

Filed: August 19, 2025

Filed on behalf of Meta Platforms, Inc.

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

META PLATFORMS, INC.,  
Petitioner

v.

DIALECT, LLC,  
Patent Owner

---

Case No. IPR2025-01334  
U.S. Patent No. 7,398,209

---

**Petition for *Inter Partes* Review of  
U.S. Patent No. 7,398,209**

**TABLE OF CONTENTS**

|                                              |           |
|----------------------------------------------|-----------|
| <b>I. INTRODUCTION</b>                       | <b>1</b>  |
| <b>II. IDENTIFICATION OF CHALLENGES.....</b> | <b>2</b>  |
| <b>III. BACKGROUND .....</b>                 | <b>2</b>  |
| A. '209 Patent.....                          | 2         |
| B. Prosecution History .....                 | 3         |
| <b>IV. PRIOR ART .....</b>                   | <b>4</b>  |
| A. Ross (EX1005) .....                       | 4         |
| B. Ittycheriah (EX1007).....                 | 6         |
| C. Hartono (EX1006).....                     | 8         |
| D. Baker (EX1008) .....                      | 11        |
| E. Ross018 (EX1009) .....                    | 13        |
| F. Aridor (EX1010) .....                     | 14        |
| <b>V. LEVEL OF ORDINARY SKILL .....</b>      | <b>16</b> |
| <b>VI. CLAIM CONSTRUCTION .....</b>          | <b>16</b> |
| <b>VII. GROUND 1.....</b>                    | <b>16</b> |
| A. Motivation to Combine .....               | 16        |
| B. Reasonable Expectation of Success .....   | 17        |
| C. Independent Claim 1 .....                 | 19        |
| 1. Preamble.....                             | 19        |
| 2. 1[a]: “receiving...” .....                | 20        |

|                        |                                   |           |
|------------------------|-----------------------------------|-----------|
| 3.                     | 1[b]: “maintaining...”            | 21        |
| 4.                     | 1[c]: “recognizing...”            | 25        |
| 5.                     | 1[d]: “parsing...”                | 26        |
| 6.                     | 1[e]: “selecting...”              | 30        |
| 7.                     | 1[f]: “formulating...”            | 31        |
| 8.                     | 1[g]: “invoking...”               | 32        |
| 9.                     | 1[h]: “presenting...”             | 32        |
| D.                     | Dependent Claim 4                 | 33        |
| 1.                     | 4[a]: “matching...”               | 33        |
| 2.                     | 4[b]: “scoring...”                | 34        |
| 3.                     | 4[c]: “determining...”            | 34        |
| <b>VIII. GROUND 2:</b> |                                   | <b>35</b> |
| A.                     | Motivation to Combine             | 35        |
| B.                     | Reasonable Expectation of Success | 36        |
| C.                     | Claim 6                           | 36        |
| 1.                     | 6[a]: “determining...”            | 36        |
| 2.                     | 6[b]: “extracting...”             | 38        |
| 3.                     | 6[c]: “inferring...”              | 40        |
| 4.                     | 6[d]: “transforming...”           | 42        |
| <b>IX. GROUND 3:</b>   |                                   | <b>43</b> |
| A.                     | Motivation to Combine             | 43        |

|           |                                            |           |
|-----------|--------------------------------------------|-----------|
| B.        | Reasonable Expectation of Success .....    | 44        |
| C.        | Claim 7 .....                              | 45        |
| 1.        | 7[a]: “directing...” .....                 | 45        |
| 2.        | 7[b]: “querying...” .....                  | 46        |
| D.        | Claim 8 .....                              | 46        |
| 1.        | 8[a]: “submitting...” .....                | 46        |
| 2.        | 8[b]: “a synchronously receiving...” ..... | 48        |
| 3.        | 8[c]: “scoring...” .....                   | 48        |
| E.        | Claim 9 .....                              | 50        |
| 1.        | 9[a]: “determining...” .....               | 50        |
| 2.        | 9[b]: “extracting...” .....                | 51        |
| 3.        | 9[c]: “evaluating...” .....                | 52        |
| 4.        | 9[d]: “selecting...” .....                 | 52        |
| <b>X.</b> | <b>GROUND 4:.....</b>                      | <b>53</b> |
| A.        | Motivation to Combine .....                | 53        |
| B.        | Reasonable Expectation of Success .....    | 55        |
| C.        | Independent Claim 1 .....                  | 56        |
| 1.        | Preamble.....                              | 56        |
| 2.        | 1[a] .....                                 | 56        |
| 3.        | 1[b] .....                                 | 57        |
| 4.        | 1[c] .....                                 | 60        |

|            |                                         |           |
|------------|-----------------------------------------|-----------|
| 5.         | 1[d] .....                              | 61        |
| 6.         | 1[e] .....                              | 63        |
| 7.         | 1[f].....                               | 64        |
| 8.         | 1[g] .....                              | 65        |
| 9.         | 1[h] .....                              | 65        |
| D.         | Dependent Claims .....                  | 66        |
| 1.         | Claim 4.....                            | 66        |
| a.         | 4[a] .....                              | 66        |
| b.         | 4[b].....                               | 66        |
| c.         | 4[c] .....                              | 68        |
| 2.         | Claim 6.....                            | 69        |
| a.         | 6[a] .....                              | 69        |
| b.         | 6[b].....                               | 70        |
| c.         | 6[c] .....                              | 71        |
| d.         | 6[d].....                               | 72        |
| 3.         | Claim 7 .....                           | 73        |
| a.         | 7[a] .....                              | 74        |
| b.         | 7[b].....                               | 74        |
| <b>XI.</b> | <b>GROUND 5:</b> .....                  | <b>75</b> |
| A.         | Motivation to Combine .....             | 75        |
| B.         | Reasonable Expectation of Success ..... | 76        |

|                                    |                                                |           |
|------------------------------------|------------------------------------------------|-----------|
| 1.                                 | Claim 8 .....                                  | 76        |
| a.                                 | 8[a] .....                                     | 76        |
| b.                                 | 8[b] .....                                     | 77        |
| c.                                 | 8[c] .....                                     | 77        |
| 2.                                 | Claim 9 .....                                  | 77        |
| a.                                 | 9[a] .....                                     | 77        |
| b.                                 | 9[b] .....                                     | 78        |
| c.                                 | 9[c] .....                                     | 78        |
| d.                                 | 9[d] .....                                     | 79        |
| <b>XII. MANDATORY NOTICES.....</b> |                                                | <b>79</b> |
| A.                                 | Real Parties-in-Interest .....                 | 79        |
| B.                                 | Related Matters.....                           | 79        |
| C.                                 | Notice of Counsel and Service Information..... | 80        |
| D.                                 | Fee for <i>Inter Partes</i> Review .....       | 81        |
| E.                                 | Grounds for Standing .....                     | 81        |
| <b>XIII. CONCLUSION .....</b>      |                                                | <b>81</b> |

**TABLE OF AUTHORITIES**

|                                                                                           | <b>Page(s)</b> |
|-------------------------------------------------------------------------------------------|----------------|
| <b>Cases</b>                                                                              |                |
| <i>Dialect LLC v. Google LLC</i> ,<br>Case No. 3:24-cv-04388 (N.D. Cal.) .....            | 83             |
| <i>Dialect LLC v. Microsoft Corp.</i> ,<br>Case No. 2:24-cv-01067 (E.D. Tex.) .....       | 83             |
| <i>Dialect LLC v. Salesforce, Inc.</i> ,<br>Case No. 7:25-cv-00061 (W.D. Tex.) .....      | 83             |
| <i>Dialect, LLC v. Meta Platforms, Inc.</i> ,<br>Case No. 7:25-cv-00060 (W.D. Tex.) ..... | 82             |
| <b>Statutes</b>                                                                           |                |
| 35 U.S.C. § 103 .....                                                                     | 2              |
| <b>Other Authorities</b>                                                                  |                |
| 37 C.F.R. § 42.10(b) .....                                                                | 84             |
| 37 C.F.R. § 42.15(a) .....                                                                | 84             |
| 37 C.F.R. §42.24(a) .....                                                                 | 90             |
| 37 C.F.R. § 42.24(d) .....                                                                | 90             |
| 37 C.F.R. § 42.104(a) .....                                                               | 84             |

**TABLE OF EXHIBITS**

| <b>Exhibit No.</b> | <b>Description</b>                                                                                                                                                   |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1001               | U.S. Patent No. 7,398,209 to Kennewick, et al. (“the ’209 patent”)                                                                                                   |
| 1002               | File History of U.S. Patent No. 7,398,209                                                                                                                            |
| 1003               | Declaration of Daniel Lopresti (“Lopresti”)                                                                                                                          |
| 1004               | Curriculum Vitae of Daniel Lopresti                                                                                                                                  |
| 1005               | U.S. Patent No. 7,085,723 to Ross et al. (“Ross”)                                                                                                                    |
| 1006               | WO 00/11571 to Hartono et al. (“Hartono”)                                                                                                                            |
| 1007               | U.S. Patent No. 5,937,383 to Ittycheriah et al. (“Ittycheriah”)                                                                                                      |
| 1008               | U.S. Patent No. 5,680,511 to Baker et al. (“Baker”)                                                                                                                  |
| 1009               | U.S. Patent No. 7,249,018 to Ross et al. (“Ross018”)                                                                                                                 |
| 1010               | U.S. Patent No. 6,636,848 to Aridor et al. (“Aridor”)                                                                                                                |
| 1011               | U.S. Patent Application Publication No. 2002/1065860 to Glover et al. (“Glover”)                                                                                     |
| 1012               | U.S. Patent No. 7,127,402 to Ross et al. (“Ross402”)                                                                                                                 |
| 1013               | Appendix A – Infringement Contentions for U.S. Patent No. 7,398,209, <i>Dialect LLC v. Meta Platforms, Inc.</i> , No. 7:25-CV-00060-DC-DTG (E.D. Tex. July 16, 2025) |
| 1014               | Reserved                                                                                                                                                             |
| 1015               | Reserved                                                                                                                                                             |

| Exhibit No. | Description                                                                                                                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1016        | U.S. Provisional Patent Application No. 60/261372 to Ross et al.                                                                                                                                                    |
| 1017        | IBM® ViaVoice® for Windows® Personal Edition, User's Guide, Release 10 (August 2002)                                                                                                                                |
| 1018        | Excerpts from IBM ViaVoice™ 98 Home Edition User Guide (July 1998)                                                                                                                                                  |
| 1019        | Excerpts from Frederick Jelinek, <i>Statistical Methods for Speech Recognition</i> , The MIT Press (1997)                                                                                                           |
| 1020        | Michael McTear, <i>Spoken Dialogue Technology: Enabling the Conversational User Interface</i> , 34 ACM Computing Surveys No. 1, pp. 90-169 (Mar. 2002)                                                              |
| 1021        | Liam Comerford, et al., <i>The IBM Personal Speech Assistant</i> , IBM Watson Research Center, IEEE (2001)                                                                                                          |
| 1022        | Eric J. Glover, et al., <i>Architecture of a Metasearch Engine that Supports User Information Needs</i> , In International Conference on Information and Knowledge Management, Proceedings, pp. 210-216, ACM (1999) |
| 1023        | Dragon Naturally Speaking® 5, User's Guide (August 2000)                                                                                                                                                            |
| 1024        | Definitions of “asynchronous” and “token” from Microsoft Computer Dictionary, Fourth Edition, Microsoft Corporation (1999)                                                                                          |
| 1025        | Definition of “dictionary” from Webster's Ninth New Collegiate Dictionary, Merriam-Webster Inc. (1990)                                                                                                              |
| 1026        | Definition of “dictionary” from The American Heritage Dictionary of The English Language, Houghton Mifflin Co. (1992)                                                                                               |

## I. INTRODUCTION

Meta Platforms, Inc. (“Petitioner”) respectfully requests *inter partes* review of claim 1, 4, 6-9 of U.S. Patent No. 7,398,209 (EX1001, “’209 patent”).

Natural language speech recognition methods have been used for decades. *See generally* Lopresti ¶¶43-49. By the mid-1990s, several companies, including IBM and Dragon Systems, had already been selling speech recognition systems off-the-shelf to process speech into text. Importantly, these applications routinely used dictionaries and phrase tables to identify words.

The ’209 describes a speech recognition system that accepts speech to make requests among various application programs. Notably, the Examiner only allowed the challenged claims using both *a dictionary and phrase table* to recognize the speech input, a well-known feature.

Accordingly, Petitioner respectfully requests the Board institute review and find all challenged claims unpatentable.

## II. IDENTIFICATION OF CHALLENGES

| Ground | Claims  | Statute                    | Prior Art                                                |
|--------|---------|----------------------------|----------------------------------------------------------|
| 1      | 1, 4    | Pre-AIA 35<br>U.S.C. § 103 | Ross (EX1005), Ittycheriah (EX1007),                     |
| 2      | 6       |                            | Ross (EX1005), Ittycheriah (EX1007),<br>Ross018 (EX1009) |
| 3      | 7-9     |                            | Ross (EX1005), Ittycheriah (EX1007),<br>Aridor (EX1010)  |
| 4      | 1, 4, 6 |                            | Hartono (EX1006), Baker (EX1008)                         |
| 5      | 7-9     |                            | Hartono (EX1006), Baker (EX1008),<br>Aridor (EX1010)     |

For this Petition only, Petitioner applies the earliest possible priority date—June 3, 2002—for the '209.

## III. BACKGROUND

### A. '209 Patent

The '209 describes well-known methods of natural language processing of a spoken request. Natural language processing and speech recognition were ubiquitous by 2002. EX1001, 1:38-43. The '209 describes an “integrated approach” to natural language processing by essentially combining admittedly well-known systems and methods. *Id.*, 1:65-2:1. 9:57-61.

The '209 describes high-level steps to process natural language speech. EX1001, Figs. 4A, 4B. Namely, the words and phrases contained in a natural language “utterance” are “recognized,” by using “one or more dictionary and phrase

tables.” *Id.*, 24:27-32. Once recognized, using a “parser,” the words and phrases are used “to determine a meaning” of the utterance using “criteria and/or parameters of the properly interpreted query and/or command.” *Id.*, 24:50-25:1, Fig 6. The ’209 next describes selecting a “domain agent” based on the determined meaning, i.e., “an autonomous executable that receives, processes, and responds to requests associated with the determined context.” *Id.*, cl. 1. The utterance “request” is formulated “in accordance with a grammar used by the selected domain agent,” and the domain agent is then able to process the request. *Id.*; *see also* 25:29-43. The results of the processed request are then presented to the user. *Id.*, cl. 1. Lopresti ¶¶55-57.

## **B. Prosecution History**

After numerous rejections, applicant contacted the Examiner for an interview on October 29, 2007. EX1002, 211. The Examiner suggested narrower subject matter “including fuzzy probabilities and specific methods of variable substitutions.” *Id.*, 212. Subsequently, the applicant cancelled all remaining claims and presented new claims 44-57. *Id.*, 328. On May 21, 2008, the Examiner issued a Notice of Allowance for these new claims. *Id.*, 341. The Examiner stated “[w]hile it is known to recognize words based on a vocabulary definition by a dictionary for speech recognition, the prior art of record does not disclose or reasonably suggest additionally using phrase tables for speech recognition.” *Id.*, 342. Lopresti ¶¶58-62.

#### IV. PRIOR ART

##### A. Ross (EX1005)

Ross is pre-AIA §102(a) and (e) prior art. EX1005, Cover (filed August 16, 2001, issued August 1, 2006).

Ross discloses a “speech center” that “coordinates speech services for a number of speech-enabled applications,” including the “task of deciding the target application for a particular utterance by a user.” EX1005, Abstract, 2:3-9, Fig. 1. Upon receiving a speech input, Ross uses an already-known speech recognition engine to convert spoken commands (i.e., requests) “into text strings that can be utilized by software applications,” i.e., a “speech representation.” EX1005, 1:16-18, 4:34-38; 6:62-7:2; Lopresti ¶¶70-72.

Ross describes a “context manager” that receives the speech representation from the “speech engine,” which proceeds to “select a context 70 for a speech enabled application 26 by checking the speech representation against contexts 70.” EX1005, 7:19-23. Ross ranks the contexts based on an “access characteristic,” such as the “recency of relevant access to the context,” as in Figure 4:

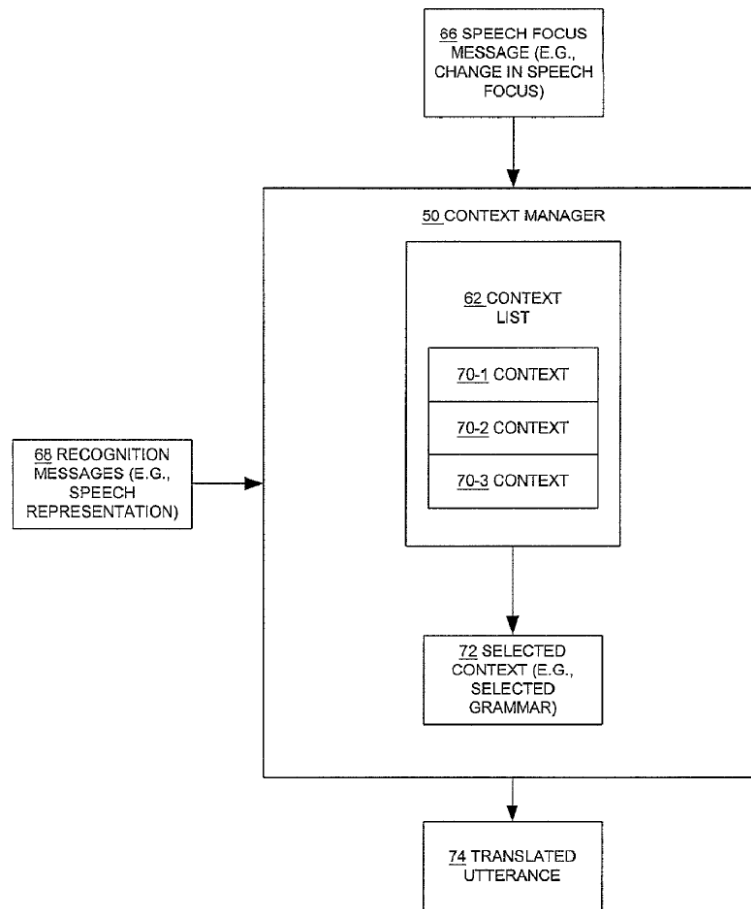


FIG. 4

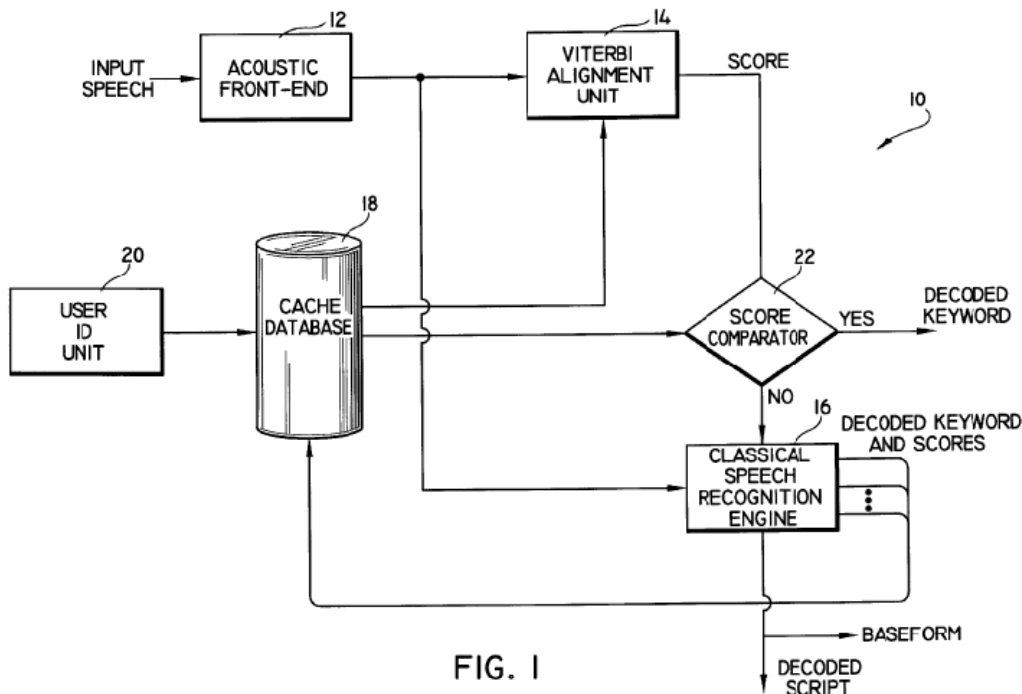
EX1005, Fig. 4, 2:47-49, 3:1-3, 6:47-61, 8:46-9:13; Lopresti ¶¶72-72. Ross thus maintains a set of prior probabilities of possible contexts “biased to send commands to the most recently accessed application,” that is continuously “reordered.” EX1005, 8:47-53; *see also* 5:58-61; Lopresti ¶73. Ross uses “the grammar associated with each context 70 to determine if the [speech representation] is capable of being processed by the grammar,” and selects the context using relevance matching (scoring). EX1005, 7:23-26. Ross translates the speech representation

“according to the selected context 72 and directs the translation to the script engine,” which runs the script against the target application, thereby invoking it to perform the function of the command and present the results to the user. EX1005, 7:27-28; Lopresti ¶¶74-75.

### B. Ittycheriah (EX1007)

Ittycheriah is pre-AIA §102(a), (b), and (e) prior art. EX1007, Cover (filed June 4, 1997, issued August 10, 1999).

Ittycheriah discloses a speech recognition engine for “performing speech recognition on speech segments frequently input by a user,” shown in Figure 1:



EX1007, Fig. 1, Abstract, 1:11-15, 1:51-54. Ittycheriah achieves this by “storing data in a cache and utilizing such data in order to quickly accept or reject keywords,

i.e., names, command, or sentences[.]” EX1007, 2:59-63. Ittycheriah’s cache “contains the word history of a keyword (name, command or sentence) as well as the likelihood (score) obtained when the word was added to the cache.” EX1007, 2:63-66; Lopresti ¶¶76-77. Ittycheriah uses Hidden Markov Models (HMMs) to perform speech recognition, explaining that HMMs provide “different scores for different hypothesis” using a set of prior probabilities for speech inputs. EX1007, 3:54-4:4; Lopresti ¶78. Ittycheriah compares the “resulting scores” to “scores which were previously stored in the cache database.” EX1007, 4:33-43. “If the new scores are within a margin of, the same as, or larger than the stored scores...in the cache, then the keyword is accepted.” EX1007, 4:43-48. If not, “the input speech segment is subjected to the complete vocabulary of the classical speech recognition engine,” from which a decoded script (keyword) is outputted. EX1007, 4:48-53; Lopresti ¶¶79-80.

Ittycheriah’s vocabulary in its “classical speech engine” contains a “larger scale language database.” EX1007, 3:62-64. It stores “keywords,” which Ittycheriah defines as “words and phrases as names, commands, and sentences, which include both functional and non-functional words,” or “other forms of vocabulary.” EX1007, 1:55-58, 2:16-19, 2:47-49. A POSITA would have understood that Ittycheriah’s vocabulary would have been considered a “dictionary”

of words and “phrase tables,” including commands and sentences as in the ’209. Lopresti ¶¶81-82.

**C. Hartono (EX1006)**

Hartono is §102(a) and (b) prior art. EX1006, Cover (published March 2, 2000).

Hartono discloses “method[s] for receiving commands, executing received commands and adaptively interacting with the user using a natural language interface.” EX1006, 1:20-23; Lopresti ¶¶85-96. Hartono describes a “natural language agent adapted to receive and interpret natural language user input and to output a command,” and an “application agent adapted to receive and further interpret the output command...and to output an executable instruction to an application program. EX1006, 4:20-25. Hartono illustrates its use of various application agents in Figure 1:

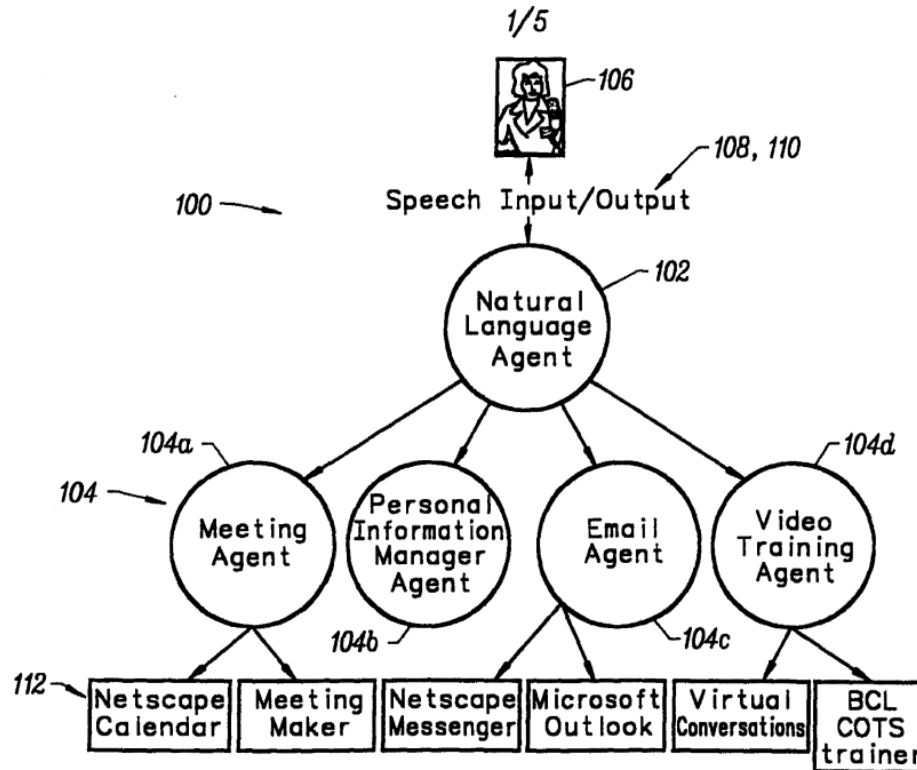


FIG. 1

EX1006, Fig. 1, 7:6-9; Lopresti ¶¶87-89. Hartono processes input speech using “[a]ny suitable off the shelf speech recognition system.” EX1006, 11:4-12. After receiving the recognized speech, Hartono discloses using a “natural language syntactic parser” to “parse natural language computer commands” from the recognized, for example, converting “the verb into a computer command action, with the noun phrase (NP) as the object and the adjective phrase (AP) as the properties of the object.” EX1006, 12:11-13:10; Lopresti ¶¶90-92. Hartono takes the parsed natural language and “maps it into a generic concept frame used to invoke the appropriate application method,” as in Table I:

| TABLE I        |               |                          |
|----------------|---------------|--------------------------|
| ACTION-CONCEPT | TOPIC CONCEPT | APPLICATION-CONTEXT      |
| Show           | Email         | Email application        |
| Show           | Email address | Address book application |
| Delete         | Email         | Email application        |
| Show           | General help  | Natural language agent   |

EX1006, Table I. To complete this mapping, Hartono's uses a "reverse grammar generation mechanism" that "includes a list or vector for each word and corresponding probabilities for each word in the list." EX1006, 14:10-20. For example, Hartono describes the words "I," "eye," or "aye," as having probabilities of 80%, 15%, and 5%, respectively, such the highest probability word is most likely to be suitable for the given context. EX1006, 15:4-12. Using both the action-concept and topic-concepts, Hartono determines a context by "determin[ing] which task agent should handle the request." EX1006, 15:13-19; Lopresti ¶¶93-95.

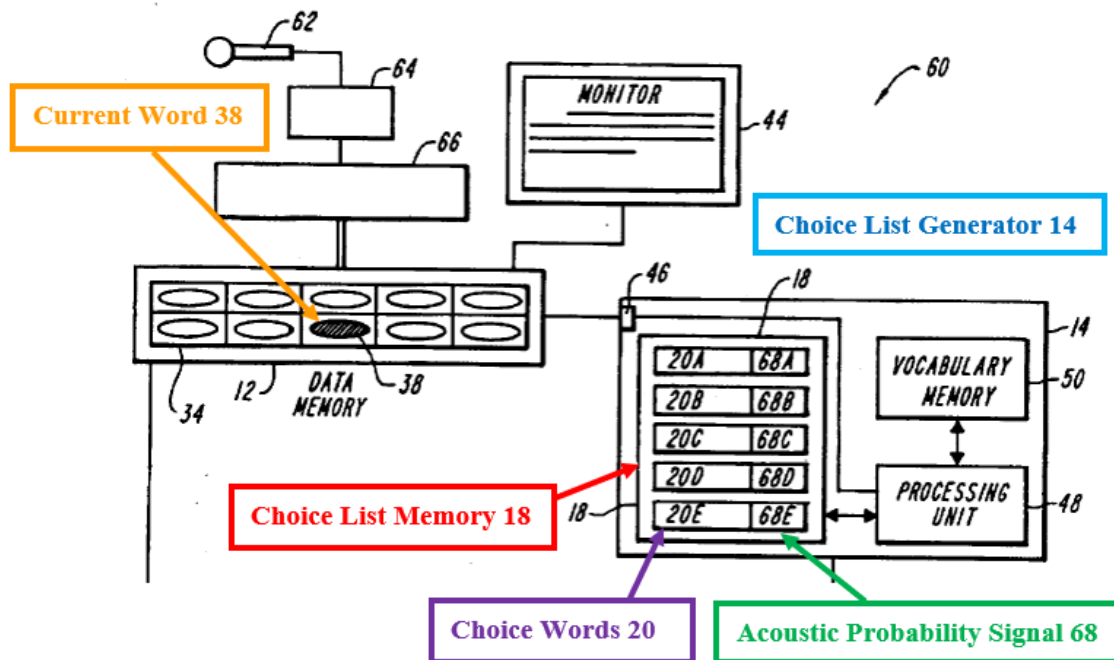
Hartono communicates with various "application class agents" (also "task agents") that "work[] with a single class of applications...that have similar conceptual operations." EX1006, 23:15-23, 28:1-3. Based on the determined context, Hartono "invokes a routine that sends a set of requests to the task application wrapper," which is "written in one of the platform specific macro languages." EX1006, 24:14-25:3. The selected application executes the command within the natural language request. EX1006, 4:20-25, 5:4-8, 5:11-15, 29:21-22, 42:11-12; Lopresti ¶¶96-98.

**D. Baker (EX1008)**

Baker is pre-AIA §102(a), (b), and (e) prior art. EX1008, Cover (filed June 7, 1995, issued October 21, 1997)

Baker discloses a speech recognition system for “detect[ing] words within...speech signals and compar[ing] these detected words to acoustic models of known words, or sequence of acoustic models of known words.” EX1008, 18:13-16; Lopresti ¶¶97-110. After detecting a “speech signal,” Baker compares the speech to “acoustic models of known words.” EX1008, 18:12-16. “This comparison leads to a probability signal that indicates the certainty that the detected word is substantially acoustically similar to the word associated with the acoustic model.” EX1008, 18:16-19. Words with a “high degree of certainty” are deemed recognized (“context words”). EX1008, 18:18-21. Words with a “low degree of certainty” have not been confidently identified (“current word”). EX1008, 18:22-24; Lopresti ¶¶99-102.

For unrecognized “current words,” Baker maintains a “choice list 18” that “store[s] the words 20 associated with the acoustic models from the vocabulary memory 50 which are acoustically similar to the current ambiguous word 38 being identified,” as in Figure 2:



EX1008, Fig. 2 (annotated)

Baker explains that each “choice word 20A-E” has a corresponding “acoustic probability signals 68A-E” that represents “the degree of certainty that the word 20 associated with the acoustic model is substantially representative of the current word 38 being recognized.” EX1008, 18:30-35, 20:30-46, 22:55-64; Lopresti ¶¶103-104.

Baker also discloses generating “word list” that has “semantic association with the choice word 20.” EX1008, 14:57-69. The “semantic association” for each choice word (or “passage words,” item 24 in Fig. 2) in the “word list” is derived from a “reference database 40,” which includes a “dictionary, thesaurus, antonym dictionary, medical dictionary, legal dictionary, scientific and engineering dictionary, or other type of reference source.” EX1008, 20:55-58. Using a

“matching function,” Baker “calculate[s] the number of matched words” between the recognized words (“context words”) with the words in the “word list” (“passage words”). EX1008, 21:19-47. Based on the “relative frequency of matches between “context words” and “passage words,” Baker calculates a “probability signal 28 representative of the likelihood that the choice word or phrase 20 substantially represents the utterance currently being recognized.” EX1008, 21:19-47, 21:49-22:6, Figs. 2-4; Lopresti ¶¶105-107.

Baker generates a “rank signal 30,” “represent[ing] the likelihood...that a choice word 20 is substantially similar to” the recognized word based on “context matching and acoustic model matching” probabilities. EX1008, 23:22-27. Baker selects the choice word with the highest rank signal and supplants the unrecognized word in the text with the ranked choice word, thus recognizing the speech. EX1008, 12:54-62, 17:23-27; Lopresti ¶¶108-109.

**E. Ross018 (EX1009)**

Ross018 is pre-AIA §102(a) and (e) prior art. EX1009, Cover (filed October 25, 2001, issued July 24, 2007).

Ross018 details transforming the natural language utterance into a grammar of a given external application or domain. It builds on Ross, by teaching a “semantics analysis module” and “syntax manager,” that processes a recognized spoken utterance “using a grammatic specification 90 based on the domain model

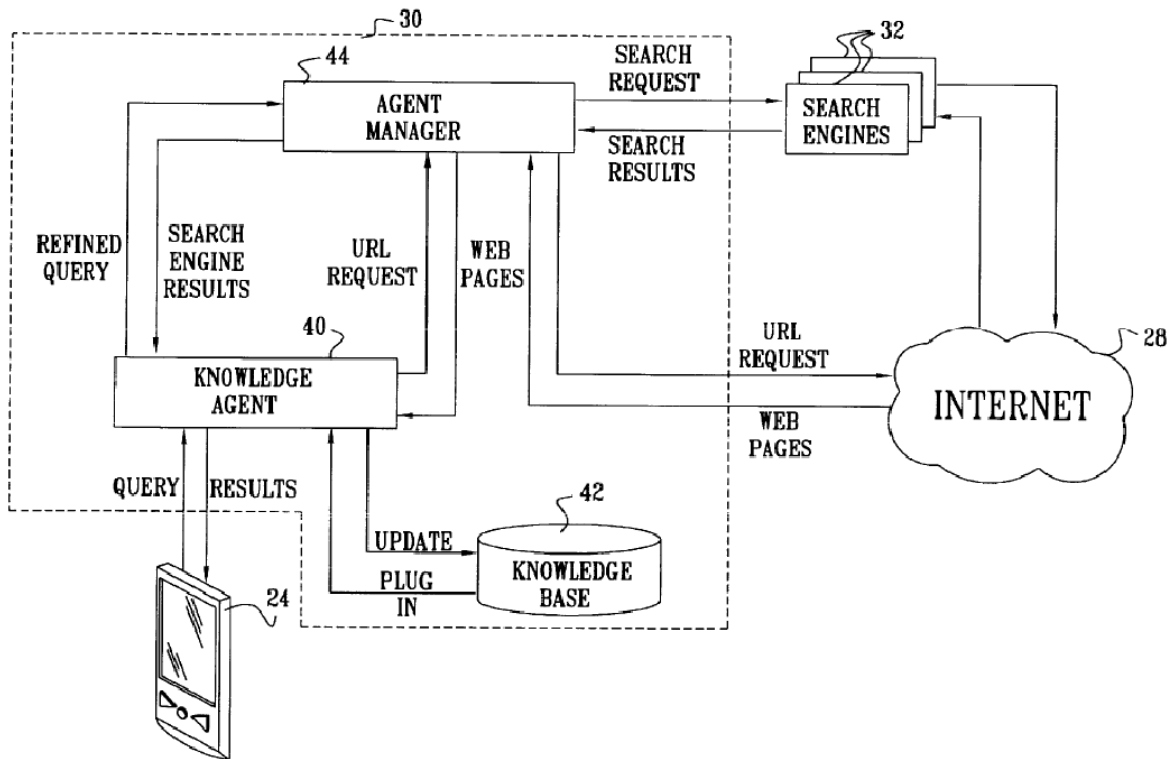
70 to produce an initial semantic representation based on the recognized utterance.” EX1009, Abstract, 11:61-65; Lopresti ¶¶83-85. More specifically, Ross018 creates a “frame structure based on the recognized spoken utterance” that is used to “provide[] a set of propositions.” EX1009, 12:4-10. These “propositions” are composed of attribute-object-value triples, that include criteria, parameters, and values associated with the spoken utterance. EX1009, 10:5-30. Using these propositions, Ross018 generates a “goal” that is used to obtain a response to the utterance, including “an action to be executed, such as a script call or command to an external application.” EX1009, 12:11-20; Lopresti ¶¶85-86.

**F. Aridor (EX1010)**

Aridor is pre-AIA §102(a) and (e) prior art. EX1010, Cover (filed June 6, 2000, issued October 21, 2003).

Aridor describes methods for searching the Internet for information using multiple search engines simultaneously, and filtering and ranking the retrieved responses, as shown below:

FIG. 2



EX1010, Abstract, Fig. 2; Lopresti ¶110. Aridor uses “knowledge agents” to perform these searches, each of which “specializes in a domain.” EX1010, 3:38-52. “The knowledge agent typically interacts with multiple search engines 32 via network 28, directing the engines to search sites 26 for information relevant to a query entered by user 22.” EX1010, 8:23-3. After “invoking several general-purpose search engines specified by the user 22,” the results are ranked and filtered by the knowledge agent, such that “the most relevant pages for the user will be listed first in the result.” EX1010, 8:45-59, 9:6-14, 11:19-22; Lopresti ¶¶111-112.

## **V. LEVEL OF ORDINARY SKILL**

A person of ordinary skill in the art (“POSITA”) in June 2002 would have had at least 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, and speech recognition and transcription. Additional experience could substitute for less education, and additional education could likewise substitute for less experience. Lopresti ¶¶64-66.

This Petition does not turn on this precise definition. The challenged claims would be unpatentable from the perspective of any reasonable POSITA at the relevant time. Lopresti ¶¶67-69.

## **VI. CLAIM CONSTRUCTION**

This Petition establishes the prior art meets each claim limitation under any reasonable construction—no express construction necessary.<sup>1</sup>

## **VII. GROUND 1**

### **A. Motivation to Combine**

A POSITA would have been motivated to use Ittycheriah’s speech recognition method to implement the details of Ross’s “speech engine.” Lopresti ¶¶115-121.

---

<sup>1</sup> Petitioner reserves the right to argue alternative constructions in other proceedings.

Ross recognizes that off-the-shelf speech engines could be used to convert speech to text in its system. EX1005, 1:16-21, 4:37-38; *see also* EX1016, 3. Thus, a POSITA would have been motivated to look for and rely on available speech recognition methods, including Ittycheriah (also developed by IBM). Lopresti ¶¶115-118.

A POSITA would have also been motivated to implement Ittycheriah to recognize speech with its cache database of recently-used keywords and probability scores, which was faster than classical engines. *See* Section IV.B; Lopresti ¶119. A POSITA would know that there were multiple design choices to improve the accuracy and speed of speech recognition, including utilizing recently used keywords and probability scores. *Id.* Thus, a POSITA would have found Ittycheriah's cache database desirable to "reduce the amount of hypotheses to consider in the decoding process." EX1007, 2:13-19, 2:58-63. These features would have advanced Ross's objective of "avoid[ing] the time, expense, and footprint of duplicating speech services." EX1005, 2:14-16. Lopresti ¶¶119-121.

#### **B. Reasonable Expectation of Success**

A POSITA would have had a reasonable expectation of success in combining Ross and Ittycheriah. Lopresti ¶¶122-125. A POSITA would have used Ittycheriah's speech recognition process to implement the details of Ross's "speech engine," as in Figure 2, which Ross teaches as any commercially-available speech recognition engine:

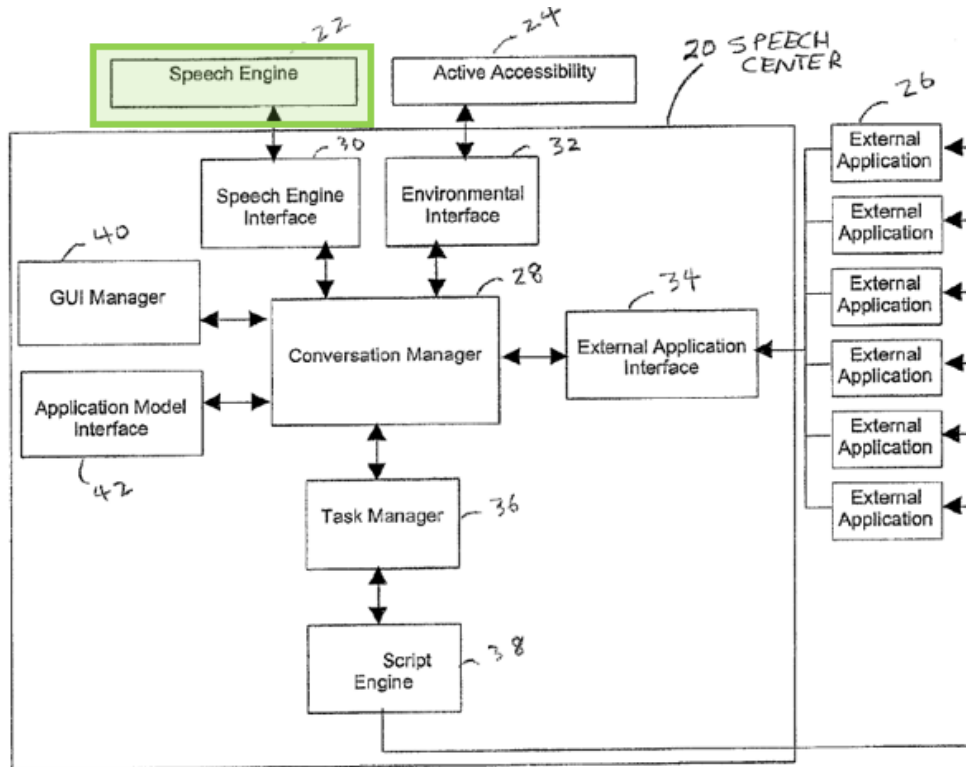


Fig. 2

**EX1005, Fig. 2 (annotated)**

This is particularly true since IBM developed both Ross and Ittycheriah. EX1005, [73]; EX1007, [73].

Indeed, a POSITA would have understood that Ittycheriah's speech recognition system is compatible with Ross, since Ittycheriah discloses that decoded keywords and scripts could be used to direct speech requests to external speech-enabled applications to perform their respective automatic functions, as in Ross. EX1007, 4:63-5:3; Lopresti ¶124.

A POSITA would have also had the skill to implement and reasonable expectation of success in achieving the combination because it would have involved combining known technologies (speech recognition and processing natural language requests) according to known methods (speech recognition engines that convert speech into text strings that can be used by software applications) to yield the predictable result of inputting natural language speech and outputting a text string that can invoke various speech-enabled applications on a computer. Lopresti ¶¶125.

### **C. Independent Claim 1<sup>2</sup>**

#### **1. Preamble**

To the extent limiting, Ross discloses the preamble. Lopresti ¶¶127-132. As discussed in Section IV.A, Ross discloses “determining a speech-enabled application to receive a spoken utterance in a multi-context speech enabled environment.” EX1005, 2:43-46, 5:34-38, Abstract.

Ross discloses examples of *responding to a user generated natural language speech utterance*. See, e.g., EX1005, 2:25-28, 2:31-36, 8:25-33. Ross describes that when “the user asks whether the user has any new messages,” the user “receives an answer supplied by a message or mail application.” EX1005, 2:28-31. A POSITA would have understood that these voice commands are *natural language* because

---

<sup>2</sup> Appendix A contains the claim language for all challenged claims.

they are human-understandable. Lopresti ¶¶129-132. This is illustrated in Figure 1, showing a processor receiving “input spoken utterance 14,” which is a “voice command or other speech input from a user of a computer system” and producing output 16 in response:

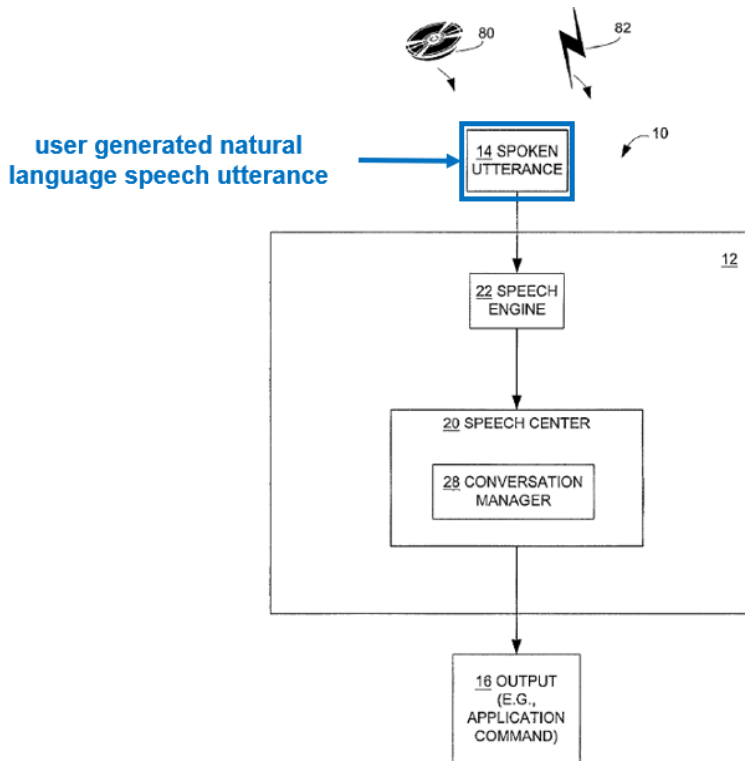


FIG. 1

**EX1005, Fig. 1 (annotated)**

See also EX1005, Fig. 1, 3:41-53; Lopresti ¶130.

**2. 1[a]: “receiving...”**

Ross discloses this limitation. Lopresti ¶¶133-136; see Section VII.C.1 (Ground 1, Claim 1[Preamble]). Ross discloses “[t]he input spoken utterance 14 is a voice command or other speech input from a user of the computer system 10 (e.g.,

into a microphone connected to the computer system 10)” (*receiving the user generated natural language speech utterance*). EX1005, 3:45-48; Lopresti ¶134 A POSITA would have understood a “voice command” is *at least one request*. EX1005, 3:45-48, 9:40-43, 2:22-36; Lopresti ¶¶135-136.

### 3. 1[b]: “maintaining...”

The '209 does not describe which of its claimed steps are a “stage of processing the received user utterance.” However, the specification discusses the use of “prior probabilities” and “fuzzy possibilities” in relation to “recognizing” in 1[c] and “parsing” in 1[d]. *See* EX1001, 17:9-21, 24:63-25:9. A POSITA would not have considered the other claimed steps to be a “stage of processing the received user utterance,” but steps associated with selecting and invoking a speech-enabled application. Lopresti ¶¶138-139. Moreover, PO interpreted this limitation in its contentions to encompass any use of “prior probabilities and fuzzy possibilities” to “understand user intent and references,” without identifying any “stage of processing the received user utterance.” EX1013, 7-8. This Petition addresses this limitation as to 1[c] and 1[d].<sup>3</sup> To the extent other limitations require the use of *prior probabilities or fuzzy possibilities*, those are separately addressed in

---

<sup>3</sup> Petitioner reserves the right to argue alternative interpretations in other proceedings.

the respective limitation, which a POSITA would have understood were similarly maintained as part of the proposed combination. Lopresti ¶139.

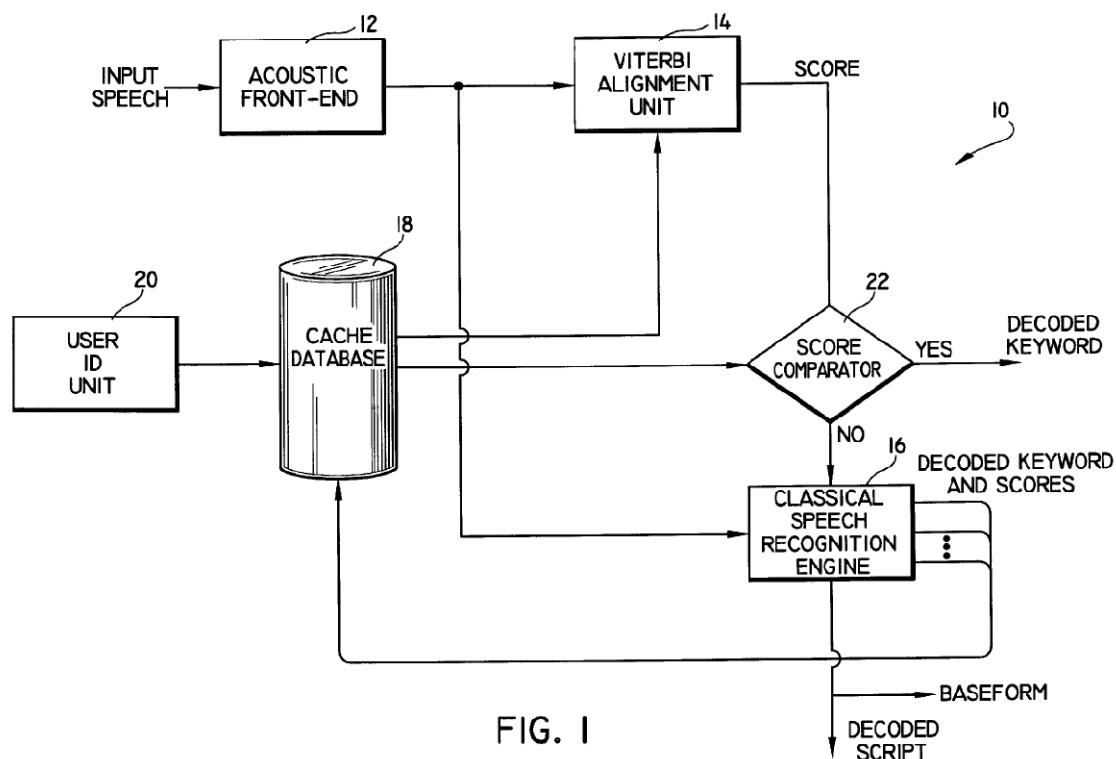
**First**, to the extent “maintaining” applies to 1[c], a POSITA would have found Ross with Ittycheriah renders this limitation obvious. Lopresti ¶¶140-152. As discussed in Section VII.A, Ross discloses well-known “speech recognition products” that could “convert speech into text strings.” EX1005, 1:17-22; Lopresti ¶141. Ittycheriah is such product also developed by IBM®. EX1007, 1:10-15; *id.*, [73]; Lopresti ¶142.

Ittycheriah discloses methods for “performing speech recognition on speech segments frequently input by a user.” EX1007, Abstract; Lopresti ¶142. Ittycheriah discloses “storing data in a cache database,” in which the cache “contains the word history of a keyword (name, command or sentence) as well as the likelihood (score) obtained when the word was added to the cache.” EX1007, 2:58-66. Ittycheriah explains the “likelihood score” is created as part of the “Viterbi alignment unit,” using a “probabilistic pattern matching technique using Hidden Markov Models (HMMs)” which “provid[es] different scores for different hypotheses.” EX1007, 3:54-60, 3:64-4:4. Lopresti ¶¶143-144.

By June 2002, HMMs were well-known and ubiquitous in the field of speech recognition. *See* EX1019; Lopresti ¶145. HMMs use probability distributions to determine the maximum likely sequence of transitions between a starting state of an

input and the probability of a given output. *See, e.g., id.*, 17-25; Lopresti ¶¶44-45. Therefore, Ittycheriah's cache database that stores "likelihood scores" for keywords *maintains a set of prior probabilities or fuzzy possibilities* that is *usable* to recognize words or phrases in the received utterance. Lopresti ¶147.

The set of prior probabilities maintained in Ittycheriah's cache database is also *dynamic*. Lopresti ¶152. Ittycheriah describes its cache database as being created from and constantly updated through the classical speech recognition engine, item 16 in Figure 1:



EX1007, Fig. 1, t 2:66-3:18, 4:33-59, 1:59-2:10; Lopresti ¶¶148-151. Upon receiving the input speech (*received utterance*), Ittycheriah applies a probabilistic pattern

matching process via the Viterbi alignment unit 14 to assign various likelihood scores to input speech. EX1007, 4:33-39; Lopresti ¶¶148-50.

The resulting scores are then “presented to the score comparator 22,” which compares the new scores relating to input speech to those of decoded keywords that “were previously stored in the cache database.” EX1007, 4:38-43; Lopresti ¶¶150-151. “If the new scores are not within a margin, or are less than any of the stored scores [in the cache database], then the decoded keywords in the cache are rejected and the input speech segment is subjected to the complete vocabulary of the classical speech recognition engine 16 whereby a decoded script is generated.” EX1007, 4:48-53; Lopresti ¶150. “Subsequently, the decoded word(s) resulting from the recognition performed by the engine is then preferably added to the cache along with associated scores.” EX1007, 3:16-18, 4:57-59, 8:3-5; Lopresti ¶151.

**Second**, to the extent “maintaining” applies to 1[d], Ross with Ittycheriah discloses this limitation. Lopresti ¶155 Ittycheriah’s “cache database” uses a dynamic set of HMM derived probabilities to create a “likelihood score” during the initial step 1[c]. Ittycheriah then outputs a decoded keyword that Ross’s conversation manager uses to determine a meaning (including determining a context) of the utterance. Accordingly, Ross with Ittycheriah maintains the same set of prior probabilities for 1[c] as in 1[d]. Lopresti ¶¶154-155.

Ross discloses this limitation in a second way. Lopresti ¶¶156-159. Ross discloses a “conversation manager” that “combines an analysis of the utterance with information on the state of the desktop and remembered context from previous recognitions to determine the intended target of the utterance.” EX1005, 5:50-60; Lopresti ¶156. Ross thus *maintains* an “ordered list of applications (e.g., a context list 62),” which is *a set of prior probabilities* that “may prioritize the contexts based on the access characteristic.” EX1005, 3:1-2, 2:47-49, 8:46-53; Lopresti ¶¶156-157. Ross’s context list is also *dynamic*, constantly being reordered “when (1) “an application 26 gains window focus, it will move to the head of the list 62” and/or (2) “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.” EX1005, 6:53-61; 7:9-15, 34-35; Lopresti ¶158.

**4. 1[c]: “recognizing...”**

Ross with Ittycheriah discloses this limitation. Lopresti ¶¶160-167. Ittycheriah acts as the speech engine Ross uses to *recognize words and phrases contained in the received utterance* and output a “recognition message” which is a “speech representation recognized by the speech engine.” EX1005, 6:12-15, 7:12; Lopresti ¶161. Ittycheriah recognizes words and phrases contained in a received utterance using “keywords” stored in a cache database that are “frequently used by a user, as well as score information attributable to the keywords.” EX1007, 1:55-

58; Lopresti ¶162. Ittycheriah defines “keyword” to “include such *words and phrases* as names, commands, and sentences....” EX1007, 2:16-19. Ittycheriah also discloses a “classical speech recognition engine” that includes a “large scale language database (vocabulary),” such that any utterance not recognized using the “cache database” is “subjected to the complete vocabulary of the classical speech recognition engine.” EX10073:62-64, 4:48-53; Lopresti ¶¶162-163.

A POSITA would have understood that cache databases and large-scale language databases of “words and phrases” that include “names, commands, and sentences” (like Ittycheriah’s) are both dictionary and phrase tables. EX1026, 3; EX1025, 3; EX1019, 73 (“Every speech recognizer is based upon a vocabulary...that includes only the words that can appear in the transcribed text.”); Lopresti ¶¶163-164. For example, IBM’s ViaVoice™ product defined the term “vocabulary” as used in its speech recognition product to be “a list of words provided with the product or obtained separately which can be recognized correctly when spoken by a user.” EX1017, 117; Lopresti ¶165. The ViaVoice™ product also allowed users to include phrases, rather than individual words. EX1017, 68 (“Add words as phrase”); Lopresti ¶165 .

## **5. 1[d]: “parsing...”**

Ross with Ittycheriah discloses this limitation. Lopresti ¶¶168-175. The output of Ittycheriah’s speech recognition engine is at least one “decoded

keyword...representative of the input speech segment.” EX1007, 2:5-7; Lopresti ¶169. This “decoded keyword” thus contains *one or more keywords contained in the recognized words and phrases*, including “names, commands, and sentences[.]” EX1007, 2:16-19; Lopresti ¶169. Ittycheriah’s “decoded keyword” therefore corresponds to the “recognition message” in Ross, which is output by the speech engine because it is a “speech representation recognized by the speech engine 22.” EX1005, 7:1-2; EX1007, 4:63-53; Lopresti ¶170. A POSITA would have thus found it obvious to use Ittycheriah’s “decoded keyword” as a recognition message in Ross’s context manager to *determine a meaning of the utterance by determining a context for the at least one request contained in the utterance*. Lopresti ¶171. Ross’s Figure 5 illustrates the process to determine a context for requests using recognition messages:

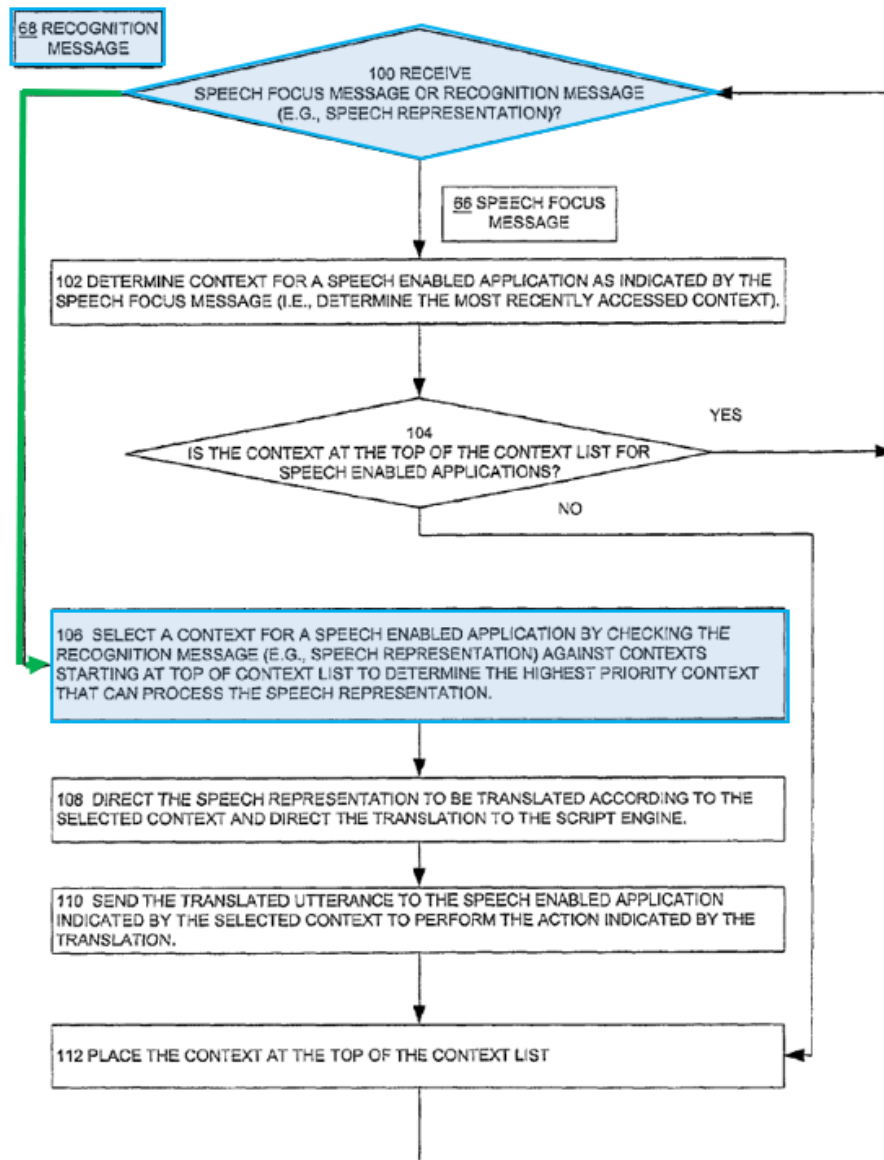


FIG. 5

**EX1005, Fig. 5 (annotated)**

Lopresti ¶171.

Specifically, Ross’s “context’s manager 50 receives the recognition message 68 and selects a context 70 for a speech enabled application 26 by checking the

speech representation against context 70.” EX1005, 7:17-21. Ross discloses that the “context manager” uses “the grammar associated with each context 70 to determine if the recognition message 68 is capable of being processed by that grammar.” EX1005, 7:23-26; Lopresti ¶171. Ross determines a context for the request contained in the input speech (*utterance*) *based on one or more keywords in the request*. Lopresti ¶172. For example, Ross discloses an “electronic mail application” that accepts keywords and phrases from the user such as “Open the first message,” “Create a message,” “Send this,” and “Print it,” based on the following sample grammar:

|                                                         |    |
|---------------------------------------------------------|----|
| <hr/>                                                   |    |
| <mail> = do I have any messages                         | 60 |
| open <message>                                          |    |
| create a message                                        |    |
| send <message>                                          |    |
| print <message>.                                        |    |
| <message> = the? <nth> message   it   this.             |    |
| <nth> = first   second   third   fourth   fifth   . . . | 65 |
| <hr/>                                                   |    |

EX1005, 7:60-66; Lopresti ¶172. Although Ross uses a Backus Naur Form grammar in the above embodiment, Ross recognizes that “another type of grammar, lists of keywords or phrases, hashed lists of words, or other suitable approaches” can be used to “allow identification of a phrase as acceptable to a speech-enabled application.” EX1005, 9:14-20; Lopresti ¶173 . A POSITA would have therefore understood that Ross determines a context for the at least one request contained in

the utterance based on one or more keywords contained in the recognized words and phrases. Lopresti ¶¶174-175.

**6. 1[e]: “selecting...”**

Ross discloses this limitation. Lopresti ¶¶176-184. The '209 describes “domain agents” as “autonomous executables that receive process, and respond to user questions, queries, and commands,” that “provide complete, convenient and redistributable packages or modules of functionality.” EX1001, 2:50-59. A POSITA would thus understand a “domain agent” to include an autonomous executable software module specific to each application area. Lopresti ¶177.

Ross discloses “external applications 26” that “communicate with the speech center 20, a speech engine 22, and active accessibility module 24.” EX1005, 4:16-21. Each of these external applications have a defined grammar, “along with associated translations for a script engine 38.” EX1005 6:29-31; Lopresti ¶178. And each of these external applications *receive, process, and respond to requests based on the determined meaning* of the request (i.e., based on the determined context). EX1005, 2:22-36; Lopresti ¶178. Ross discloses a task manager that “provides the capability to proceed with multiple execution requests simultaneously, to queue up additional script commands for busy applications 26, and to track the progress of execution.” EX1005, 5:1-5; Lopresti ¶179. A POSITA would have thus understood the external applications are autonomous executables triggered by the script

commands issued by Ross's task manager. Lopresti ¶179. Accordingly, each external, speech-enabled applications is a *domain agent* as in the '209. Lopresti ¶180.

Ross further discloses *selecting at least one domain agent based on the determined meaning*. Lopresti ¶181. Ross “maintains a context list of speech enabled applications” that the user has accessed and teaches “deciding the target application (i.e., selecting a domain agent) for a received utterance based on the determined meaning. *See* limitation 1[d]. After Ross determines the context for the received utterance, it selects an application (*domain agent*). Lopresti ¶181. For example, Ross teaches that if only one grammar for a speech-enabled external application accepts the received utterance, “that associated application 26 is the one that should be targeted to receive the corresponding command.” EX1005 8:33-36; Lopresti ¶182. Ross further selects the domain agent based on “recency of access” when more than one grammar of an external application can accept the received utterance. EX1005, 8:37-40, 9:7-13; Lopresti ¶¶181-184.

**7. 1[f]: “formulating...”**

Ross discloses this limitation. Lopresti ¶¶185-188. After selecting the target external application (*domain agent*), Ross translates the received utterance for processing by the external application. EX1005, 5:55-57, 6:15-18, 6:48-51, 7:26-29, 9:9-13, Abstract; Fig. 5 (step 108); Lopresti ¶186. This translation is associated

with “grammar defined for each application.” EX1005, 6:29-31. A POSITA would thus understand Ross’s “translation” of the received utterance *formulates the at least one request (command) contained in the utterance in accordance with a grammar used by the selected domain to process requests associated with the determined context*. E.g., EX1005, 6:48-51; Lopresti ¶¶187-188.

**8. 1[g]: “invoking...”**

Ross discloses this limitation. Lopresti ¶¶189-192. Ross discloses that once translated (*formulated*), the “appropriate script engine 38 calls and [the translated script] is dispatched to the target application.” EX1005, 5:55-57; Fig. 2; Lopresti ¶190. The external application (*domain agent*) then processes the command by executing the call. EX1005, 6:15-17, 10:19-23; Lopresti ¶190. A POSITA would have understood when the command is dispatched to the target application (*domain agent*) for processing, it *invokes the selected domain agent to process the formulated request*. Lopresti ¶¶191-192.

**9. 1[h]: “presenting...”**

Ross discloses this limitation. Lopresti ¶¶193-195. Ross provides several examples of *presenting results of the processed request that were generated as a result of the invoked domain agent processing the formulated request*. EX1005 2:22-30, 2:31-36 (launching word processing application in response to request to take letter), 8:47-9:6 (describing print requests made to electronic mail and calendar

applications resulting in the printing of messages and appointments, respectively); Lopresti ¶¶194-195.

**D. Dependent Claim 4**

Ross with Ittycheriah discloses this limitation for the same reasons as claim 1, and below for 4[a]-4[c]. Lopresti ¶196.

**1. 4[a]: “matching...”**

Ross with Ittycheriah discloses this limitation. Lopresti ¶¶197-202. As discussed for 1[d], the Ittycheriah’s output is the recognition message input into Ross’s conversation manager (part of *parsing*). See Section VII.C.5. As discussed for 1[c], Ross’s “ordered list of applications (e.g., a context list 62),” is *a dynamic set of prior probabilities* that “may prioritize the contexts based on the access characteristic.” See Section VII.C.4; EX1005, 3:1-2, 2:47-49; Lopresti ¶198. Ross further discloses that each application in the context list has *associated keywords* as a design choice. EX1005, 9:16-20. Ross further discloses *matching the keywords contained in the recognized words and phrases* to those keywords in the context list, by, for example, describing how a request to “print an appointment” would be matched to keywords associated a calendar application. EX1005, 8:1-24; Lopresti ¶¶199-202.

**2. 4[b]: “scoring...”**

Ross discloses this limitation. Lopresti ¶¶203-208. As discussed for 1[b], Ross uses “recency of relevant access to the context” (i.e., a relevance score) as an “access characteristic” such that Ross “is biased to send commands to the most recently accessed application.” EX1005, 8:46-53; Lopresti ¶204. Ross uses the example of the spoken command “print it,” which can match an electronic mail application or calendar application. EX1005, 8:1-24. Ross scores the most recent, relevant context as the highest, forming a context list (*a context stack of one or more recent context*). Ross explains: “If the sentence is ‘print it’” such that multiple grammars can accept the utterance, “[t]he context manager 50 therefore has to make a choice by referring to the context list 62 of applications in order of recency of access.” EX1005, 8:37-46, 8:54-9:13. A POSITA would have thus understood Ross scores the possible contexts based on the context stack of one or more recent contexts, by “test[ing] the utterance against” multiple grammars in the context list and scoring the possible contexts as not matching or matching. EX1005, 8:37-46, 8:54-9:1; Lopresti ¶¶205-208.

**3. 4[c]: “determining...”**

Ross discloses this limitation. Lopresti ¶¶209-213. As discussed for 1[e], Ross scores possible contexts based on whether the application can accept the received utterance. EX1005, 8:37-40, 9:7-13; Lopresti ¶210. A POSITA would have

understood by finding a context that accepts the utterance, that context must have a *score satisfying a predetermined confidence level*. Lopresti ¶¶211-213. In contrast, a POSITA would have understood a context that does not match the utterance well enough would not meet the predetermined confidence level. Lopresti ¶212.

## **VIII. GROUND 2:**

### **A. Motivation to Combine**

A POSITA would have been motivated to use Ross018's sophisticated parser with Ross to "provide the speech center system with more knowledge about a domain," and aid the speech center in "perform[ing] a form or reasoning about the user's utterances that it receives and how best to respond to them." EX1009, 2:3-8; Lopresti ¶214. Ross and Ross018 share Provisional Application No. 60/261,372. EX1005, [3]; EX1009, [73]. A POSITA would have been motivated to consult patents related to Ross, including Ross018, to determine additional features or qualities to consider in developing a natural language processing system. Lopresti ¶215.

The shared provisional explains that the two patents are complementary, with Ross018 being "basically an expansion" of Ross's conversation manager. EX1016, 8. By increasing the sophistication and complexity of the parser, Ross018 provides more natural interaction between the user and Ross's conversation manager. Lopresti ¶216. A POSITA would have thus been motivated to add Ross018's

parsing features to Ross to better respond to user's utterances by adding application domain information and additional reasoning to generate a response. Lopresti ¶¶217-218.

**B. Reasonable Expectation of Success**

Ross018 is an explicit complement to Ross. A POSITA would have had the skill to implement and the reasonable expectation of success in achieving such a combination because it would have involved a combination of known technologies (speech recognition and processing natural language requests) according to known methods (speech recognition engines that convert speech into text strings that can be used by software applications) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various speech-enabled applications on a computer as well as making use of knowledge of a user's domain, plans, tasks, and processes in parsing the user's natural language utterance. Lopresti ¶¶219-222.

**C. Claim 6**

To the extent limiting, Ross and Ittycheriah with Ross018 discloses this limitation for the same reasons as claim 1, and below for 6[a]-6[d]. Lopresti ¶223.

**1. 6[a]: “determining...”**

Ross with Ross018 discloses this limitation. Lopresti ¶¶224-239. Ross018 adds to Ross's “conversation manager” (part of *parsing*) a “semantics analysis

module” that “processes the spoken recognized spoken utterance 15 using a grammar specification 90 based on the domain model 70 to produce an initial semantic representation based on the recognized spoken utterance.” EX1009, 11:61-65; Lopresti ¶230. In Ross018, each speech-enabled external application (*domain agent*) can “load custom grammars or define task specific vocabulary” to Ross’s speech center. EX1009, 5:21-30. Using those grammars or vocabulary, after speech recognition, “the conversation manager 28 combines an analysis of the utterance with information on the state of the desktop and remembered context from previous recognitions to determine the intended target of the utterance.” EX1009, 5:63-56. “The utterance 15 is then translated into the appropriate script engine 38 calls and dispatched to the target application 26.” EX1009, 5:67-6:2; Lopresti ¶¶230-231.

In translating (*formulating*) the request contained in the utterance to the *grammar used by the domain agent*, Ross018 *determines one or more required values and one or more optional values in the grammar used by the selected domain agent*. Lopresti ¶232. Ross018’s conversation manager “converts recognized utterances 15 into sequences of script calls to frame-building functions,” which are executed to create “an initial semantic representation of the utterance 15.” EX1009, 9:51-62. Ross018 uses these initial semantic representations to create a series of “attribute-object-value triples, which are termed ‘propositions.’” EX1009, 9:62-63. These “values” include “action,” the object or “patient” of the action (“the thing

acted upon by an action”), and the “destination,” and represent “the kinds of objects, attributes, and operations that the application 26 makes available” to Ross018’s speech center, which would include both *required* and *optional values* to process the request. EX1009, 10:5-15, 7:35-51; Lopresti ¶¶233-234. A POSITA would have understood and expect that grammars accepted by speech-enabled applications would include both required and optional values. Lopresti ¶¶236-238.

The ’209 does not distinguish between “required value[s]” and an “optional value[s].” EX1001, 18:10-12, 18:49-52, 24:58-61, 25:33-36. Nevertheless, because the triples in Ross018 would be inclusive of all values contained in the utterance, a POSITA would have understood such triples would include both all optional and required values. EX1005, 9:49-10:35; Lopresti ¶239. These attribute and object values in Ross018 align with “values” in the ’209 as criteria that describe different attributes, such as “gender” and “occupation” as personal profile values. EX1001, 14:32-50, 18:12-16; Lopresti ¶239.

## **2. 6[b]: “extracting...”**

Ross with Ross018 discloses this limitation. Lopresti ¶¶240-245. The ’209 uses “criteria,” “parameters” and “values,” somewhat interchangeably. *See*

EX1001, 11:59-64, 14:10-16.<sup>4</sup> However, a POSITA would have understood “criteria” includes the attributes of a request (e.g., location) and “parameter” includes the attribute’s value (e.g., Virginia). Lopresti ¶241.

Ross018 discloses *extracting one or more criteria and one or more parameters from one or more keywords contained in the one or more recognized words or phrases*. Lopresti ¶242. In generating its “propositions” Ross018 converts the recognized utterance into frame representations that serve as input to the reasoning facility. EX1009, 9:59-62; Lopresti ¶242. The “criteria” of the ’209 includes “attributes” of Ross018’s attribute-object-value triples. *Id.* Ross018 provides an example of the command “make this column green,” in which it extracts from the keywords in the command the criteria (make, object, color) and the associated parameters (column, green) as in the below frame structure:

---

<sup>4</sup> PO points to the same evidence for “values,” “criteria,” and “parameters” in its contentions with no differentiation. EX1013, 18-25.

```
5  -----  
    Begin("command")  
      AssociateValue("action")  
      Begin("action")  
        AssociateClass("make")  
        AssociateValue("patient")  
10     Begin("thing")  
        AssociateClass("column")  
        End("thing")  
        AssociateValue("destination")  
        AssociateParameter("green")  
      End("action")  
15  End("command")  
    -----
```

EX1009, 9:59-10:35; Lopresti ¶¶242-243.

Ross018 develops its attribute-object-value triples *using procedures sensitive to the determined context*. Lopresti ¶244. Before processing the initial semantic representation, Ross's context manager determines the recognized utterance is associated with a context of a specific external application. Ross018 further discloses that its frame structures (extracted from the attribute-object-value triples) are based in part on the "domain model," which includes the concepts, grammatic and semantic specifications, of the speech enabled applications. EX1009, 6:40-47, 7:10-21. By accounting for the domain model, Ross018 uses procedures sensitive to the determined context to extract its criteria and parameters. Lopresti ¶¶244-245.

### 3. 6[c]: "inferring..."

Ross with Ross018 discloses this limitation. Lopresti ¶¶246-253.

Ross018 discloses an inference agent that *uses a dynamic set of prior probabilities or fuzzy possibilities*. Lopresti ¶¶249-250. Ross018 describes the

reasoning facility as “a goal-directed rule based system composed of an inference engine, memory, rule base and agenda.” EX1009, 10:36-51. The “rules consist of some number of condition propositions and some number of action propositions,” while “[a] goal is a proposition that may contain a variable for one or more of its elements.” *Id.* The reasoning facility thus creates attribute-object-value triples in which one of the elements is a “variable” that must be inferred. Lopresti ¶250. The reasoning facility *uses fuzzy possibilities* by scanning through already registered rules to see what, if any, rules include “actions unifying with the goal” searching for a match. EX1009, 10:51-57; Lopresti ¶¶195-196.<sup>5</sup> If matched, “[t]hese become new goals for the inference engine of the reasoning facility 52 to achieve.” EX1009, 10:57-58. A POSITA would have thus understood Ross018’s reasoning facility *infers one or more further criteria and one or more further parameters associated with the request.* Lopresti ¶¶249-252; EX1009, 11:8-16.

The fuzzy possibilities are *dynamic*, such that when “no appropriate operations can be found to satisfy a goal,” then “a question to the user will be generated,” which become “new goals for the inference engine of the reasoning facility.” EX1009, 10:59-64. These goals are based on the “conversation record,”

---

<sup>5</sup> The ’209 does not define *fuzzy possibilities*. PO’s contentions implies it includes any logic to infer criteria. EX1013, 24-25.

which purges “when it is no longer relevant to active goals and after some predefined period of time has elapsed.” EX1009, 11:2-7. Because the record is constantly evolving, a POSITA would have understood the logic of the inference engine is *dynamic*. Lopresti ¶253.

#### 4. 6[d]: “transforming...”

Ross with Ross018 discloses this limitation. Lopresti ¶¶254-259. Ross018’s reasoning facility “generates a response by analyzing the utterance representation received from the syntax manager 62,” which includes “an action to be executed, such as a script call or command to an external application 26, (e.g., a call to open a message).” EX1009, 12:15-20. The generated propositions (attribute-object-value triples) “can contain embedded script function calls” for external applications (*domain agents*). A POSITA would have thus understood that in generating an action to be executed by a script call or command, Ross018 *transforms the extracted criteria and parameters and the inferred criteria and parameters into one or more tokens having a format compatible with the grammar used by the domain agent*. Lopresti ¶¶255-258. A specific script call or command to be processed by an external application with the criteria and parameters of the request would be understood by a POSITA to be a token, because it is any individuated or “non-reducible text element in data.” EX1024 (defining “token”); Lopresti ¶257.

Moreover, a POSITA would have found it obvious based on Ross018 that the script call or command (as indicated by one or more *tokens*) must include *all of the required values* and *one or more of the optional values associated with formulating the request in the grammar used by the domain agent* for the external application to provide a proper response to the user's command, as discussed. Otherwise, "[w]hen no appropriate operations can be found to satisfy a goal, a question to the user will be generated." EX1009, 10:59-61. Accordingly, a POSITA reading Ross018 would understand in formulating the script call or command (*token*), all required values and optional values would be included or Ross018 would prompt the user for more information, including for additional required values. Lopresti ¶¶258-259.

## **IX. GROUND 3:**

### **A. Motivation to Combine**

A POSITA would have been motivated to add Aridor's "knowledge agents" that perform Internet metasearching to Ross's external applications (*domain agents*) to broadly collect information from the Internet and rank the results' relevance. Lopresti ¶¶260-264. Ross discloses that its external applications "perform the function of the command" in the natural language utterance but does not address the Internet. EX 1005, 9:44-50, 12:6-13. However, by 2002, a POSITA would have understood the Internet provided a robust source of information. Lopresti ¶¶260-261. Accordingly, a POSITA would have been motivated to look for and rely on

Internet searching mechanisms in retrieving and processing responses to a user's queries (like Aridor, also developed by IBM). Lopresti ¶262.

A POSITA would have understood that Aridor's metasearch functionality to be compatible with Ross, as Aridor operates by using domain-based knowledge agents, such that each of Ross's external applications could have its own knowledge agent. Lopresti ¶262. Combining Aridor with Ross would also improve Aridor's methods by obviating the need for a user to "choose the knowledge agent that has the relevant specialization" to perform the search, but rather to have Ross select the domain agent as in Section VII.C.6-7 (Ground 1, 1[e]-1[f]). EX1010, 3:53-59. A POSITA would want to add Aridor's "ranking and filtering" features to Ross, which process the results of the user's query to find the most relevant information. Lopresti ¶263. A POSITA would have also been motivated to use Aridor's metasearch functionality because its knowledge agents "make use of persistent, acquired knowledge in the domain that is defined and then refined by a user," based on the user's queries made over time. EX1010, 3:65-4:2, 14:56-15:19; Lopresti ¶264.

#### **B. Reasonable Expectation of Success**

Aridor and Ross are IBM systems. Aridor expands on Ross's ability to process natural language commands and questions by extending its Internet search capabilities. Lopresti ¶¶265-267. A POSITA would have the skill to implement and the reasonable expectation of success in achieving such a combination because it

would have involved a combination of known technologies (speech recognition and Internet searching based on text queries) according to known methods (speech recognition engines that convert speech into text strings and using text strings to perform Internet searching) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various speech-enabled applications to retrieve information from the Internet. Lopresti ¶¶266-267.

### **C. Claim 7**

To the extent limiting, Ross and Ittycheriah with Aridor discloses this limitation for the same reasons as claim 1, and below for 7[a]-7[b]. Lopresti ¶¶268-269. Ross provides the example of a user “ask[ing] what is the current stock price for a particular company,” which is both a question (what is the stock price) and a command (retrieve the stock price). EX1005, 2:25-30; Lopresti ¶269.

#### **1. 7[a]: “directing...”**

Ross with Aridor discloses this limitation. Lopresti ¶¶270-273. Implementing Aridor’s knowledge agent as Ross’s external application provides the details of processing and returning results to the user’s query. For example, Aridor discloses that “[a] user 22 uses a computing device 24 to connect to a network 28, typically the Internet, and to conduct a search among information repositories, such

as Web sites 26,” which can “any suitable general purpose computer processor having a connection to network 28.” EX1010, 8:1-12; Lopresti ¶¶271-273.

**2. 7[b]: “querying...”**

Ross with Aridor discloses this limitation. Lopresti ¶274. As discussed for 7[a], Ross with Aridor allows Ross’s external applications (*domain agents*) to query a network information source, such as the Internet to *resolve information requested in the at least one question*. Section IX.C.1; Lopresti ¶274.

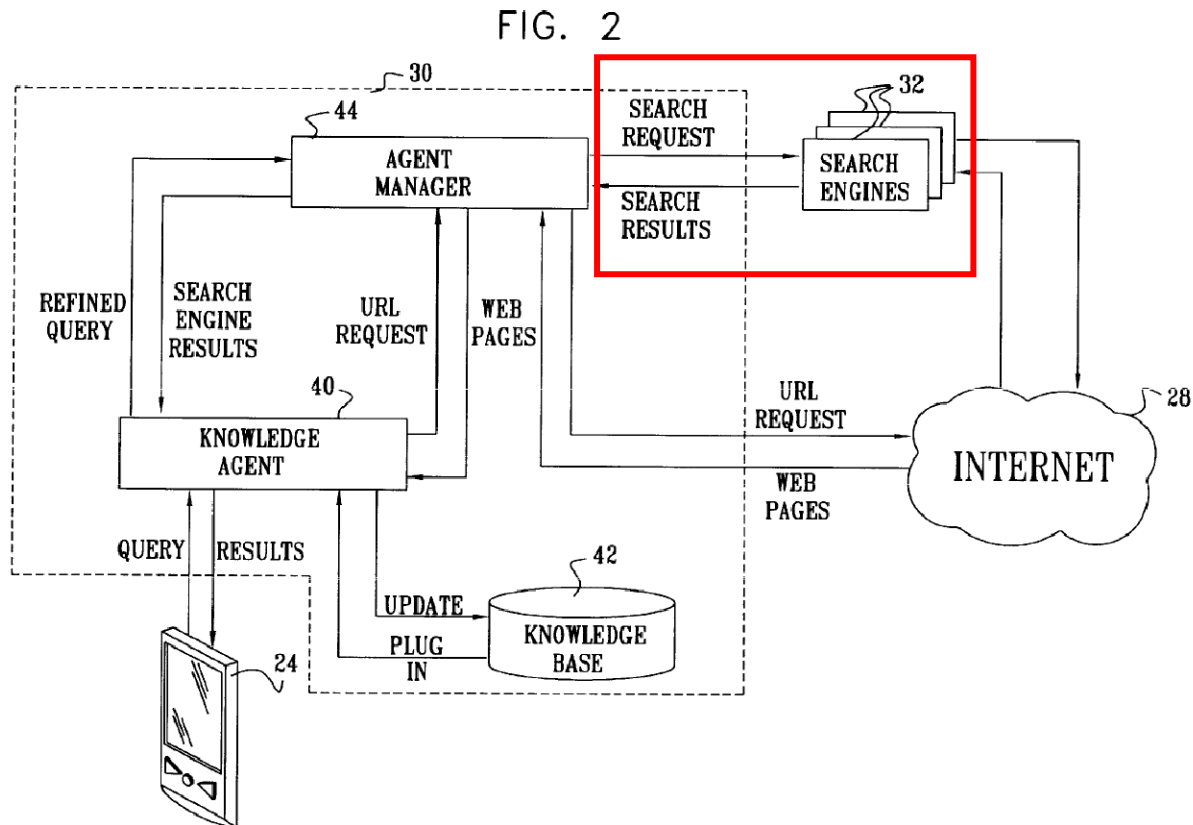
**D. Claim 8**

To the extent limiting, Ross and Ittycheriah with Aridor discloses this limitation for the same reasons as claim 1 and 7, and below for 8[a]-8[c]. Lopresti ¶275.

**1. 8[a]: “submitting...”**

Ross with Aridor discloses this limitation. Lopresti ¶¶276-279. As discussed, a POSITA would be motivated to add to Ross’s external applications (*domain agents*) the features and functionalities of Aridor’s “agent manager” and “knowledge agents” (or “knowledge agent program”) to facilitate searching a corpus of documents. Section IX.A. Aridor discloses that its knowledge agents “receive[] the user’s query and carries out the search by simulating the steps involved in the conventional interactive search process.” EX1010, 3:33-35. Because Aridor searches over multiple search engines, it *submits multiple asynchronous queries to*

*the one or more network information sources.* EX1010, 3:33-35, 4:10-16, 8:22-28, 8:45-59, Fig. 2. This is also in Figure 2, submitting a search request to multiple search engines:



**EX1010, Fig. 2 (annotated)**

A POSITA would have understood that a metasearch that sends the same search requests to different search queries *submits duplicate queries to distinct information sources.* Lopresti ¶¶276-279.

**2. 8[b]: “a synchronously receiving...”<sup>6</sup>**

Ross with Aridor discloses this limitation. Lopresti ¶¶280-282. Aridor discloses the user’s query is “submitted to the user’s choice of one or more general-purpose search engines 32, and a user-specified number of results are collected from each of these engines.” EX1010, 10:18-21, 9:6-7, Fig. 2. A POSITA would have understood each search engine sends results *asynchronously*. Lopresti ¶¶281-282. For example, Aridor employs general-purpose search engines such as AltaVista, Google or Hotobot, which are separate search engines that operate independently from one another. EX1010, 8:52-59; Lopresti ¶282.

**3. 8[c]: “scoring...”**

Ross with Aridor discloses this limitation. Lopresti ¶¶283-287. Aridor discloses *scoring a relevance of each of the asynchronously received results...to determine one or more best responses to the at least one question*. The results are returned to Aridor’s knowledge agent for “ranking and filtering.” EX1010, 8:56-57. The knowledge agent “traverses the retrieved pages...and ranks them such that the most relevant pages for the user will be listed first in the result” following receipt. EX1010, 11:19-34. “Textual similarity scores 68 are computed to measure the

---

<sup>6</sup> For the purposes of this Petition, Petitioner assumes this claim recites “asynchronously.”

relevance of the pages retrieved both to the specific query and to the agent's specialized domain," and topological aspects are scored based on whether a site has been "proven to be highly relevant to many of the queries that the user 22 has submitted to the knowledge agent 40." EX1010, 11:19-12:67 (textual scores), 13:1-15:11 (topology scores), Fig. 3. The two scores are then "normalized and summed to give overall combined scores 74 for all of the pages," and the search results are then "ranked in the order of their combined scores." EX1010, 11:32-34; Lopresti ¶¶284-285.

*Aridor uses a dynamic set of prior probabilities or fuzzy possibilities to score the relevance of the results.* Lopresti ¶286. In computing both "textual score" and "link topology score," Aridor employs a "knowledge base" that "contains a bounded collection of ranked sites and aggregate profile of the textual content of these sites." EX1010, 11:60-12:13, 14:24-34. "Each site kept in the knowledge base is assigned a history score, which reflects the site's relevance to the domain through the course of the use of the agent," and sites with high history scores are assumed to be of higher quality. EX1010, 14:35-43. A POSITA would have thus understood the history scores represent *a prior probability or fuzzy possibility* that the site will provide a relevant result to a user's query. Lopresti ¶286. Aridor's knowledge base is also *dynamic* as it includes an "evolutionary adaption mechanism," that over time increases the weight of history for more mature agents over time and "prune[s] out"

old sites “while new and useful sites are added.” EX1010, 14:35-43, 14:56-15:19; Lopresti ¶287.

**E. Claim 9**

To the extent limiting, Ross and Ittycheriah with Aridor discloses this limitation for the same reasons as claims 1, 7-8, and below for 9[a]-9[d]. Lopresti ¶288.

**1. 9[a]: “determining...”**

Ross with Aridor discloses this limitation. Lopresti ¶¶289-291. Upon receiving the formatted script call from Ross, Aridor’s knowledge agent processes the question as a text query (“keyword-based query such as those typically submitted to a general-purpose Web search engine.”). EX1010, 9:42-49. Aridor creates a “query profile” that “consist[s] of each word in the text query (not including stop-words), along with its lexical affinities, and indicating the number of times each word appears in the query.” EX1010, 11:55-59. A POSITA would have understood, in creating the query profile, Aridor *determines one or more tokens required to formulate a response to the question* because the query profile includes the specific instances of a word or phrase that appears in the utterance for which values are sought (requested information). Lopresti ¶¶290-291.

**2. 9[b]: “extracting...”**

Ross with Aridor discloses this limitation. Lopresti ¶¶292-296. Aridor discloses submitting the query to “one or more general purpose search engines,” and collecting results (called “root set 54 of sites 26”). EX1010, 10:18-23. Aridor then adds to the “root set,” sites already known to the knowledge agent (“knowledge base”) as having relevant and quality information, and backward and forward sites, to those in the returned results, forming a “final site list” of potential results. EX1010, 10:40-11:18; Lopresti ¶293.

Aridor then *applies one or more scraping criteria* to the *asynchronously received* results returned from the “site list” to *extract one or more values for the determined tokens*. Aridor scrapes the results for both textual similarity and for topological aspects, assigning scores for both features. EX1010, 11:19-34, 11:46-51, 12:13-62, Fig. 3.

Aridor also applies a *scraping criteria* to the site list to compute a “link topology score,” based on “anchor text” (text description of linked page identified in the search result) and “anchor links” (links connecting knowledge base sites to non-knowledge-based sites), because the connected site “presumably answers the specific query.” EX1010, 13:14-43. Where there are “many neighboring sites” to root sites within the site list, the topology score has greater significance in

identifying an answer to the query (*value for the determined token*). EX1010, 14:11-23; Lopresti ¶¶295-296.

### 3. 9[c]: “evaluating...”

Ross with Aridor discloses this limitation. Lopresti ¶¶297-300. Aridor’s “textual score” and “link topology score” both use *a dynamic set of prior probabilities or fuzzy possibilities* through “domain textual scores” and “anchor links,” computed using Aridor’s dynamic knowledge base, as discussed above in Section IX.D.3 (Ground 3, limitation 8[c]). Aridor *evaluate[s] the extracted values* by “rank[ing] the found documents with respect to the relevance to the domain responsive to the textual scores and topological scores” using the same “knowledge base.” EX1010, 3:53-4:11, 4:20-30, 14:56-15:11; Lopresti ¶¶298-299. A POSITA would have understood that Aridor’s use of its knowledge base—the domain of interest—to increase the accuracy of results and find the most authoritative pages would *include resolving ambiguous, incomplete, or conflicting information associated with the extracted values*. Lopresti ¶¶299-300.

### 4. 9[d]: “selecting...”

Ross with Aridor discloses this limitation. Lopresti ¶¶301-303. Aridor evaluates the extracted criteria using a ranking algorithm that normalizes and sums the “textual score” and “link topology score” “to give overall combined scores 74 for all of the pages,” and the “search results, typically comprising the top few entries

on a final list of sites 76, [are] ranked in the order of their combined scores,” and “downloaded to the device.” EX1010, 11:19-34. In ranking and downloading the “top few entries,” a POSITA would understand Aridor to be *selecting one or more best values for the determined tokens from the evaluated values*, including a top or best value. Lopresti ¶¶302-303.

**X. GROUND 4:**

**A. Motivation to Combine**

A POSITA would have been motivated to use Baker’s speech recognition method as Hartono’s “automatic speech recognition system,” to recognize speech using acoustic and context probabilities. Lopresti ¶¶304-311. Hartono discloses that speech is input into an “automatic speech recognition” module, shown below (item 102a) in Figure 2:

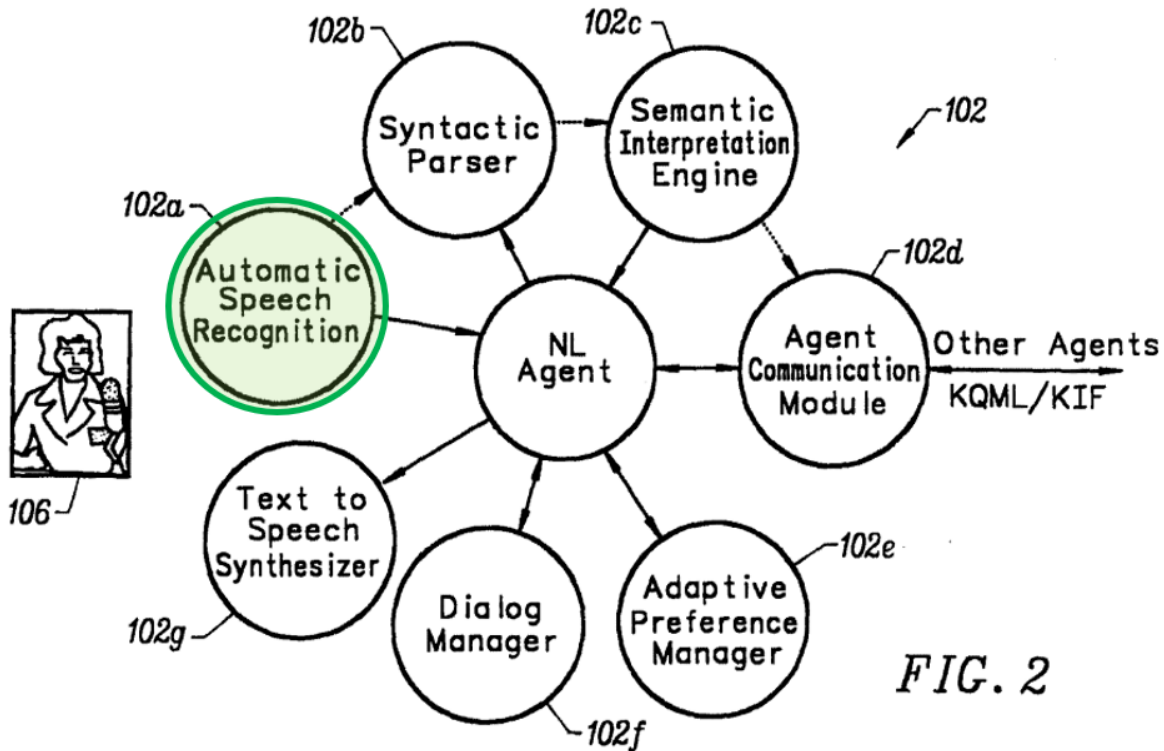


FIG. 2

EX1006, Fig. 2 (annotated)

Hartono recognizes that by 1999, “automatic speech recognition systems for speech input are readily and commercially available off the shelf.” EX1006, 11:4-12. Hartono thus discloses that “[a]ny suitable off the shelf speech recognition systems may be used as the automatic speech recognition system 102a.” EX1006, 11:4-12; A POSITA reading Hartono would have thus been motivated to look for available speech recognition systems to use to with Hartono. Lopresti ¶¶305-307.

Baker is such a speech recognition system developed by Dragon Systems, a well-known developer of such software. EX1008, [73]; Lopresti ¶308. A POSITA would have understood Baker’s speech recognition system would be compatible

with Hartono, as Baker discloses recognizing “words, or a series of words,” for “translat[ing] into machine readable form.” EX1008, 1:5-15; Lopresti ¶¶308-309. Hartono also employs the Government and Binding approach to parsing, which is based on the “implied syntactic structure in English language sentences.” EX1006, 12:1-10. Baker’s system similarly recognizes words based on context, and is thus compatible with Hartono’s syntactic parser, which identifies noun phrases, verb phrases, and adjective phrases. EX1006, 12:15-13:10; Lopresti ¶310.

A POSITA would have also been motivated to use Baker’s speech recognition system with Hartono as it recognizes words “on the fly” using existing and on-line reference sources. Lopresti ¶311. This allows Baker to avoid data-intensive statistical language models or data-training sets. EX1008, 3:46-4:56 (describing drawbacks of prior art systems based on “precomputed statistical counts” and “vocabulary training.”), 4:57-61, 6:18-22, 13:66-14:13. A POSITA would have been motivated to use Baker’s faster processing to more quickly respond to specific user requests. Lopresti ¶311.

#### **B. Reasonable Expectation of Success**

A POSITA would have had a reasonable expectation of success in combining Hartono’s and Baker’s teachings. Lopresti ¶¶312-314. Hartono’s is designed to use “[a]ny suitable off the shelf speech recognition system.” EX1006, 11:4-12. A POSITA would have had the skill to implement and the reasonable expectation of

success in achieving such a combination because it would have involved a combination of known technologies (speech recognition and processing natural language requests) according to known methods (speech recognition engines that convert speech into text strings) to yield the predictable result of a method for inputting natural language speech and outputting a text string to invoke various speech-enabled applications on a computer. Lopresti ¶¶313-314.

### **C. Independent Claim 1**

#### **1. Preamble**

To the extent limiting, Hartono discloses the preamble. Lopresti ¶¶316-318. Hartono discloses a “method for receiving commands and/or adaptively outputting results using a natural language speech interface[.]” EX1006, 8:8-10; *see id.*, 1:19-23, 4:20-25, 5:16-24, 10:13-20; Lopresti ¶¶317-318.

#### **2. 1[a]**

Hartono discloses this limitation. Lopresti ¶¶319-324. Hartono teaches “receiving commands and/or adaptively outputting results and responses using a natural language interface, such as a natural language speech interface.” EX1006, 4:9-11, 8:8-10; *see* EX1006, 1:20-23, 4:20-25, 5:11-12, 5:17-23, 25:20-23, 29:2-6; Lopresti ¶¶320-323. Hartono further discloses that the “user 160 communicates an input phrase, command or sentence 108 to the natural language agent 102 which processes the input sentence and sends the input sentence to the appropriate one of

the back-end application task agents 104a-d.” EX1006, 9:3-10. A POSITA would have understood a “command” to be *at least one request*. Lopresti ¶¶321-324.

### 3. 1[b]

As discussed for Ground 1, the ’209 patent does not define what is “each stage of processing the received user utterance.” *See* Section VII.C.3 (Ground 1, 1[b]). Accordingly, the Petition addresses this limitation as to 1[c] and 1[d]. To the extent other limitations require the use of *prior probabilities or fuzzy possibilities*, those are separately addressed in the respective limitation, which a POSITA would have understood were similarly maintained. Lopresti ¶326.

**First**, to the extent 1[b] applies to 1[c], Hartono with Baker discloses this limitation. Baker discloses methods “speech recognition, optical character recognition, machine translation, and other recognition systems.” EX1008, 6:29-32. Figs. 2, 3 (speech recognition system); Lopresti ¶¶327-328. As discussed in Section IV.D (Baker Overview), Baker discloses a “choice list” that is stored in memory (*maintains a set of prior probabilities or fuzzy possibilities*) which is used to recognize words or phrases in the received utterance.” Lopresti ¶¶329-332. Baker’s “choice list 20” is *dynamic* as it is created by the “choice list generator 14” in response to the input speech by “speech analysis unit 66” that is used for “analyzing the speech signals and for recording with a certain degree of probability, the words that make up the speech signals.” EX1008, 18:1-4, 19:53-20:29 (explaining how

“list of choice words 20” generated by “continuous speech utterance”); Lopresti ¶333.

Baker discloses this limitation in another way. Lopresti ¶¶334-342. Once “choice word 20” has been identified, Baker generates a “word list” that has “a semantic association with the choice word 20.” EX1008, 14:57-69. The “semantic association” for each choice word (also referenced as “passage words,” item 24 in Fig. 2) in the “word list” is derived from a “reference database 40” which can be a “dictionary, thesaurus, antonym dictionary, medical dictionary, legal dictionary, scientific and engineering dictionary, or other type of reference source.” EX1008, 20:55-58; Lopresti ¶335. Baker “performs a matching function to calculate the number of matched words” between those word recognized by the speech analysis unit 66 with high confidence (“context words”) with the words in the “word list” (“passage words”). EX1008 21:19-47. By looking at the “relative frequency of matches between the “context words” and the “passage words,” the data processor calculates a “probability signal 28 representative of the likelihood that the choice word or phrase 20 substantially represents the utterance currently being recognized.” EX1008, 21:19-47, 21:49-22:6, Figs. 2-4; Lopresti ¶¶336-337.

Baker then generates a “rank signal 30” that “represents the likelihood...that a choice word 20 is substantially similar to” the word being recognized” as a function of “context matching and acoustic model matching” probabilities. EX1008, 23:22-

27; *see id.* 21:48-22:6 (computing ranking signal 30 per techniques “well known in the field of mathematics and any of the techniques capable of generating a probability signal given a set of constraints”). Accordingly, a POSITA would have understood Baker *maintains* at least three *dynamic sets of prior probabilities* used in 1[c]: (1) prior probabilities 68 based on an “acoustic model,” (choice list) (2) prior probabilities based on context 28 (word list), (3) and rank signals 30, that are probability signals as a function of signals 68 and 28. Lopresti ¶¶338-342.

**Second**, to the extent this limitation applies to 1[d], Hartono with Baker discloses this limitation. Lopresti ¶¶343-344. In the combination, Baker’s output is the recognized words of a spoken utterance “just as if the data were being typed.” EX1008, 1:4-7, 1:32-36. Those recognized words are then inputted into Hartono’s natural language “syntactic parser” and “semantic interpreter.” EX1006, 11:14-16:11, Fig. 2. Accordingly, Hartono with Baker maintains the same set prior probabilities for 1[c] as it does for 1[d]. Lopresti ¶344.

Moreover, Hartono alone discloses this limitation as applied to “parsing” step 1[d]. As discussed in Section X.C.5, Hartono’s “natural language syntactic parser 102b” and “natural language semantic interpreter 102c” perform the “parsing” in limitation 1[d]. *See* Section X.C.5 (Ground 4, 1[d]); Lopresti ¶¶345-349. Hartono discloses that after the utterance has been syntactically parsed, its “semantic interpretation engine” applies “context sensitive methodologies” to invoke the

appropriate application. EX1006, 13:13-21; Lopresti ¶346. To do so, Hartono’s teaches using a “reverse grammar generation mechanism” that maintains “a list or vector for each word and corresponding probabilities for each word in the list.” EX1006, 14:10-20; Lopresti ¶346. For example, Hartono describes how the “associated vector or list” that includes the words “‘I,’ ‘eye,’ or ‘aye,’” have “corresponding probabilities of 80%, 15%, and 5%.” EX1006, 14:10-20. A POSITA would have understood Hartono’s probability list to be *maintaining a set of prior probabilities or fuzzy possibilities* as it applies to 1[d]. Lopresti ¶¶347-348. Hartono’s probability list is also *dynamic* as they “may be adjusted depending upon each user’s selection of words used or depending upon a subset or all of the users’ selection of words used.” EX1006, 14:17-20; Lopresti ¶349.

#### 4. 1[c]

Hartono with Baker discloses this limitation. Lopresti ¶¶350-357. Hartono recognizes that “[a]ny suitable off the shelf speech recognition systems may be used as the automatic speech recognition system 102a in the natural language interface system 100.” EX1006, 11:4-12. Baker is such a system. Lopresti ¶351.

As discussed in Section X.C.3 (Ground 4, 1[b]), Baker *uses information in one or more dictionary and phrase tables to recognize words and phrases contained in the received utterance.* Lopresti ¶¶352-356. Baker teaches its “word list generator 16 can select relevant passages from the reference source 40 for each word

in the choice phrase 20,” and that it uses those relevant passages “to determine the likelihood that the choice phrase represents the utterance being analyzed.” EX1008, 21:19-47, 21:49-22:6. Specifically, Baker’s generated “word list” includes words that “have a semantic association with the choice word 20.” EX1008, 14:57-69. The “semantic association” for each choice word (also referenced as “passage words,” item 24 in Fig. 2) in the “word list” is derived from a “reference database 40” which “can be electronic memory that stores passages of text in an addressable format which typically includes a relevance indicator, such as a dictionary head word, that provides entry point information for the word list generator 16.” EX1008, 9:18-24; Lopresti ¶¶352-353. Baker provides several examples of “reference database 40,” including “dictionary, thesaurus, antonym dictionary, medical dictionary, legal dictionary, scientific and engineering dictionary, or other type of reference source.” EX1008, 20:55-58, 15:50-8, 15:16-31, 21:6-12. Baker further discloses that the “reference source 40” can include “phrase tables,” explaining that “reference source 40 can be a conventional software dictionary reference that includes discrete passages of text....” EX1008, 15:16-20, 15:32-37 (using passages from “Grolier’s Encyclopedia”), 16:8-12 (Encyclopedia Britannica); Lopresti ¶¶354-357.

## 5. 1[d]

Hartono discloses this limitation. Lopresti ¶¶358-363. Hartono’s “natural language syntactic parser 102b” and “natural language semantic interpreter 102c”

perform the “parsing” step 1[d]. Hartono discloses first syntactically parsing the natural language input into a structure using a Government and Binding approach (a computation method based on linguistic theory). EX1006, 12:1-20. This parsing “allow[s] the computer to map the verb into a computer command action, with the noun phrase (NP) as the object and the adjective phrase (AP) as properties of the object. EX1006, 13:8-10; Lopresti ¶359.

Hartono’s semantic interpreter then applies “context sensitive methodologies” to the syntactic parse to *determine a content for the at least one request contained in the utterance*. EX1006, 13:13-21; Lopresti ¶360. Specifically, Hartono “takes the syntactic parse of the spoken language request and maps it to a generic concept frame,” as in Table 1:

| TABLE I        |               |                          |
|----------------|---------------|--------------------------|
| ACTION-CONCEPT | TOPIC CONCEPT | APPLICATION-CONTEXT      |
| Show           | Email         | Email application        |
| Show           | Email address | Address book application |
| Delete         | Email         | Email application        |
| Show           | General help  | Natural language agent   |

EX100613:18-21, Table I, Table III. The semantic interpreter uses *keywords contained in the recognized words or phrases* to determine a context, by using verbs in the recognized words or phrases as “action-concepts” and nouns that “describ[e] the objects on which the actions are performed” as “topic-concepts.” EX1006, 14:1-7; 15:13-19 (Hartono “uses the combination of the action-concept and the topic-

concepts” to “determine which task agent should handle the request.”). The determined context is shown in Table I as the “Application-Context.” EX1006, Table I, 6:2-5; Lopresti ¶¶360-363.

6. 1[e]

Hartono discloses this limitation. Lopresti ¶¶364-370; *see also* Section VII.C.6 (Ground 1, 1[e]) (discussing *domain agents* in '209).

Like the '209, Hartono “utilizes an agent-based architecture” that uses “application class task agents for each class of applications,” as in Figure 1:

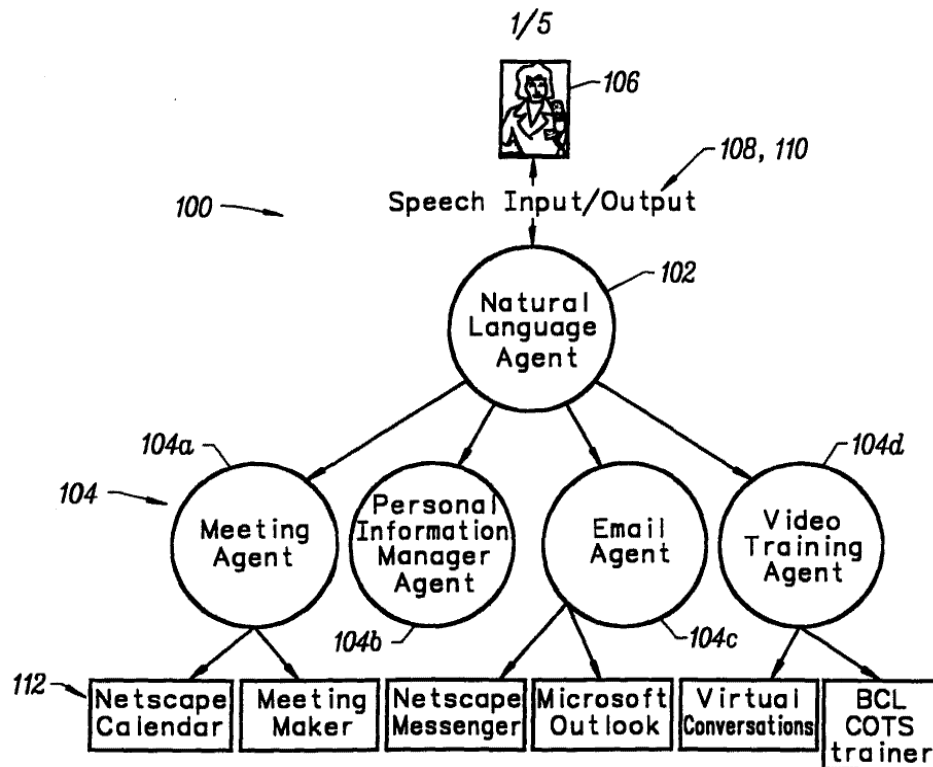


FIG. 1

EX1006, Fig 1, 8:10-13, 9:9-14. A POSITA would have understood that Hartono’s “task agents” (also called “application class agents”) are *domain agents* because they

are software modules specific to an application or application area. EX1006, 23:19-20, 28:7-29 (Table IV listing “task agents”); Lopresti ¶¶366-367. Hartono’s task agents are *autonomous executables* as each agent “includes a process-specific execution module for each application.” EX1006, 30:15-20; Lopresti ¶368.

Hartono further teaches using a “task interpretation engine” to *select at least one domain agent based on the determined meaning*, determined by the semantic interpreter. EX1006, 24:21-25:3, 15:13-15; Lopresti ¶¶369-370.

## 7. 1[f]

Hartono discloses this limitation. Lopresti ¶¶371-375. Hartono discloses that each task agent includes a “wrapper [that] is preferably written in one of the platform specific macro languages,” and is “configured to communicate with a corresponding application program.” EX1006, 24:14-16, Table II (showing platform specific macro languages); *see* EX1006, 5:4-8. The application wrapper “translates the conceptual action to a set of application specific operations.” EX1006, 24:4-13, 6:6-7, 24:21-25:3; Lopresti ¶¶372-373. A POSITA would have understood Hartono’s task agent that translates the parsed utterance into a set of application specific operations for execution by the application therefore teaches *formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context*. Lopresti ¶¶374-375.

**8. 1[g]**

Hartono discloses this limitation. Lopresti ¶¶376-378. Hartono discloses its task agents are *invoked* to output an executable instruction to an application program (*process the formulated request*). EX1006, Abstract (“[A]t least one application agent [is] adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program.”), 4:20-25, 5:4-15, 5:21-25, 36:7-9; Lopresti ¶¶377-378.

**9. 1[h]**

Hartono discloses this limitation. Lopresti ¶¶379-383. Hartono teaches that it “outputs results and responses” to natural language commands that “allow[] users to control multiple desktop applications by abstract commands.” EX1006, 6:1-9. Hartono discloses several different applications for executing the natural language input *generated in response to the utterance via the speech unit*, for example, electronic mail, schedulers, word processors, spreadsheets, and personal information managers among many others that present the results of the processed request to the user. EX1006, 44:1-13. Moreover, it would be obvious to a POSITA that once an application receives a command via an agent, it would process the command (*formulated request*) and present the results to the user. EX1006, 5:23-24, 9:13-25; Lopresti ¶¶380-383.

**D. Dependent Claims**

**1. Claim 4**

To the extent limiting, Hartono with Baker discloses this limitation for the same reasons as claim 1, and below for 4[a]-4[c]. Lopresti ¶¶384.

**a. 4[a]**

Hartono discloses this limitation. Lopresti ¶¶385-388. As discussed for 1[b] and 1[d], Hartono *maintains a dynamic set of prior probabilities or fuzzy possibilities* that is usable *to determine the context* for the at least one request from the recognized words and phrases. Sections VII.C.3, VII.C.5; Lopresti ¶386.

Hartono further explains that “[u]pon receiving the syntactic parse of the spoken language request, the semantic interpretation engine 102c determines the permutations of the syntactic parse using the list for each word.” EX1006, 14:21-15:1. By determining the different permutations, Hartono *matches the keywords contained in the recognized words or phrases to the dynamically updated set of keywords*. Lopresti ¶¶387-388.

**b. 4[b]**

Hartono discloses this limitation. Lopresti ¶¶389-407.

Using the permutations from the matched keywords, Hartono “determines which words best fit the grammar of the syntactic parse,” by determining if a word permutation “is suitable given the context.” EX1006, 14:21-15:1. If the word does not match a context (i.e., “is not suitable given the context”), “the next word having

each next highest probability is then evaluated and determined if that word is suitable given the context.” EX1006, 14:21-15:1; Lopresti ¶¶390-394.

Hartono’s semantic interpreter uses a “reverse grammar generation mechanism” that maintains “a list or vector for each word and corresponding probabilities for each word in the list.” EX1006, 14:10-20. For example, “associated vector or list” that includes the words “‘I,’ ‘eye,’ or ‘aye,’” have “corresponding probabilities of 80%, 15%, and 5%”), resulting in different permutations of the natural language utterance, scored from highest to lowest probability. EX1006, 14:10-20. Using these different probability-based permutations, Hartono “determines which words best fit the grammar of the syntactic parse,” which are *determined scores for possible contexts*. EX1006, 15:1-12. A POSITA would have understood that to determine “best fit,” a value or score would need to be assigned to the evaluated grammar of the syntactic parse to determine whether the word is “suitable given the context.” Specifically, Hartono will iterate through the recognized words received from Baker depending on the probabilities and ranks provided. *See* EX1006, 15:8-12 (“If the word is not suitable given the context, the next word having each next highest probability is then evaluated and determined if that word is suitable given the context, until a suitable word is determined.”); Lopresti ¶¶395-404.

Recognizing that “[r]elevance feedback is widely used for preference optimization,” Hartono discloses using an “adaptive preference manager 102e...associated with the natural language each agent 102 and with each user 106” that “learn[s] what default conditions are preferred by the user 106 by either monitoring the users[sic] action implicitly (i.e. observing in the background) and/or by being instructed explicitly by the user 106 on positive and/or negative preferences.” EX1006, 17:23-18:5, 6:19-20. Because Hartono is constantly monitoring a user’s actions, a POSITA would have found it obvious to include information about the user collected by the adaptive preference manager (*one or more of a profile of the user*) to aid in the suitability determination. Lopresti ¶¶405-407.

**c. 4[c]**

Hartono discloses this limitation. Lopresti ¶¶408-411. By evaluating whether a word is sufficiently suitable for a context, Hartono finds that the word satisfies a predetermined confidence level and thereby determines (and selects) the context. *See* EX1006, 15:8-12; Lopresti ¶409. Once the semantic interpreter has scored the words in the recognized natural language utterance, it then *determines the context by selecting one or more of the possible contexts* that have been determined to be suitable (*having a score satisfying a predetermined confidence level*) by “take[ing]

the syntactic parse of the spoken language request and map[ing] it to a generic concept frame,” as in Table 1:

| TABLE I        |               |                          |
|----------------|---------------|--------------------------|
| ACTION-CONCEPT | TOPIC CONCEPT | APPLICATION-CONTEXT      |
| Show           | Email         | Email application        |
| Show           | Email address | Address book application |
| Delete         | Email         | Email application        |
| Show           | General help  | Natural language agent   |

EX1006, 13:18-21, Table I, Table III. Lopresti ¶410. Specifically, the semantic interpreter uses verbs in the recognized words or phrases as “action-concepts” and nouns that “describ[e] the objects on which the actions are performed” as “topic-concepts.” EX1006, 14:1-7. Hartono “uses the combination of the action-concept and the topic-concepts” to “determine which task agent should handle the request.” EX1006, 15:13-19; *id.*, 6:2-5; Lopresti ¶411.

## 2. Claim 6

To the extent limiting, Hartono with Baker discloses this limitation for the same reasons as claim 1, and below for 6[a]-6[d]. Lopresti ¶412.

### a. 6[a]

Hartono discloses this limitation. Lopresti ¶¶413-419. As discussed for 1[e], Hartono discloses a task agent (*domain agent*). See Section X.C.6 (Ground 4, 1[e]); Lopresti ¶¶414-416. Hartono further discloses that after parsing natural language commands into a computer action (with noun phrases (NP) as the object and the

adjective phrase (AP) as properties of the object), the spoken language request is then mapped “into a generic concept frame used to invoke the appropriate application method.” EX1006, 12:11-10, 13:12-20; Lopresti ¶417. Hartono provides an example of the command “show me the first message,” which is then parsed to include values for the verb (“show”), and nouns (“me” and “message”) and adjective (“first”). EX1006, 12:21-13:10. Hartono shows how its commands include optional values in Table V—for example, “I want to check out my mail” and “I want to know if I got new mail,” would result in the same semantic action such that whether the mail is “new” is an optional value. EX1006, 30:21-31:5; Lopresti ¶418. A POSITA would have thus understood the identification of these noun and adjective phrases to be *determining one or more required values and one or more optional values associated with formulating request in the grammar used by the domain agent*. Lopresti ¶419.

**b. 6[b]**

Hartono discloses this limitation. Lopresti ¶¶420-425; *see also* Section VIII.C.2 (Ground 2, 6[b]) (discussing “criteria” and “parameter” in ’209).

Hartono’s semantic interpreter applies “context sensitive methodologies” to the syntactic parse to *extract one or more criteria and one or more parameters from one or more keywords contained in the one or more recognized words or phrases*. EX1006, 13:13-21; Lopresti ¶¶421-423. Specifically, Hartono “takes the syntactic

parse of the spoken language request and maps it to a generic concept frame,” thus extracting the various criteria and parameters of the request, as in Table 1:

| TABLE I        |               |                          |
|----------------|---------------|--------------------------|
| ACTION-CONCEPT | TOPIC CONCEPT | APPLICATION-CONTEXT      |
| Show           | Email         | Email application        |
| Show           | Email address | Address book application |
| Delete         | Email         | Email application        |
| Show           | General help  | Natural language agent   |

EX1006, 13:18-21, Table I, Table III. The semantic interpreter uses *keywords contained in the recognized words or phrases*, by using verbs in recognized words or phrases as “action-concepts” and nouns that “describ[e] the objects on which the actions are performed” as “topic-concepts.” EX1006,14:1-7; 15:13-19. The determined context is shown in Table I as the “Application-Context.” EX1006, Table I, 6:2-5; Lopresti ¶¶424-425.

**c. 6[c]**

Hartono discloses this limitation. Lopresti ¶¶426-430. As part of its “semantic interpretation engine,” Hartono discloses using a “reverse grammar generation mechanism” that maintains “a list or vector for each word and corresponding probabilities for each word in the list.” EX1006, 14:10-20. For example, Hartono describes how the “associated vector or list” that includes “‘I,’ ‘eye,’ or ‘aye,’” which have “corresponding probabilities of 80%, 15%, and 5%,” respectively. EX1006, 14:10-20. A POSITA would have understood Hartono’s

probability list to be a *set of prior probabilities or fuzzy possibilities*. Lopresti ¶¶427-429. Hartono's probability list is also *dynamic* as they "may be adjusted depending upon each user's selection of words used or depending upon a subset or all of the users' selection of words used." EX1006, 14:17-20; Lopresti ¶428.

Hartono discloses using the "reverse grammar generation mechanism" to *infer one or more further criteria and one or more further parameters associated with the request*. Lopresti ¶¶429-430. Hartono's reverse generation grammar mechanism uses various permutations of the syntactic parse to "determine[] which words best fit the grammar of the syntactic parse." EX1006, 15:4-4. Using the word with the highest probability first, Hartono "evaluate[s] and determine[s] if that word is suitable given the context" and "if no suitable word is determined, then the natural language agent may request clarification or correction from the user." EX1006, 15:5-12. By using the *prior probabilities* to determine the best fitting word, a POSITA would have understood that Hartono infers one or more further criteria and one or more further parameters associated with the request, and if they are missing would seek clarification or correction from the user. Lopresti ¶¶430.

**d. 6[d]**

Hartono discloses this limitation. Lopresti ¶431. Hartono teaches that each task agent includes a "wrapper [that] is preferably written in one of the platform specific macro languages," and "configured to communicate with a corresponding

application program.” EX1006, 24:14-16, Table II; *see* EX1006, 5:4-8. The application wrapper “translates the conceptual action to a set of application specific operations.” EX1006, 24:4-13, 6:6-7, 24:21-25:3; Lopresti ¶¶432-433. Hartono’s “application specific operations” to be executed by the application would have been understood by a POSITA to be *one or more tokens*, because it comprises individuated or “non-reducible text element in data.” EX1024 (defining “token”); Lopresti ¶433. A POSITA would understand Hartono’s task agent, using the task application wrapper, translates the parsed utterance into a set of application specific operations for execution by the application therefore discloses *transforming the extracted and the inferred criteria and parameters into a format compatible with the grammar used by the selected domain agent, the transformed criteria and parameters including all of the required values and one or more of the optional values*. Lopresti ¶¶432-433.

### **3. Claim 7**

To the extent limiting, Hartono with Baker discloses this limitation for the same reasons as claim 1. Lopresti ¶435. Hartono also provides examples of a request contained *in the utterance* that includes *at least one question and at least one command*. Hartono describes the request “can you create a letter for my friend, John Smith?,” which includes both a question (can you create?) and a command (create a letter to John Smith). EX1006, 29:2-6; Lopresti ¶435. As below for 7[a]-7[b],

Hartono discloses that its invoked task agent (*domain agent*) processes the formulated request.

**a. 7[a]**

Hartono discloses this limitation. Lopresti ¶¶436-438. Hartono discloses that its semantic mapper “maps the user the input to a specific action within the small set of semantic actions 142,” by “draft[ing] a letter to John Smith and the same task is performed.” EX1006, 29:2-30:6, 30:12-14. A POSITA would have understood that Hartono’s task agent (*domain agent*) must be *directed to a local or remote processing device* to perform the command on a computer. EX1006, 33:21-35:3 (describing Hartono’s implementation on computer including multi-processor system); Lopresti ¶¶437-438.

**b. 7[b]**

Hartono discloses this limitation. Lopresti ¶¶439-440. Hartono discloses that its request application class agents, or task agents, depending on the requested question *will query local or network information sources to resolve information requested in the at least one question*. EX1006, Table IV, 39:1-13. For example, Table IV shows how certain task agents (*domain agents*) will locate information requested in a question over the Internet or an address book. EX1006, Table IV; Lopresti ¶440.

## **XI. GROUND 5:**

### **A. Motivation to Combine**

A POSITA would have been motivated to implement Aridor’s metasearch functionality as a Hartono task agent (*domain agent*) to collect information from the Internet and filter and rank the results based on relevance. Lopresti ¶¶441-445. Hartono describes its domain agents that search the Internet but does not detail how. EX1006, Table IV, 39:1-13. Accordingly, a POSITA would have been motivated to look for and rely on known Internet searching mechanisms in retrieving and processing responses to user queries (like Aridor). Lopresti ¶¶441-442.

A POSITA would have understood that Aridor’s metasearch would be compatible with Hartono, as Aridor uses domain-based knowledge agents. Lopresti ¶443. Combining Aridor with Hartono would also improve Aridor’s methods by obviating the need for users to “choose the knowledge agent that has the relevant specialization” to perform the search. EX1010, 3:53-59; Lopresti ¶443. A POSITA would have further been motivated by Aridor’s “ranking and filtering” features that process the results of the user’s query to find the most relevant information. Lopresti ¶444. A POSITA would have been also motivated by Aridor’s knowledge agents to “make use of persistent, acquired knowledge in the domain that is defined and then refined by a user,” based on the user’s queries made over time. EX1010, 3:65-4:2, 14:56-15:19; Lopresti ¶445.

**B. Reasonable Expectation of Success**

Aridor merely extends Hartono's natural language processing of Internet queries to metasearch functionality to obtain relevant results. Lopresti ¶¶446-448. A POSITA would have the skill to implement and the reasonable expectation of success in achieving such a combination because it would have involved a combination of known technologies (speech recognition and Internet searching) according to known methods (speech recognition engines that convert speech into text strings and using text strings to perform Internet searching) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke to retrieve information from the Internet. Lopresti ¶¶447-448.

**1. Claim 8**

To the extent limiting, Hartono and Baker with Aridor discloses this limitation for the same reasons as claims 1 and 7, and below for 8[a]-8[c]. Lopresti ¶449.

**a. 8[a]**

Hartono with Aridor discloses this limitation. Lopresti ¶¶450-453. As discussed in Ground 3, Aridor discloses knowledge agents that submit duplicate search queries to multiple search engines. *See* Section IX.D.1 (Ground 3, 8[a]); Lopresti ¶¶452-453. A POSITA would have been motivated to implement Aridor's

knowledge agent as Hartono's task agent to improve its Internet search functionality.  
*See also* Section XI.A.

**b. 8[b]**

Hartono with Aridor discloses this limitation. Lopresti ¶¶454-456. As discussed in Ground 3, Aridor discloses receiving results from the different search engines *asynchronously*. *See* Section IX.D.2 (Ground 3, 8[b]); Lopresti ¶¶455-456. A POSITA would have been motivated to implement Aridor's asynchronous results because it would have allowed for faster processing. *See also* Section XI.A.

**c. 8[c]**

Hartono with Aridor discloses this limitation. Lopresti ¶¶457-465. As discussed in Ground 3, Aridor discusses ranking and filtering the results based on history associated with prior sites that impact the probabilities maintained by Hartono. *See* Section IX.D.3 (Ground 3, 8[c]); Lopresti ¶¶458-465. A POSITA would have been motivated to implement Aridor's ranking and filtering to provide better results. *See also* Section XI.A.

**2. Claim 9**

To the extent limiting, Hartono and Baker with Aridor discloses this limitation for the same reasons as claims 1, 7-8, and below for 9[a]-9[d]. Lopresti ¶466.

**a. 9[a]**

Hartono with Aridor discloses this limitation. Lopresti ¶¶466-469. As discussed for Ground 3, Aridor discloses creating a query profile that includes

specific instances of a word or phrase that appears in the spoken utterance for which values are sought. *See* Section IX.E.1 (Ground 3, 9[a]); Lopresti ¶¶467-469. A POSITA would have been motivated to use Aridor’s query profile to provide more robust results. *See also* Section XI.A.

**b. 9[b]**

Hartono with Aridor discloses this limitation. Lopresti ¶¶470-474. As discussed for Ground 3, Aridor discloses extracting scores by scraping the results for both textual similarity and topological aspects. *See* Section IX.E.2 (Ground 3, 9[b]). Lopresti ¶¶471-474. A POSITA would have been motivated to scrape results as taught by Aridor to provide more robust results. *See also* Section XI.A.

**c. 9[c]**

Hartono with Aridor discloses this limitation. Lopresti ¶¶475-478. Aridor’s “textual score” and “link topology score” both use *a dynamic set of prior probabilities or fuzzy possibilities* using, in part, “domain textual scores” and “anchor links,” which are computed using Aridor’s dynamic knowledge base. *See* Section IX.E.3 (Ground 3, 9[c]); Lopresti ¶¶476-478. A POSITA would have been motivated to use Aridor’s evaluation to provide more robust results. *See also* Section XI.A.

**d. 9[d]**

Hartono with Aridor discloses this limitation. Lopresti ¶¶479-481. As discussed in Ground 3, Aridor evaluates the extracted criteria using a ranking algorithm. *See* Section IX.E.4 (Ground 3, 9[d]); Lopresti ¶¶480-481. A POSITA would have been motivated to use Aridor’s evaluation to provide more robust results. *See also* Section XI.A.

**XII. MANDATORY NOTICES**

**A. Real Parties-in-Interest**

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

**B. Related Matters**

U.S. Patent Office records indicate that the ‘209 patent is assigned to Dialect, LLC (“PO”), which is currently asserting the ‘209 patent in the following concurrent litigation filed on February 7, 2025: *Dialect, LLC v. Meta Platforms, Inc.*, 7:25-cv-00060 (W.D. Tex.).

There are five patents asserted in the concurrent litigation. Petitioner has filed petitions for *inter partes* review against U.S. Patent No. 8,447,607 (IPR2025-01332) and U.S. Patent No. 9,263,039 (IPR2025-01333) and related family members U.S. Patent No. 8,015,006 (IPR2025-01335) and U.S. Patent No. 9,734,825 (IPR2025-01336).

The '209 patent is also asserted by PO in the following pending case: *Dialect LLC v. Salesforce, Inc.*, Case No. 7:25-cv-00061 (W.D. Tex.), *Dialect LLC v. Microsoft Corp.*, Case No. 2:24-cv-01067 (E.D. Tex.), *Dialect LLC v. Google LLC*, Case No. 3:24-cv-04388 (N.D. Cal.).

**C. Notice of Counsel and Service Information**

Meta Platforms, Inc. provides the following designation of counsel:

|                 |                                                                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Lead Counsel    | Lisa K. Nguyen (Reg. No. 58,018)<br>Paul Hastings LLP<br>1117 S. California Avenue<br>Palo Alto, CA 94303<br>Telephone: 650.320.1800<br>Fax: 650.320.1900<br>Email: PH-META-DialectIPR@paulhastings.com         |
| Back-Up Counsel | David Tennant (Reg. No. 48,362)<br>Paul Hastings LLP<br>2050 M St., N.W.<br>Washington, DC, 20036<br>Telephone: 202.551.1700<br>Fax: 202.551.1705<br>Email: PH-META-DialectIPR@paulhastings.com                 |
| Back-Up Counsel | Diane Ghrist ( <i>pro hac vice</i> to be filed)<br>Paul Hastings LLP<br>2050 M St., N.W.<br>Washington, DC, 20036<br>Telephone: 202.551.1700<br>Fax: 202.551.1705<br>Email: PH-META-DialectIPR@paulhastings.com |
| Back-Up Counsel | Rachel Wu Hankinson ( <i>pro hac vice</i> to be filed)<br>525 South Flower Street<br>Twenty-Fifth Floor                                                                                                         |

|  |                                                                                                                     |
|--|---------------------------------------------------------------------------------------------------------------------|
|  | Los Angeles, CA 90071<br>Telephone: 213.683.6112<br>Fax: 213.627.0705<br>Email: PH-META-DialectIPR@paulhastings.com |
|--|---------------------------------------------------------------------------------------------------------------------|

Pursuant to 37 C.F.R. § 42.10(b), a Power of Attorney accompanies this petition. Petitioner consents to electronic service by e-mail.

**D. Fee for *Inter Partes* Review**

The Director is authorized to charge the fee specified by 37 C.F.R. § 42.15(a) to Deposit Account 50-2613.

**E. Grounds for Standing**

Petitioner certifies pursuant to 37 C.F.R. § 42.104(a) that the ‘209 patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

**XIII. CONCLUSION**

For these reasons, Petitioner respectfully requests institution.

Dated : August 19, 2025

Respectfully submitted,

By: / 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](mailto:lisanguyen@paulhastings.com)

U.S. Patent No. 7,398,209  
Petition for *Inter Partes* Review

*Counsel for Petitioner*  
**META PLATFORMS, INC.**

**CLAIM LISTING APPENDIX**

The following claim listing adds letters ([a],[b], etc.) to the independent claims for clarity:

1. Preamble: “A method responsive to a user generated natural language speech utterance, comprising:”

[a] “receiving the user generated natural language speech utterance, the received user utterance containing at least one request”

[b] “maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance”

[c] “recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables”

[d] “parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases”

[e] “selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context”

[f] “formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context”

[g] “invoking the selected domain agent to process the formulated request; and”

[h] “presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.”

4. The method according to claim 1, wherein determining the context for the at least one request contained in the utterance includes:

[a] “matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords, the dynamically updated set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities”

[b] “scoring one or more possible contexts for the matched keywords using one or more of a profile of the user, data content associated with the domain agents, or a context stack of one or more recent contexts; and

[c] “determining the context by selecting one or more of the possible contexts having a score satisfying a predetermined confidence level.”

6. The method according to claim 1, wherein formulating the at least one request in accordance with the grammar used by the selected domain agent includes:

[a] determining one or more required values and one or more optional values for a request in the grammar used by the selected domain agent;

[b] extracting one or more criteria and one or more parameters from the keywords contained in the recognized words and phrases using procedures sensitive to the determined context;

[c] inferring one or more criteria and one or more parameters using the dynamic set of prior probabilities or fuzzy possibilities; and

[d] transforming the extracted and the inferred criteria and parameters into a format compatible with the grammar used by the selected domain agent, the transformed criteria and parameters including all of the required values and one or more of the optional values.

7. The method according to claim 1, wherein the at least one request contained in the utterance includes at least one question and at least one command, wherein the invoked domain agent processes the formulated request by:

[a] directing the at least one command to local or remote processing devices; and

[b] querying local or network information sources to resolve information requested in the at least one question.

8. The method according to claim 7, wherein querying the local or network information sources to resolve the information requested in the at least one question includes:

[a] submitting a plurality of asynchronous queries to the information sources, the plurality of queries including a plurality of duplicate queries submitted to distinct information sources;

[b] a synchronously receiving a plurality of results of the queries from the information sources; and

[c] scoring a relevance of each of the asynchronously received results using the dynamic set of prior probabilities or fuzzy possibilities to determine one or more best responses to the at least one question.

9. The method according to claim 8, wherein determining the one or more best responses to the at least one question includes:

[a] determining one or more tokens required to formulate a response to the question;

[b] extracting one or more values for the determined tokens from the asynchronously received results, the one or more values extracted by applying one or more scraping criteria to the asynchronously received results;

[c] evaluating the extracted values using the dynamic set of prior probabilities or fuzzy possibilities, wherein evaluating the extracted values includes resolving ambiguous, incomplete, or conflicting information; and

[d] selecting one or more best values for the determined tokens from the evaluated values, wherein the one or more best responses include the one or more selected best values.

**CERTIFICATE OF COMPLIANCE**

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that foregoing  
**PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 7,398,209**  
exclusive of the parts exempted as provided in 37 C.F.R. §42.24(a), contains 13,958  
words and therefore complies with the type-volume limitations of 37 C.F.R.  
§42.24(a).

Dated : August 19, 2025

Respectfully submitted,

By: / 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**

I hereby certify that a true and correct copy of the foregoing **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 7,398,209 AND EXHIBITS EX1001-1026** are being served on August 19, 2025, via express overnight mail on counsel of record for U.S. Patent No. 7,398,209 as addressed below:

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

A courtesy copy is also being served on counsel for the patent holder in the pending litigation:

Garland Stephens garland@bluepeak.law)  
Richard Koehl (richard@bluepeak.law)  
**Blue Peak Law Group LLP**  
3139 West Holcombe Blvd. PMB 8160  
Houston, TX 77025

William Ellerman(ellerman@cjsjlaw.com)  
Cherry Johnson Siegmund James, PLLC  
One Glen Lakes Tower  
8140 Walnut Hill Lane, Suite 105  
Dallas, TX 75231

Mark D. Siegmund  
(msiegmund@cjsjlaw.com)  
Shuya Yang  
(gyang@cjsjlaw.com)  
**Cherry Johnson Siegmund James,  
PLLC**

7901 Fish Pond Road, 2nd Fl.  
Waco, TX 76710

Dated : August 19, 2025

Respectfully submitted,

By: / 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](mailto:lisanguyen@paulhastings.com)

U.S. Patent No. 7,398,209  
Petition for *Inter Partes* Review

*Counsel for Petitioner*  
**META PLATFORMS, INC.**