with one or more previously stored phonemes. ... Alternatively, the visual speech feature vectors may be labeled with visemes which, as previously mentioned, are visual phonemes or canonical mouth shapes that accompany speech utterances.

Ex.1005, 18:42–63; Ex.1003, ¶102.

Visemes, or visual phonemes, in Maes, are “generally canonical mouth shapes that accompany speech utterances which are categorized and pre-stored similar to acoustic phonemes.” Ex.1005, 29:27–30. Dr. Houh explains that a visemes sequence (“*a non-speech-based textual message*”) generated in Maes is essentially a string of visemes that is the tangible result of a visual recognition system’s interpretation of a user’s lip movement accompanying a spoken utterance and provides a visual representation of the non-speech content. Ex.1003, ¶103 (citing Ex.1028, 1, 2, Figures 1, 5).



Ex.1005, Figure 4 (showing video information processing path (Path I) annotated in red and visual speech feature vectors denoted by the letter “V”), 23:1–5; Ex.1003, ¶104.

Dr. Houh explains that Maes provides several examples of visual speech features and describes multiple ways of extracting visual speech features. Ex.1003, ¶¶105–106 (citing Ex.1005, 17:55–63, 18:1–20, and 21:43–50).

Referencing the in-vehicle system example, Maes describes the “*transcribing*” step: “[t]he I/O manager [in Maes’s system] receives the raw multi-modal data and **abstracts the data into a form that represents ... a spoken**

utterance. As is known, **a data abstraction operation may involve generalizing details associated with all or portions of the input data so as to yield a more generalized representation of the data for use in further operations.**” Ex.1005, 6:21–38; Ex.1003, ¶108.

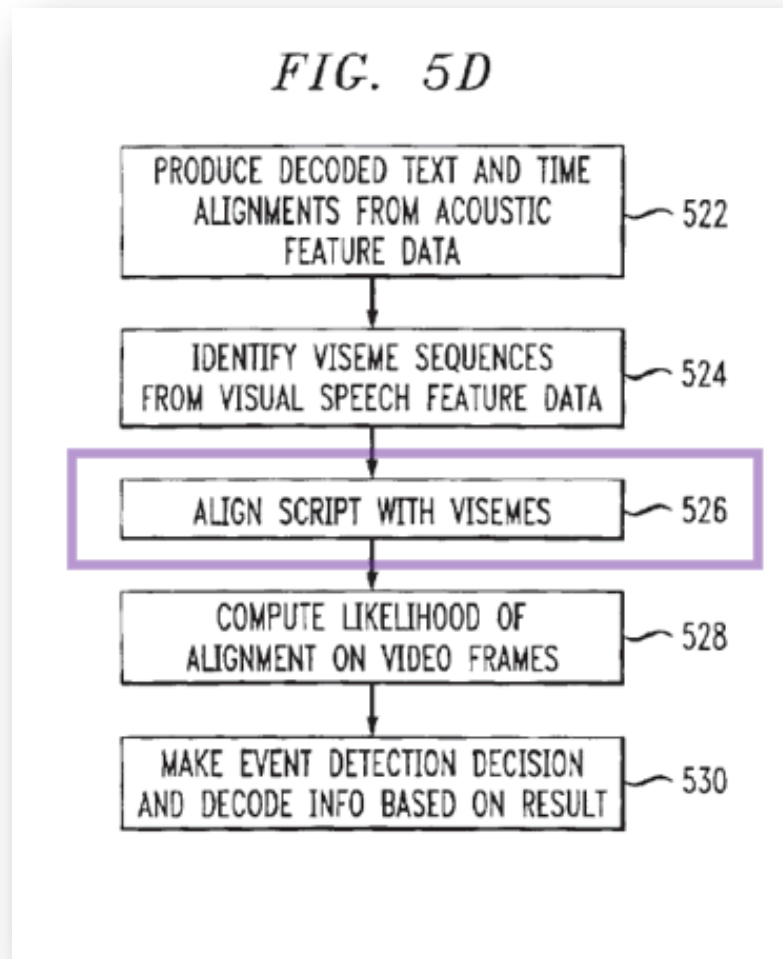
- d. **[13.3] merging the speech-based textual message and the non-speech-based textual message to generate a query;**
  - i. **Merging the speech-based and non-speech-based textual messages**

Maes discloses “the visual phonemes (visemes) sequence” (“*the non-speech-based textual message*”) and “decoded text” (or, script) (“*the speech-based textual message*”). See [13.2]; Ex.1003, ¶112.

Referencing step 526 of Figure 5D, Maes discloses:

the script is aligned with the visemes. A rapid (or other) alignment may be performed in a conventional manner in order to attempt to **synchronize the two information streams [i.e, the audio and video information streams]**.

Ex.1005, 21:51–54.



Ex.1005, Figure 5D (block 526); Ex.1003, ¶113.

Maes explains the purpose of aligning/synchronizing the audio and video information streams. “A goal associated with utterance verification is to make a determination that the speech used to verify the speaker in the audio path I and the visual cues used to verify the speaker in the video path II correlate or align. This allows the system to be confident that the speech data that is being used to recognize

the speaker is actually what the speaker uttered.” Ex.1005, 29:30–36; Ex.1003, ¶114.

Further, as Dr. Houh explains, Maes describes that aligning/synchronizing the audio and video information streams has “many advantages,” e.g., detecting errors in decoding and others. Ex.1003, ¶115 (citing Ex.1005, 29:37–43).

Therefore, by teaching synchronization of the two information streams (the audio and video information streams) or, specifically, alignment of “the decoded text” (or, script) (“*the speech-based textual message*”) with “the visual phonemes (visemes) sequence” (“*the non-speech-based textual message*”), Maes discloses the “*merging*” step. Ex.1003, ¶116.

Referencing the in-vehicle system example, Maes discloses:

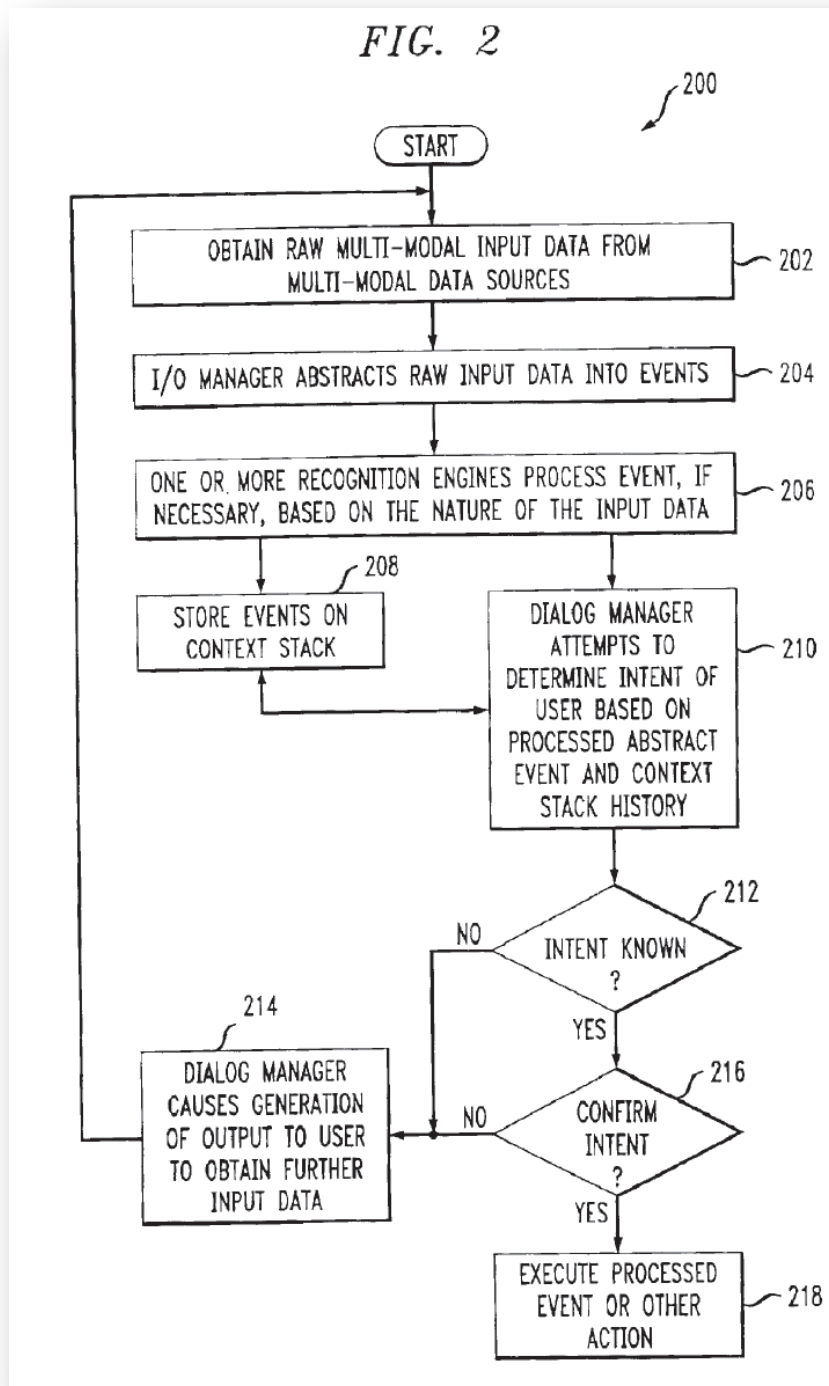
The dialog manager would ... receive **the results of the recognized events associated with the spoken utterance “turn it on” and the gesture of pointing to the radio.** Based on these events, the dialog manager does a search of the existing applications, transactions or “dialogs,” or portions thereof [stored on the context stack], with which such an utterance [including the accompanying lip movement] and gesture could be associated.

Ex.1005, 7:65–8:4, 8:37–42. In this example, the aligned decoded script and visemes sequence created from merging the transcriptions includes the results of the

recognized events associated with the spoken utterance “turn it on” (with the accompanying lip movement), therefore, constitutes the outcome of the “*merging* ...” step; Ex.1003, ¶117.

Indeed, with regards to module 430 in Figure 4, Maes identifies “the **first joint use of the visual information and audio information** in module 430” and “an explanation of how the two types of information [i.e, the audio information included in the decoded text/script and video information included in the viseme sequence] are combined to provide improved recognition accuracy.” Ex.1005, 20:5–7, 11:55–63. In view of these teachings, a POSITA would have understood that module 430 provides the “*merging*” step and the outcome of this step produces the aligned decoded script and visemes sequence (i.e., a merged transcription), e.g., associated with the spoken utterance “turn it on” with the accompanying lip movement. Ex.1003, ¶118.

Maes explains of how the audio and video signals are synchronized in the “*merging*” step. For example, Figure 2, below, is described in Maes as depicting “a flow diagram illustrating a referential ambiguity resolution methodology performed by a multi-modal conversational computing system.” Ex.1005, 3:24–28.



Ex.1005, Figure 2. In step 206, “the abstracted data [representing one or more events] ... is ... sent by the I/O manager ... to one or more recognition engines ...

to have the event recognized ... . That is, depending on the nature of the event, one or more recognition engines may be used to recognize the event. For example, ... **the event may be sent to an audio-visual speech recognition engine to have the utterance recognized using both the audio input and the video input associated with the speech.**” Ex.1005, 6:38–50; *see also* 5:57–6:8 and 6:28–35; Ex.1003, ¶119.

In Maes, one of the ways in which the audio-visual speech recognition engines processes or otherwise, recognizes event data is by generating an outcome in which the transcribed audio information such as the decoded text/script is aligned or synchronized with the transcribed video information such as the visemes sequence (*i.e.*, the outcome of the “*merging ...*” step). *See also* Ex.1005, 21:50–54 (“**[T]he script is aligned with the visemes.** A rapid (or other) alignment may be performed in a conventional manner in order to attempt to **synchronize the two information streams.**”), 21:59–61 (“a likelihood on the alignment is computed to determine how well **the script aligns to the visual data**”), 7:22–32 (**two, more or even all of the input modes described herein may be synchronized**” via the techniques disclosed in U.S. App. No. 09/507,526) Ex.1003, ¶120.

**ii. Merging generates a query**

Maes teaches:

[A] data abstraction operation [e.g., comprising the “*merging ...*”] may involve generalizing details associated with all or portions of the input data ... to yield [“*generate*”] a more generalized representation of the data [“*a query*”] for use in further operations.

Ex.1005, 6:32–38; Ex.1005, 4:6–22. Therefore, the generalized representation of the input data (including the spoken utterance and accompanying lip movement) is a computerized representation of the user’s query that is generated as the tangible outcome of an abstraction operation (e.g., comprising the “*merging ...*” step). *Id.* When an abstract event occurs, Maes’s system determines the target of the event, such as by performing the “*merging ...*” step and then “**launches the action associated to the user’s query.**” Ex.1005, 36:59–37:3, *see also* 7:60–67; Ex.1003, ¶123.

Indeed Maes’s system “*generate[s] a query*” because it:

provides the capability to: (i) determine an object, application or appliance addressed by the user; ... (iii) **understand queries** based on who said or did what, what was the focus of the user when he gave a multi-modal **query/command** ....

Ex.1005, 2:54–67. Therefore, Maes discloses that the “*merging*” is “*to generate a query.*” Ex.1003, ¶124.

e. [13.4] searching the query for text combinations;

For example, Maes teaches:

[A] data abstraction operation may involve generalizing details associated with all or portions of the input data ... to yield **a more generalized representation of the data [e.g., “*the query*”] for use in further operations [e.g., *searching*].**

Ex.1005, 6:35–38; Ex.1003, ¶126.

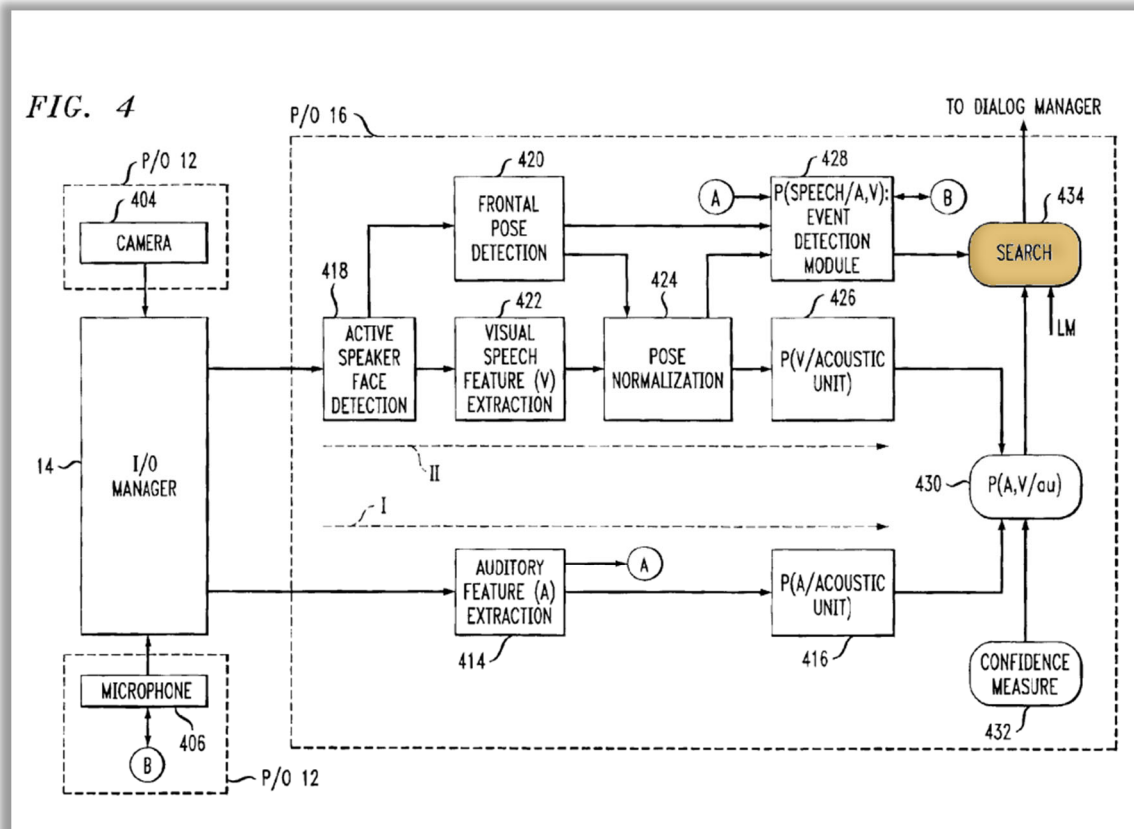
Maes provides additional details of the “*searching*” step. Referencing Figure 4, Maes discloses “**a search is performed in search module<sup>5</sup> 434 [“*searching*”] with language models (LM) ... [and] the acoustic units ... **representing what was uttered ... are put together to form words**[, which] are output by the search engine 434 as the decoded system output. A conventional search engine may be employed. This output is provided to the dialog manager 18 of FIG. 1 for use in disambiguating the user’s intent.” Ex.1005, 20:11–20; Ex.1003, ¶127.**

In Maes, therefore, the search engine 434 operates on the generalized representation of the data (“*the query*”), which represents what was uttered “*for*” providing (as output) words put together from acoustic units (“*text combinations*”).

---

<sup>5</sup> Maes uses the terms “search module 434” and “search engine 434” interchangeably. Ex.1005, 20:11 and 20:16–17.

Dr. Houh explains a conventional search engine (such as search engine 434) operates on queries, or otherwise generalized representations of data, to provide outputs.



Ex.1005, Figure 4 (annotated); Ex.1003, ¶128.

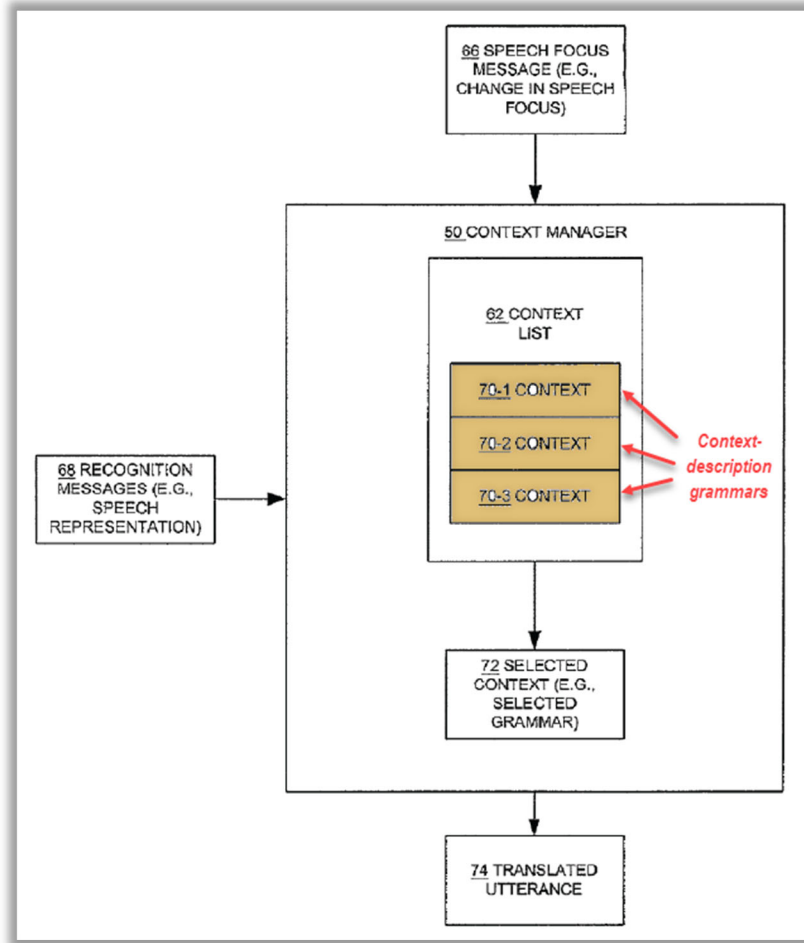
f. **[13.5] comparing the text combinations to entries in a context description grammar;**

As Dr. Houh explains, Maes generally describes use of “grammar” and “grammar database” in its system. Ex.1003, ¶¶131–134 (citing Ex.1005, 31:15–18, 34:61–67 (“data needed by any recognition engine (e.g., grammar, ... ), 33:11-21 (“us[e] the appropriate data files ... (e.g., contexts, finite state grammars, vocabularies ...)), 41:13–20 and 39:22–29 (dialog management unit 902 can “include ... finite state grammar (FSG) ... for machine-prompting the user”), 39:59, 41:35–38 (“a speech recognizer ... which can ... include ... grammar database 930”)).

Maes does not expressly disclose “*comparing the text combinations to entries in a context description grammar,*” but Ross does. Ex.1003, ¶135.

First, Ross teaches “*entries in [a] context description grammar.*” For example, referencing Figure 4 (reproduced below), Ross discloses “**context list includes contexts 70 (e.g., 70-1, 70-1, 70-3, etc.) for speech-enabled applications 26, which represent the grammars for the applications 26.**” Ex.1022, [0033]. Because Ross describes contexts 70-1, 70-1, 70-3 represent the grammars for the applications, Ross’s contexts representing grammars (such as 70-1, 70-1, 70-3) constitute “*context description grammar[s]*” describing contexts for three speech-enabled applications. For instance, context 70-1 is a grammar for a first speech-

enabled application, context 70-1 is a grammar for a second speech-enabled application, and context 70-3 is a grammar for a third speech-enabled application.



Ex.1022, Figure 4 (annotated); Ex.1003, ¶136.

Ross provides examples demonstrating “entries” included in a “context description grammar.” In Ross, “a grammar is defined for each application 26” and describes an example of selection between two grammars that serve as the contexts representing the grammars (“context description grammar[s]”) for two applications: one for an electronic mail application:

```
<mail> = do I have any messages |  
  open <message> |  
  create a message |  
  send <message>|  
  print <message>.  
<message> = the? <nth> message | it | this.  
<nth> = first | second | third | fourth | fifth | . . .
```

(Ex.1022, [0035], [0040]-[0041]) and another for a calendar application:

```
<appointment> = do I have any appointments |  
  open <appointment> |  
  create an appointment |  
  print <appointment>.  
<appointment> = the? <nth> appointment | it | this.
```

(Ex.1022, [0046]-[0047]). The above-mentioned grammars comprise “*entries*.” Specifically, in the grammar for the electronic mail application shown above, the line “<message> = the? <nth> message | it | this,” defines a rule named “message” which includes phrases (e.g., “the,” “message”), a reference to another rule (“<nth>”) and a grammar operator (“?”), which are examples of “*entries*.” Similarly, in the grammar for the calendar application shown above, phrases (e.g., “the,”

“appointment”), a reference to another rule (“<nth>”) and a grammar operator (“?”) are examples of “*entries*.” Ex.1003, ¶137.

Ross provides examples explaining how the entries (the phrases, keywords, and operators) included in a grammar are used. With respect to the grammar for the electronic mail application, Ross describes allowing a user’s spoken phrases (e.g., processed to generate “*the text combinations*”) such as “open the first message,” “create a message,” “send this,” and “print it” to be “match[ed]” (“*compar[ed]*”) against entries in the grammar for the electronic mail application (“*entries in a context description grammar*”). The grammar for the calendar application allows “match[ing]” (“*comparing*”) of spoken phrases such as “open the first appointment,” “create an appointment,” “print the fourth appointment,” and “print it” against entries in the grammar for the calendar application. Ex.1022, [0041]-[0051]; Ex.1003, ¶138.

Ross describes the “*comparing*” step:

Recognition messages 68 from the speech engine interface 30 are tested by the context manager 50 against the active grammars in the context list 62 in priority order. When a successful match is found, ... the priority of the matching grammar (i.e., selected context 72) is raised.

Ex.1022, [0034]. Ross notes that **“when an utterance is recognized, it will be tested against each application’s grammar to see if the grammar will accept it.”**

Ex.1022, [0035]. Furthermore, Ross explains “maintain[ing] the priority and state of the various grammars in the context list 62 in the system” so that “recognition messages 68 from the speech engine interface 30 are tested by the context manager 50 against the active grammars in the context list 62 in priority order.”

Ex.1022, [0034]. Ross further explains that “prior to evaluating the contexts,” (the “*comparing ...*” step), its system “create[s] the contexts for the speech enabled applications in the speech enabled environment.” Ex.1022, [0010]; Ex.1003, ¶139.

Dr. Houh explains that Ross describes one goal of matching (“*comparing*”) a recognized utterance (e.g., “*the text combinations*”) against data included in context grammars (“*entries in a context description grammar*”) is to find matches between the utterance and such grammars. Ex.1003, ¶140 (citing Ex.1022, [0034]). Therefore, Ross “uses a grammar to identify which speech enabled application is to receive the representation of the spoken utterance” and specifically, Ross **“uses the grammar to determine if a representation of a spoken utterance from a user is acceptable to (can be processed by) a particular speech-enabled application.”** Ex.1022, [0013]; Ex.1003, ¶140.

As explained above, *see* § VII.C, it would have been obvious to a POSITA to modify Maes's teachings of generalized representations of data (or, simply queries/user queries) generated as a result of merging the transcriptions (Ex.1005, 6:32–50) to additionally include Ross's teachings such that queries generated using Maes's system would be compared against data (e.g., keywords/phrases used in applications) included in context grammars, per Ross. Ex.1003, ¶141.

**g. [13.6] accessing a plurality of domain agents that are associated with the context description grammar;**

The 039 Patent describes “*domain agents*” as used for organizing “generic and domain specific behavior and information .... The domain agents provide complete, convenient and re-distributable packages or modules for each application area.” Ex.1001, 14:40–44. Thus, “*domain agents*” in the 039 Patent broadly refer to software modules that are specific to each application area. Ex.1003, ¶144.

Ross teaches multiple speech-enabled applications, such as word processing application, mail application, spreadsheet application, calendar application, and stock monitoring application running on a computer system in a multi-context speech enabled environment, which serve as examples of “*domain agents*.” Ex.1022, [0004], [0007]–[0008], [00010]; Ex.1003, ¶145.

Ross describes an example of selection between two applications: an electronic mail application and a calendar application (“*a plurality of domain*

*agents*”) which are candidate applications “targeted to receive” the user’s spoken utterance “print it.”

If the sentence is “print it however, **both grammars are capable of accepting the utterance**. The context manager 50 therefore has to make a choice by referring to the context list 62 of applications in order of recency of access. The context manager 50 **tests the utterance against these grammars** (indicated by the contexts 70 in the context list 62) in priority order, and **passes the commands on to the first application 26** [“*accessing a plurality of domain agents*”] **that has a grammar that will accept the phrase** [“*associated with the context description grammar*”].

Ex.1022, [0053]; *see also* [0045], [0051] (confirming “print it” phrase in grammars for both applications); Ex.1003, ¶146.

Therefore, Ross “uses a grammar to identify which speech enabled application is to receive the representation of the spoken utterance” and specifically, if such an utterance “is acceptable to (can be processed by) a particular speech-enabled application.” Ex.1022, [0013]; *see also* [0052] (“If both of these grammars were loaded into the context manager 50, the speech center system 20 is listening for any of the phrases accepted by either grammar. ... Only one grammar will accept the phrase, which thus indicates the selected context 72 for that phrase and that

associated application 26 is the one that should be targeted to receive the corresponding command.”). Ex.1003, ¶147.

**h. [13.7] generating a relevance score based on results from comparing the text combinations to entries in the context description grammar;**

Ross describes its system using an “access characteristic” (“*generating a relevance score*”) “*based on*” “recency of relevant access to the context [by] determin[ing] the context 70 for a speech enabled application 26 as indicated by the speech focus message 66,” which serve as describing “*results from comparing the text combinations to entries in the context description grammar.*” Ex.1022, [0012], [0036]; Ex.1003, ¶150.

Ross explains how its recency of access characteristic (“*relevance score*”) is used. For example, Ross states that the priority order of context grammars in its context list (such as context list 62) is maintained based on the recency of access characteristic (“*relevance score*”) of context grammars in a context list.

The context manager 50 maintains the priority and state of the various grammars in the context list 62 in the system.

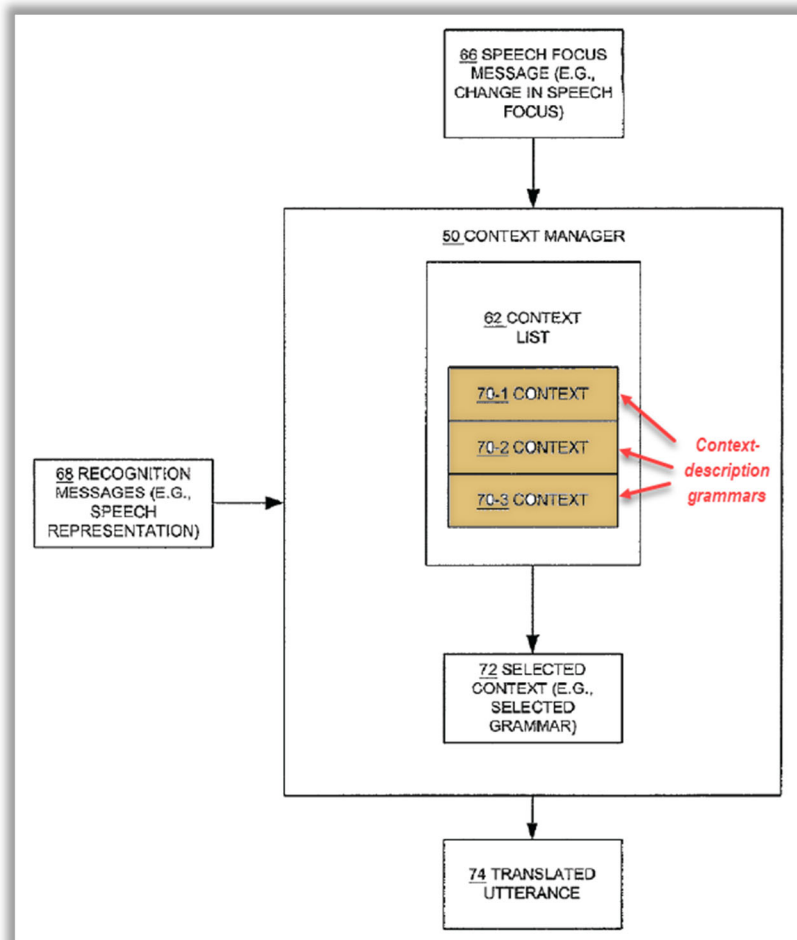
Ex.1022, [0034]; Ex.1003, ¶151.

The context manager 50 tests the utterance against these grammars (indicated by the contexts 70 in the context list 62) in priority order, and passes the commands on to the first application 26 that has a grammar that will accept the phrase.

The context list 62 of applications is reordered every time a recognition is directed to an application 26 other than the highest priority one, or whenever another application 26 gets windows focus (for example, because it was clicked on) .... The result is that the system 20 is biased to send the commands to the most recently accessed application 26.

Ex.1022, [0053]-[0054]. Thus, the application that gets window focus is considered to be highly relevant in Ross and accordingly given highest priority. Ex.1003, ¶151.

In Figure 4 (below), Ross illustrates the priority order of the context grammars in the context list:



Ex.1022, Figure 4 (annotated). For example, within context list 62, in priority order, 70-1 (appearing at the top of the context list) is the most-recently accessed grammar having the highest recency of access characteristic (highest “*relevance score*”); 70-2 (appearing in the middle of the context list) is the next most-recently accessed grammar having a medium recency of access characteristic (medium “*relevance score*”); and 70-3 (appearing at the bottom of the context list) is the least accessed grammar having the lowest recency of access characteristic (lowest “*relevance score*”). Ex.1003, ¶152.

Ross further explains its use of recency of access characteristic (e.g., “*a relevance score*”) of a context grammar is “*based on results from,*” for instance, on “the last application 26 that the user touched or talked to,” which is determined upon matching (“*comparing*) *processed spoken utterances* (“*the text combinations*”) “*to*” data (e.g., the phrases, keywords, and operators) in an application’s grammar (“*entries in the context description grammar*”):

[W]hen an utterance is recognized, it will be tested against each application’s grammar to see if the grammar will accept it. The order of testing is based on a dynamic speech focus priority. The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26. The speech focus priority is established by maintaining an

ordered list of applications (e.g., context list 62). Whenever an application 26 gains window focus, it will move to the head of the list 62. Likewise, whenever an application 26 which is not the top priority application 26 is chosen as the target for a speech command, the application 26 indicated by the selected context 72 will move to the head of the list 62. In this way, **the last application 26 that the user touched or talked to** will get the first opportunity at interpreting the next user utterance, and the other applications 26 will be ordered in a most-recently-accessed way.

Ex.1022, [0035]. *See also* [13.5] describing the “*comparing*” step. Ex.1003, ¶153.

**i. [13.8] selecting one or more domain agents based on results from the relevance score;**

Ross “maintains a context list of speech enabled applications” that the user has accessed and teaches “deciding the target application” (“*selecting one or more domain agents*”) for a particular utterance provided by the user, e.g., “*based on*” “a recency of access characteristic” (“*the relevance score*”). Ex.1022, [0005], Abstract; *see also* [13.7] (explaining the use of recency of access characteristic as “*relevance score*”). Ex.1003, ¶156.

Ross provides additional details of the “*selecting*” using the recency of access characteristic (“*the relevance score*”):

Whenever an application 26 gains window focus, it will move to the head of the list 62. Likewise, whenever an application 26 which is not the top priority application 26 is chosen as the target for a speech command, the application 26 indicated by the selected context 72 will move to the head of the list 62. In this way, **the last application 26 [“domain agent”] that the user touched or talked to will get the first opportunity at interpreting the next user utterance**, and the other applications 26 will be ordered in a most-recently-accessed way.

Ex.1022, [0035]. Identifying the last application that the user accessed (e.g., touched or talked to) would result in having that application having a higher recency of access characteristic, and therefore, Ross discloses “*selecting one ... domain agent[] based on*” recency of access (“*results from the relevance score*”). Ex.1003, ¶157.

Furthermore, Ross describes its system “biased to send the commands to the most recently accessed application 26” (“*based on results from the relevance score*”) (Ex.1022, [0054]) and provides an example of “*selecting*” a calendar application (“*one or more domain agents*”) “targeted to receive” the user’s spoken utterance “print it” because, as Ross describes, “**the context ... for the calendar application ... supersedes the context ... for the electronic mail application ... in the context list 62**” “*based on*” the recency of access characteristic (“*results from the relevance score*”). Ex.1022, [0058]; Ex.1003, ¶158.

In this example, “[t]he user ... us[ing] the mouse to click on the fourth appointment displayed in a window for the calendar application,” causes the

calendar application to “gain[] window focus” or otherwise have a higher recency of access characteristic (e.g., higher “*relevance score*”) to accept the spoken utterance “print it.” This example further confirms Ross discloses the “*selecting*” step. Ex.1022, [0058], [0035]; Ex.1003, ¶159.

As Dr. Houh explains, Ross describes details of “tracking window focus” for applications (in “*selecting one or more domain agents*”) using Microsoft Active Accessibility (MSAA) from Microsoft Corporation. Ex.1003, ¶160.

**j. [13.9] obtaining content that is gathered by the selected domain agents; and**

Initially, while there is recitation of “*selecting one or more domain agents*” in a previous claim element, there is no reference to “*selected domain agents*.” For purposes of analysis, Petitioner interprets “***the*** *selected domain agents*” in this claim element to mean “***the*** *selected one or more domain agents*.” Ross discloses this claim element as described below. Ex.1003, ¶162.

As explained in [13.8], Ross discloses the step of “*selecting one or more domain agents*” with an illustrative example in which the calendar application (“*the selected domain agent[]*”) is “targeted to receive” user’s spoken utterance “print it.” Ex.1022, [0058]. Because Ross teaches that “a grammar is defined for each application,” consequently, an application “targeted to receive” a user’s spoken utterance has “a grammar that will accept the phrase.” Ex.1022, [0035], [0053].

Further, Ross teaches applications “load[ing]” “grammars,” e.g., represented by contexts. Ex.1022, [0028] (disclosing the system “allows applications ... to load ... grammars”), [0033] (“contexts 70 (e.g., 70-1, 70-1, 70-3) for speech enabled applications ..., which represent the grammars for the applications ...”), and [0052] (“grammars ... loaded into the context manager”). Thus, in the calendar application example above, Ross’s system receives (“*obtain[s]*”) “the context ... for the calendar application” (“*content*”), e.g., including the data/entries for the calendar application (Ex.1022, [0046]) “load[ed]” (“*gathered by*”) the calendar application (“*the selected domain agent[]*”).

---

```

<appointment> = do I have any appointments |
  open <appointment> |
  create an appointment |
  print <appointment>.
<appointment> = the? <nth> appointment | it | this.

```

---

Ex.1022, [0046] (showing exemplary “*content*”); Ex.1003, ¶163.

- k. **[13.10] generating a response from the content, wherein the content is arranged in a selected order based on results from the relevance score.**

As explained in [13.9], in Ross, “the context ... for the calendar application” constitutes an example of “*the content*.” Furthermore, by teaching that “the context ... for the calendar application” “supersedes the context ... for the electronic mail application in the context list,” Ross describes that “*the content is arranged in a selected order*.” Ex.1022, [0058]; Ex.1003, ¶165.

Furthermore, Ross discloses its system “determines whether the context 70 [“*the content*”] for that application 26 “is **at the top** of the context list 62” [“*arranged in a selected order*”] and further that the system “has to make a choice by referring to the context list 62 of applications” “in order of recency of access” (“*based on results from the relevance score*”). Ex.1022, [0036], [0053]. Thus, with respect to the calendar application example in [13.9], the outcome of determining whether the context for the calendar application is at the top “*generat[es] a response*” identifying that “the context ... for the calendar application” (“*the content*”) “supersedes the context ... for the electronic mail application in the context list.” Ex.1022, [0058]; Ex.1003, ¶166.

For the additional reasons below, Ross discloses that “*the content is arranged in a selected order based on results from the relevance score*,” as recited in this claim element. Ex.1003, ¶167.

For example, Ross explains “grammars (indicated by the contexts 70 [e.g., including the grammar for the calendar application] in the context list 62) in **priority order.**” Ex.1022, [0053]; *see also* [0034] (“active grammars in the context list 62 in priority order” and “maintain[ing] the priority and state of the various grammars in the context list 62 in the system.”). Ex.1003, ¶168.

Ross explains the priority order of the various grammars (or, equivalently contexts represented by the grammars) is “*based on*” the most-recently accessed application, such that grammar of the last application that the user touched or talked to (“*results from the relevance score*”) has highest priority:

**The speech focus priority is established by maintaining an ordered list of applications (e.g., context list 62).**

Whenever an application 26 gains window focus, it will move to the head of the list 62. Likewise, whenever an application 26 which is not the top priority application 26 is chosen as the target for a speech command, the application 26 indicated by the selected context 72 will move to the head of the list 62. In this way, **the last application 26 that the user touched or talked to** will get the first opportunity at interpreting the next user utterance, and the other applications 26 will be ordered in a most-recently-accessed way.

Ex.1022, [0035]. Therefore, the priority order (or, equivalently the speech focus priority) of Ross's context list (comprising context grammars) in which the grammar for the most-recently accessed application (e.g., the calendar application) is at the top of the context list constitutes "*the content ... arranged in a selected order.*" Ex.1003, ¶169.

Ross further explains that its system:

**prioritize[s] the contexts [included in the context list] based on the access characteristic [which depends] on recency of relevant access to the context.**

Ex.1022, [0012]; *see also* [0053] ("**context list 62 of applications [arranged] in order of recency of access**"); Ex.1003, ¶170.

Ross continues how the "*the content ... [is] arranged in a selected order*":

[t]he context list 62 of applications is reordered every time a recognition is directed to an application 26 other than the highest priority one, or whenever another application 26 gets windows focus (for example, because it was clicked on).

Ex.1022, [0054]; Ex.1003, ¶171.

## 2. Claim 14

- a. **[14.0] The method according to claim 13, further comprising generating an aggregate response that**

**includes the content that is gathered by the selected domain agents.**

Ross provides examples of a grammar for a calendar application and a grammar for an electronic mail application, each of which includes the “**print it**” phrase. Ex.1022, [0045], [0051]. Ross explains: “[i]f the sentence is ‘print it,’” the system determines “**both grammars are capable of accepting the utterance**” and thus the system “tests the utterance against these grammars (indicated by the contexts 70 in the context list 62) in priority order.” Ex.1022, [0053]. As the above-mentioned disclosures demonstrate, the outcome of identifying that both grammars are capable of accepting the “print it” utterance and testing the processed utterance against these grammars, therefore, is “*generating an aggregate response.*” Furthermore, the “*aggregate response*” concerns a determination involving both grammars. Ex.1003, ¶173.

The “*aggregate response*” includes the “the context ... for the calendar application” (“*the content*”) “load[ed]” (“*gathered by*”) the calendar application (“*the selected domain agent[]*”) because, Ross teaches “[o]nly one grammar will accept the phrase, which thus indicates the selected context 72 for that phrase and that **associated application 26 is the one that should be targeted to receive the corresponding command.**” Ex.1022, [0052]. Indeed, Ross teaches that “the context ... for the calendar application” (“*the content*”) “supersedes the context ...

for the electronic mail application in the context list.” Ex.1022, [0058]. *See also* [13.9] and [13.10] explaining “*content ... gathered by the selected domain agents*” and “*a response from the content.*” Ex.1003, ¶174.

### 3. Claim 15

- a. **[15.0] The method according to claim 13, further comprising: receiving a follow-up speech and non-speech communications;**

Referencing Figure 2, Maes explains that when the system does not have enough original input to determine the user intent, “in step 214,” the system “generat[es] ... an output to the user **requesting further input data** [*“a follow-up speech and non-speech communications”*] so that the user’s intent can be disambiguated,” “the system ... **obtains** [*“receiv[es]”*] **the raw input data**, again in step 202, and the process ... iterates based on the new data. Such iteration can continue as long as necessary ... to determine the user’s intent.” Ex.1005, 8:47–50, 8:58–65. The system may “seek confirmation in step 216 from the user in the same manner as the request for more information (step 214).” Ex.1005, 8:66–9:1; Ex.1003, ¶175.

Accordingly, Maes’s system, which “*receiv[es] the speech and non-speech communications*” (*see* [13.1]), applies the same receiving technique of the “further input data” (“*a follow-up speech and non-speech communications*”) as the originally-received/raw input data. Ex.1003, ¶176.

Referencing the in-vehicle system example and the iterative Figure 2 process, Dr. Houh explains that, in response to the system requesting further input data, (e.g., with the question “what device do you want to have turned on?”), the user’s feedback saying the spoken utterance “the radio” and the accompanying lip movement constitutes “*a follow-up speech and non-speech communications.*” Ex.1003, ¶¶177–178 (citing Ex.1005, 8:43–65, Figure 2); *see also* Ex.1005, 8:43–65, 36:59–37:2.

- b. [15.1] transcribing the follow-up speech and non-speech communications to create a follow-up speech-based textual message and a follow-up non-speech-based textual message; and**

Maes’s system, which “*transcrib[es] the speech and non-speech communications to create a speech-based textual message and a non-speech-based textual message*” (*see* [13.2]), involves applying the same technique and proceeds in the same manner for the “further input data” (“*the follow-up speech and non-speech communications*”). Ex.1003, ¶¶179–180 (citing Ex.1005, 8:66–9:6).

Accordingly, a follow-up decoded text or script (“*a follow-up speech-based textual message*”) and a follow-up visual phonemes (visemes) sequence (“*a follow-up non-speech-based textual message*”) is created. Ex.1003, ¶181 (citing Ex.1005, 6:43–50, [13.2]).

- c. [15.2] merging the follow-up speech-based textual message and the follow-up non-speech-based textual message to generate a follow-up query.**

Maes discloses synchronizing/aligning the follow-up decoded text information and the follow-up visual phonemes information for the “further input data” (“*the follow-up speech and non-speech communications*”). Ex.1003, ¶183.

Moreover, the outcome of the “*merging*” for the “further input data” (“*the follow-up speech and non-speech communications*”) is “*to generate a follow-up query.*” Ex.1003, ¶184.

Additionally, referencing Maes’s in-vehicle system example (Ex.1005, 8:43–65) and the iterative Figure 2 process, Maes teaches the system “generat[ing] (“*generat[ing]*”) a predetermined question “what device do you want to have turned on?” (“*a follow-up query*”). Ex.1003, ¶¶185–186 (citing Ex.1005, 8:54–58); *see also* Ex.1005, 36:61-63, 67–37:2.

#### 4. Claim 17

a. [17.0] **The method according to claim 13, further comprising generating a context stack that includes one or more contexts that are selected based on the query.**

i. **Generating a context stack including contexts**

Maes describes “the context stack 817 may be **implemented**” (“*generating a context stack*”) as part of the context stack 20 of Figure 1. Ex.1005, 37:53–55.

Referencing Figure 1, Maes describes “the multi-modal conversational computing system 10 comprises ... a context stack 20.” Ex.1005, 3:66–4:6; Ex.1003, ¶187.

Maes's “*context stack*” as stores “historical information” e.g., past input/output (I/O) events or “*one or more contexts.*”

[H]istorical information (e.g., past events) stored in the context stack [such that] the context stack ... is associated with the **organized/sorted context** corresponding to each active **dialog**.

Ex.1005, 7:62–63, 37:60–61.

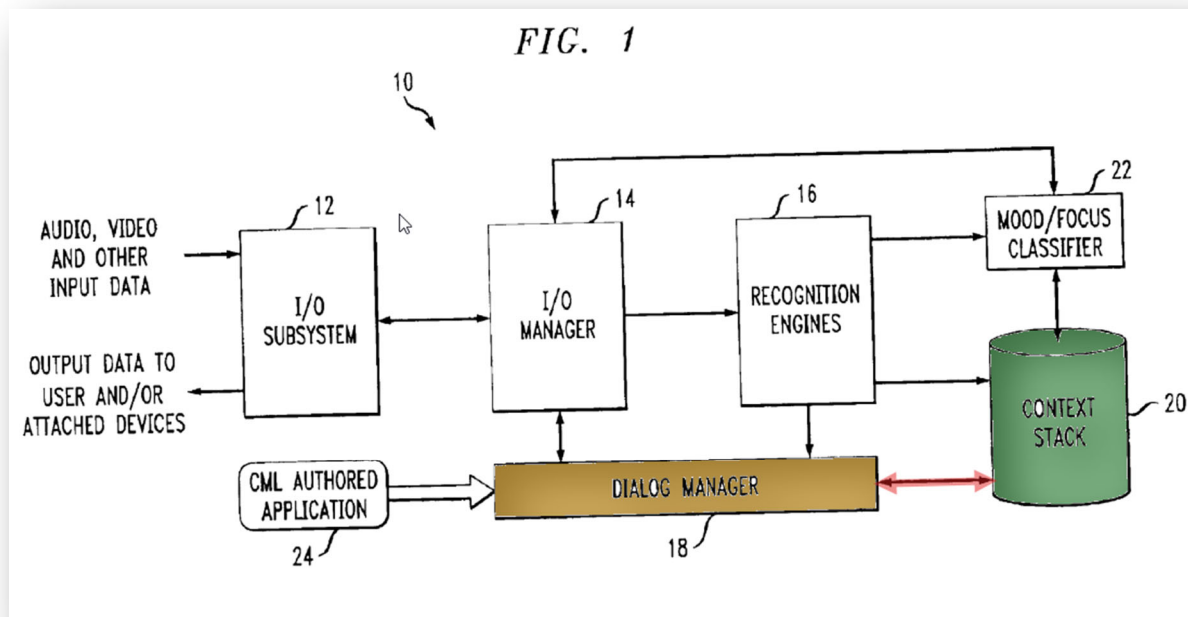


Figure 1 (annotated); Ex.1003, ¶188 (citing Ex.1005, 7:40–45, 5:16–19).

**ii. Contexts selected based on the query**

Maes discloses “*one or more contexts ... [are] based on the query*” because Maes’s context stack includes “queries to the backend.” Ex.1005, 37:55–62; Ex.1003, ¶189.

Maes describes “*one or more contexts ... are selected*” or simply, “[*selecting*] *one or more contexts.*” Referencing the in-vehicle example in which a user says “turn it on,” Maes describes its system identifying words in a recognized utterance from contexts—such as by matching the utterance “turn it on” to the context “radio”:

The dialog manager would ... receive the results of the recognized events associated with the spoken utterance “turn it on” and the gesture of pointing to the radio. Based on these events, the dialog manager does a search of the existing applications, transactions or “dialogs,” or portions thereof [stored on the context stack], with which such an utterance [including the accompanying lip movement] and gesture **could be associated**.

Ex.1005, 7:65–8:4.

[T]his recognized spoken utterance event is stored on the context stack. Then, when the recognized gesture event (e.g., pointing to the radio) is received, the dialog manager takes this event and the previous spoken utterance event stored on the context stack and makes a determination that the user intended to have the radio turned on.

Ex.1005, 8:37–42; Ex.1003, ¶¶190–191.

As Dr. Houh explains, Maes teaches **associating** the results of the recognized events (e.g., produced as a result of merging the transcriptions) with an organized/sorted context in a context stack such as context stack 817 in Figure 8 or context stack 20 in Figure 1 corresponding to an active dialog, which serves as describing the step of “[*selecting*] *one or more contexts*” from the context stack. Ex.1003, ¶192.

## 5. Claim 18

- a. **[18.0] The method according to claim 17, wherein the one or more contexts are generated based on applying prior probabilities or fuzzy possibilities to (i) keyword matching, (ii) user profiles, (iii) a dialog history, or any combination of (i) to (iii).**

Maes discloses maintaining a “*dialog history*.” Ex.1003, ¶195 (citing Ex.1005, 7:41–42, 37:55–61).

Maes also describes phonemes as examples of “*the one or more contexts*” stored on the context stack. Ex.1003, ¶197 (citing Ex.1005, 12:29–33, 7:60–8:4, 8:37–42, 37:60–61).

Further, Maes teaches a phoneme having an associated probability indicating the likelihood (“*based on ... [a] fuzzy possibilit[y]*”) that it was that particular phoneme/acoustic unit that was spoken (“*[applied] ... to dialog history*”). Ex.1003, ¶198.

For example, referencing Figure 4, Maes discloses:

After the acoustic feature vectors, denoted in FIG. 4. by the letter A, are extracted, the probability module labels the extracted vectors with one or more previously stored phonemes which, as is known in the art, are sub-phonetic or acoustic units of speech. ... Each phoneme associated with one or more feature vectors **has a probability associated therewith indicating the likelihood that it was that particular acoustic unit that was spoken.**

Ex.1005, 12:29–37.

[T]he probability module 416 in the audio information path ... labels the acoustic feature vectors with one or more phonemes.

Ex.1005, 18:46–48.

Thus, the probability module **yields likelihood scores for each considered phoneme in the form of the probability that, given a particular phoneme or acoustic unit (au), the acoustic unit represents the uttered speech characterized by one or more acoustic feature vectors A or, in other words,  $P(A|\text{acoustic unit})$ .**

Ex.1005, 12:38–43.

... Again, each phoneme associated with one or more visual speech feature vectors has **a probability associated therewith indicating the likelihood that it was that particular acoustic unit that was spoken in the video segment being considered. Thus, the probability module yields likelihood scores for each considered phoneme in the form of the probability that, given a particular phoneme or acoustic unit (au), the acoustic unit represents the uttered speech characterized by one or more visual speech feature vectors V or, in other words,  $P(V|\text{acoustic unit})$ .**

Ex.1005, 18:46–60. Accordingly, in Maes, a phoneme (“*the one or more contexts ... generated*”) having an associated probability indicating the likelihood that it was that particular phoneme/acoustic unit that was spoken, therefore, teaches or suggests “*the one or more contexts are generated*” “*based on applying ... fuzzy possibilities to ... a dialog history.*” Ex.1003, ¶199.

Moreover, as Dr. Houh explains, at the time of the 039 Patent, using likelihoods to calculate fuzzy probabilities was well-known. Ex.1003, ¶200 (citing Ex.1027, 43).

#### **IX. THE BOARD SHOULD NOT EXERCISE ITS DISCRETION AND DENY INSTITUTION**

This petition should be instituted for efficiency and fairness considerations, and further in view of its overwhelming strength on the merits. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 5 n.7 (Mar. 20, 2020) (“*Fintiv*”) (precedential). If Patent Owner disagrees and opts to file a discretionary denial brief, Petitioner will file an opposition in accordance with the Acting Director’s March 26, 2025 Memorandum.

#### **X. CONCLUSION**

Claims 13-15 and 17-18 of the 039 Patent are unpatentable for the reasons discussed above.

Dated: July 25, 2025

Respectfully Submitted,

/ Lisa K. Nguyen /

Lisa K. Nguyen (Reg. No. 58,018)

**PAUL HASTINGS LLP**

1117 S. California Avenue

Palo Alto, CA 94304

Telephone: 650-320-1800

lisanguyen@paulhastings.com

*Counsel for Petitioner*

*Meta Platforms, Inc.*

**CERTIFICATE OF COMPLIANCE**

I hereby certify that this brief complies with the type-volume limitations of 37 C.F.R. § 42.24, because it contains 12,394 words (as determined by the Microsoft Word word-processing system used to prepare the brief and including annotated figures), excluding the parts of the brief exempted by 37 C.F.R. § 42.24.

Dated: July 25, 2025

Respectfully Submitted,

/ Lisa K. Nguyen /

Lisa K. Nguyen (Reg. No. 58,018)

**PAUL HASTINGS LLP**

1117 S. California Avenue

Palo Alto, CA 94304

Telephone: 650-320-1800

lisanguyen@paulhastings.com

*Counsel for Petitioner*

*Meta Platforms, Inc.*

**CERTIFICATE OF SERVICE**

Pursuant to 37 C.F.R. § 42.6(e), I hereby certify that on this 25th day of July, 2025, I caused to be served a true and correct copy of the foregoing and any accompanying exhibits by U.S. Priority Mail Express on the following:

David Gerasimow  
The Law Offices of David A. Gerasimow, P.C.  
P.O. Box 10861  
Chicago, IL 60610

A courtesy copy of this Petition and supporting material was also served on litigation counsel for Patent Owner via email:

Garland Stephens (garland@bluepeak.law)  
Richard Koehl (richard@bluepeak.law)  
**BLUE PEAK LAW GROUP LLP**  
3139 West Holcombe Blvd. PMB 8160  
Houston, TX 77025

Mark D. Siegmund (msiegmund@cjsjlaw.com)  
Shuya Yang (gyang@cjsjlaw.com)  
**CHERRY JOHNSON SIEGMUNG JAMES, PLLC**  
7901 Fish Pond Road, Second Floor  
Waco, TX 76710

William D. Ellerman (ellerman@cjsjlaw.com)  
**CHERRY JOHNSON SIEGMUNG JAMES, PLLC**  
One Glen Lakes Tower  
8140 Walnut Hill Lane, Suite 105  
Dallas, TX 75231

Dated: July 25, 2025

Respectfully Submitted,

/ Lisa K. Nguyen /

Lisa K. Nguyen (Reg. No. 58,018)

**PAUL HASTINGS LLP**

1117 S. California Avenue

Palo Alto, CA 94304

Telephone: 650-320-1800

lisanguyen@paulhastings.com

*Counsel for Petitioner*

*Meta Platforms, Inc.*

**CLAIM LISTING**

<b>Claim 13</b>	
[13.0]	A method of processing speech and non-speech communications, comprising:
[13.1]	receiving the speech and non-speech communications;
[13.2]	transcribing the speech and non-speech communications to create a speech-based textual message and a non-speech-based textual message;
[13.3]	merging the speech-based textual message and the non-speech-based textual message to generate a query;
[13.4]	searching the query for text combinations;
[13.5]	comparing the text combinations to entries in a context description grammar;
[13.6]	accessing a plurality of domain agents that are associated with the context description grammar;
[13.7]	generating a relevance score based on results from comparing the text combinations to entries in the context description grammar;
[13.8]	selecting one or more domain agents based on results from the relevance score;
[13.9]	obtaining content that is gathered by the selected domain agents; and
[13.10]	generating a response from the content, wherein the content is arranged in a selected order based on results from the relevance score.
<b>Claim 14</b>	
[14.0]	The method according to claim 13, further comprising generating an aggregate response that includes the content that is gathered by the selected domain agents.
<b>Claim 15</b>	
[15.0]	The method according to claim 13, further comprising: receiving a follow-up speech and non-speech communications;
[15.1]	transcribing the follow-up speech and non-speech communications to create a follow-up speech-based textual message and a follow-up non-speech-based textual message; and
[15.2]	merging the follow-up speech-based textual message and the follow-up non-speech-based textual message to generate a follow-up query.
<b>Claim 17</b>	
[17.0]	The method according to claim 13, further comprising generating a context stack that includes one or more contexts that are selected based on the query.

**Claim 18**

[18.0] The method according to claim 17, wherein the one or more contexts are generated based on applying prior probabilities or fuzzy possibilities to (i) keyword matching, (ii) user profiles, (iii) a dialog history, or any combination of (i) to (iii).