Filed: August 15, 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. IPR2025-01335
U.S. Patent No. 8,015,006

**Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006**

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes

Other Authorities

TABLE OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 8,015,006 to Kennewick, et al. ("the '006 Patent")
1002	File History of U.S. Patent No. 8,015,006
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	Reserved
1014	Appendix B – Infringement Contentions for U.S. Patent No. 8,015,006, *Dialect LLC v. Meta Platforms, Inc.*, No. 7:25-CV-00060-DC-DTG (E.D. Tex. July 16, 2025)
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, *Statistical Methods for Speech Recognition*, The MIT Press (1997)
1020	Michael McTear, *Spoken Dialogue Technology: Enabling the Conversational User Interface*, 34 ACM Computing Surveys No. 1, pp. 90-169 (Mar. 2002)
1021	Liam Comerford, et al., *The IBM Personal Speech Assistant*, IBM Watson Research Center, IEEE (2001)
1022	Eric J. Glover, et al., *Architecture of a Metasearch Engine that Supports User Information Needs*, 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 claims 5 and 6 of U.S. Patent No. 8,015,006 (EX1001, "'006 patent").

Natural language speech recognition methods have been used for decades. *See generally*, Lopresti ¶¶43-49. By the mid-1990s, companies including IBM and Dragon Systems, were selling speech recognition systems off-the-shelf to process natural language speech into machine-readable text to be used by various autonomous applications. They used statistical probabilities, language modeling, and context-recognition to recognize and parse speech. Because these systems employed separate, autonomous applications, those applications required formatting queries in an acceptable grammar for the application.

The '006 patent describes speech recognition methods that accept natural language speech queries or commands for use among various application programs. Although the original claims of the '006 patent were rejected, the Examiner allowed claims after the applicant heavily amended them to require formulating a natural language request into a grammar of the external application and scraping criteria to extract values from information sources. But by 2002, reformulating natural language requests into grammars that could be understood by external, speech-enabled applications was necessary to generate a response. Likewise, by 2002, using scraping criteria to extract information from source documents was commonplace.

Accordingly, Petitioner respectfully requests the Board institute review and find all challenged claims of the ’006 patent unpatentable.

II. IDENTIFICATION OF CHALLENGES

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

III. BACKGROUND

A. ’006 Patent

The ’006 patent describes well-known methods for users to submit natural language speech queries to use multiple applications. The ’006 patent recognizes that “[m]uch work covering multiple methods has been done in the fields of natural language processing and speech recognition,” and “[s]peech recognition…today is successfully used in a wide range of applications.” EX1001, 1:45-48. The ’006 further recognizes that “[n]atural language processing has previously been applied to the parsing of speech queries.” *Id.*, 1:49-50.

The ’006 patent recites nothing more than a combination of known methods. At a high level, claims 5 and 6 of the ’006 patent recite the basic steps of receiving a natural language utterance, recognizing the “words or phrases contained in the utterance, parsing the utterance to determine a meaning and context of the utterance, processing the request and presenting the results. EX1001, Claims 5, 6; Lopresti ¶¶55-58. Claim 5 focuses on "formulating…the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context," which is used to "invoke the correct agent." EX1001, Claim 5, 18:12-35, 18:50-56. Claim 6 focuses on performing search queries and evaluating the relevancy of the returned responses. EX1001, Claim 6. But these are all logical, common-sense steps used to format a spoken query into a computer command that an external application can accept, process, and return results. Lopresti ¶¶55-59.

B. Prosecution History

The '006 patent issued from U.S. Patent Application No. 12/130,397. The original claims were subject to a single prior art rejection, primarily in view of U.S. Patent No. 6,757,718 to Halverson et al. EX1002, 141-147.

Following an examiner interview, the applicant cancelled or heavily amended the claims. Application claims 24 and 27 (issued claims 5 and 6, respectively) were amended to be independent claims, which included limitations relating to a "speech recognition engine," a "parser," and steps for "formulating, at the parser, the request in contained in the utterance in accordance with a grammar used by a domain agent," among others. EX1002, 172, 203-204. The applicant's cancellation of claims and amendments mooted the non-final rejection, resulting in allowance. EX1002, 201-

209. For claim 5, the Examiner recognized that while "[i]t is known individually to provide grammars with required and optional values, and to infer criteria from a current context…the prior art of record does not disclose or reasonably suggest at least a format transformation in combination." EX1002, 238. For claim 6, the Examiner stated the prior art did not teach "applying scraping criteria to extract the values" for responding to a user's query. EX1002, 238-39.

IV. THE PRIOR ART

A. Ross (EX1005)

Ross is a U.S. Patent assigned to IBM entitled "System and Method for Determining Utterance Context in Multi-Context Speech Application" filed on August 16, 2001 and issued on August 1, 2006. Ross is prior art at least under pre-AIA §102(a) and (e). EX1005, Cover. Ross is presented as a primary reference.

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; Lopresti ¶¶68-73. Upon receiving a speech input, Ross uses a 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, 6:62-7:2. Ross recognizes that by 2001, speech engines were already well known. EX1005, 1:17-22, 4:34-38.

Ross describes a "context manager" that receives the speech representation from such a "speech engine" and 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 list based on an "access characteristic," such as the "recency of relevant access to the context," as shown below in Figure 4:

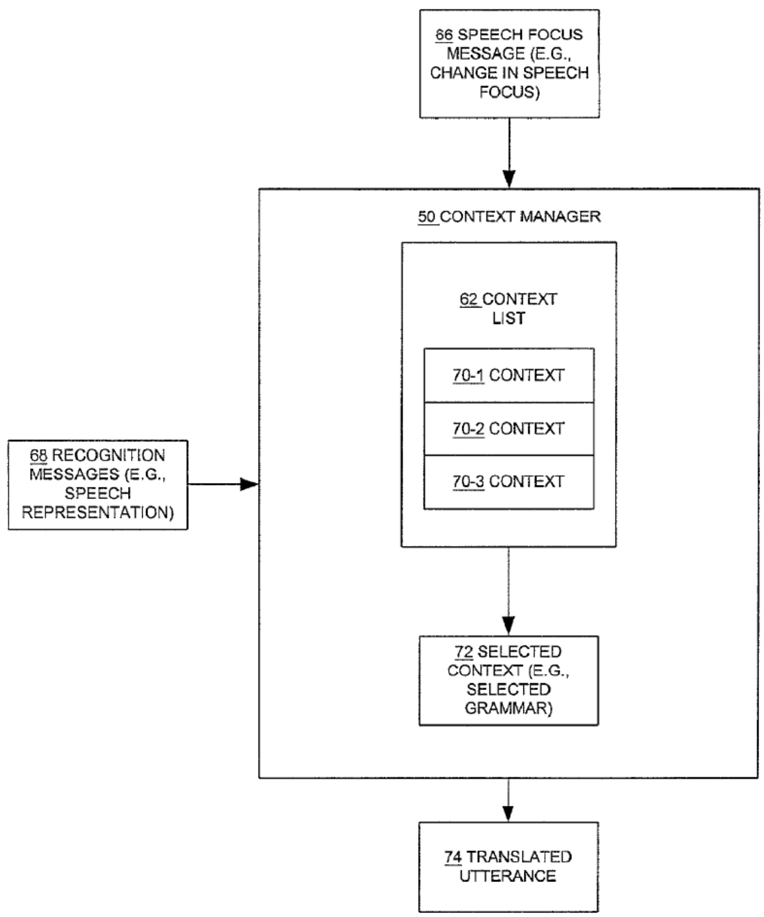


FIG. 4

EX1005, Fig. 4, 2:47-49, 3:1-3, 6:47-61, 8:46-9:13; Lopresti ¶¶68-73. 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 ¶¶70-71. 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 (i.e., scoring). EX1005, 7:23-26. Ross translates the speech representation "according to the selected context 72 and directs the translation to the script engine," the script engine then 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 ¶¶71-72.

B. Ittycheriah (EX1007)

Ittycheriah is a U.S. patent entitled "Apparatus and Methods for Speech Recognition Including Individual or Speaker Class Dependent Decoding History Caches for Fast Word Acceptance or Rejection" filed on June 4, 1997, and issued on August 10, 1999. EX1007, Cover. Ittycheriah is thus prior art under pre-AIA §102(a), (b), and (e).

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

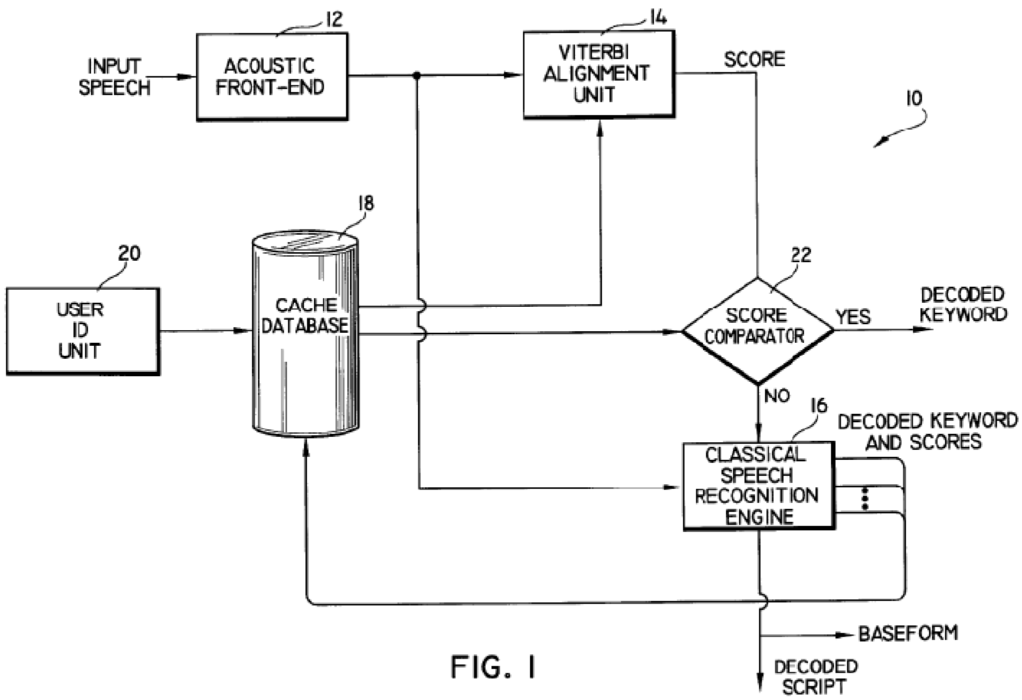


FIG. 1

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 ¶¶74-75. Ittycheriah uses Hidden Markov Models (HMMs) to perform speech recognition, explaining that Hidden Markov modeling "provides different scores for different hypothesis" using a set of prior probabilities for speech inputs. EX1007, 3:54-4:4; Lopresti ¶76. 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 the new scores are "not within a margin of error, or are less than any of the stored scores…the input speech segment is subjected to the complete vocabulary of the classical speech recognition engine," from which a decoded script (i.e., a keyword) is outputted. EX1007, 4:48-53; Lopresti ¶¶76-78.

Ittycheriah's vocabulary in its "classical speech engine" contains a "larger scale language database." EX1007, 3:62-64. The vocabulary 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 in the same way as the '006 patent. Lopresti ¶¶79-80.

A. Ross018 (EX1009)

Ross018 is a U.S. Patent entitled "System and Method for Relating Syntax and Semantics for a Conversational Speech Application," filed on October 25, 2001, and issued on July 24, 2007. EX1009, Cover. Ross018 is prior art at least under pre-AIA §102(a) and (e).

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 ¶¶81-84. 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 ¶¶83-84.

B. Hartono (EX1006)

Hartono is a PCT application entitled "Adaptive Natural Language Interface," with published on March 2, 2000. EX1006, Cover. Hartono is thus prior art under §102(a) and (b). Hartono is presented as a primary reference.

Hartono discloses a "method 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 from the natural language agent 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, below:

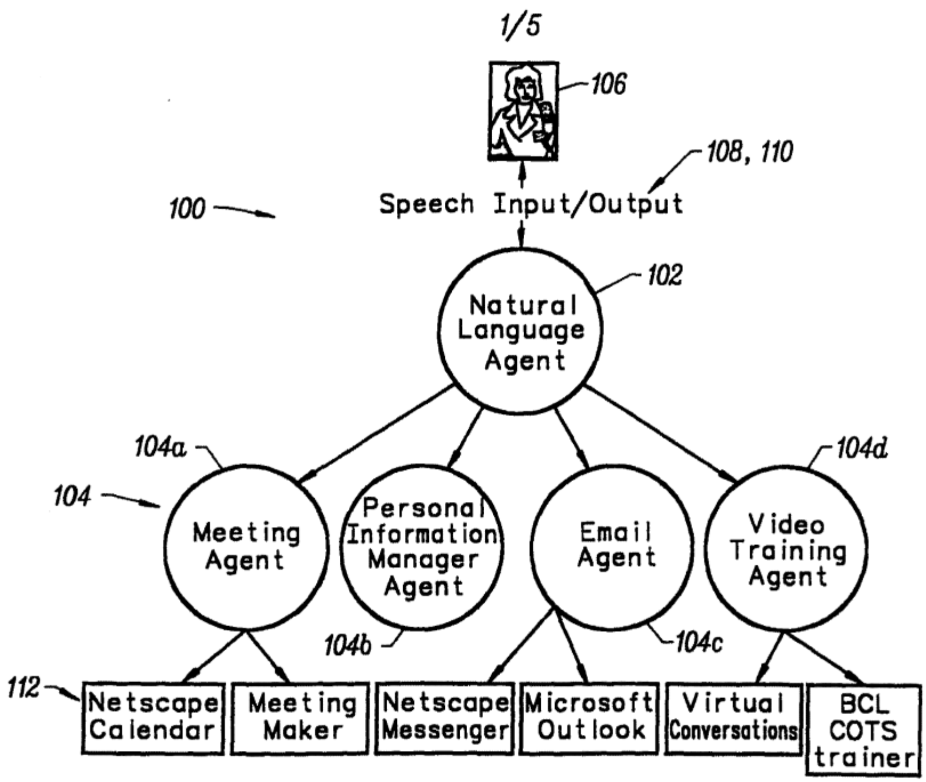


FIG. 1

EX1006, Fig. 1, 7:6-9; Lopresti ¶¶86-87. 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 ¶¶88-89. Hartono takes the parsed natural language and "maps it into a generic concept frame used to invoke the appropriate application method," as shown 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," having probabilities of 80%, 15%, and 5%, respectively such that the word with the highest probability 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 "determine[ing] which task agent should handle the request." EX1006, 15:13-19; Lopresti ¶¶89-93.

Hartono communicates with various "application class" or "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.

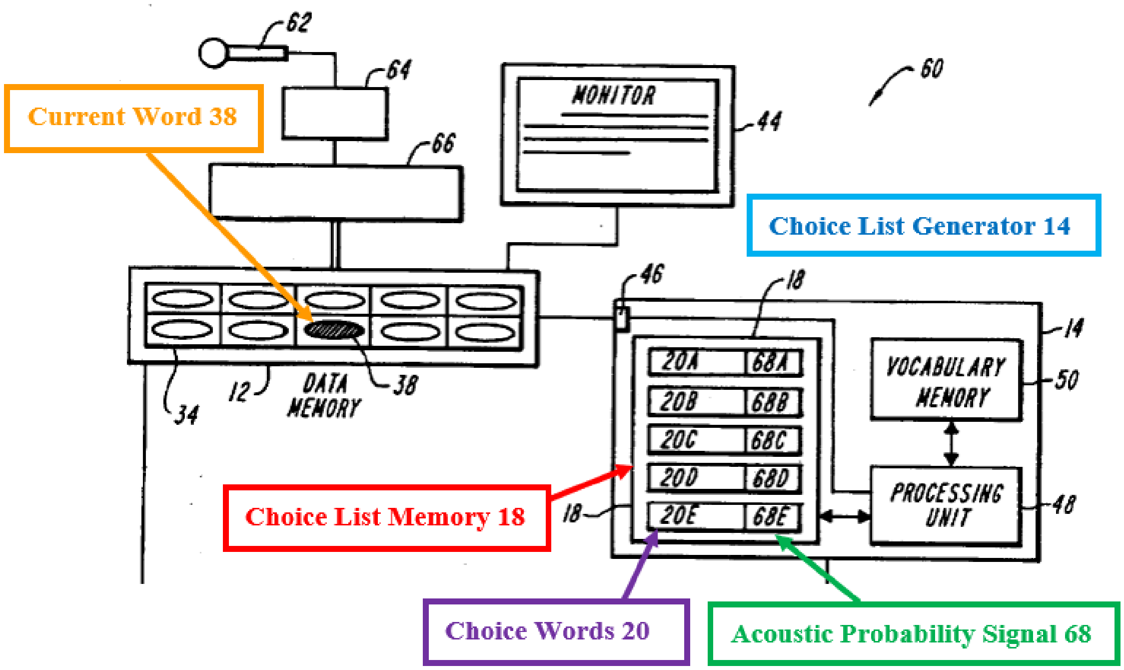
The selected application then executes the command within the natural language request. EX1006, 4:20-25, 5:4-8, 5:11-15, 29:21-22, 42:11-12; Lopresti ¶¶94-96.

C. Baker (EX1008)

Baker is a U.S. Patent entitled "Systems and Methods for Word Recognition" filed on June 7, 1995 and issued on October 21, 1997. EX1008, Cover. Baker is thus prior art under pre-AIA §102(a), (b), and (e).

Baker describes methods 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. Upon detection of 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 (the "current word"). EX1008, 18:22-24; Lopresti ¶¶97-101.

For unrecognized 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 illustrated below in the following excerpt of 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 ¶¶102-103.

Baker also discloses generating a "word list" of words that "have a 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 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.

Baker "perform[s] a matching function to calculate the number of matched words" between those words recognized by the speech analysis unit 66 with high confidence (the "context words") with the words in the "word list" (the "passage words"). EX1008, 21:19-47. By looking at the "relative frequency of matches between the "context words" and the "passage words," the 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 ¶¶104-107.

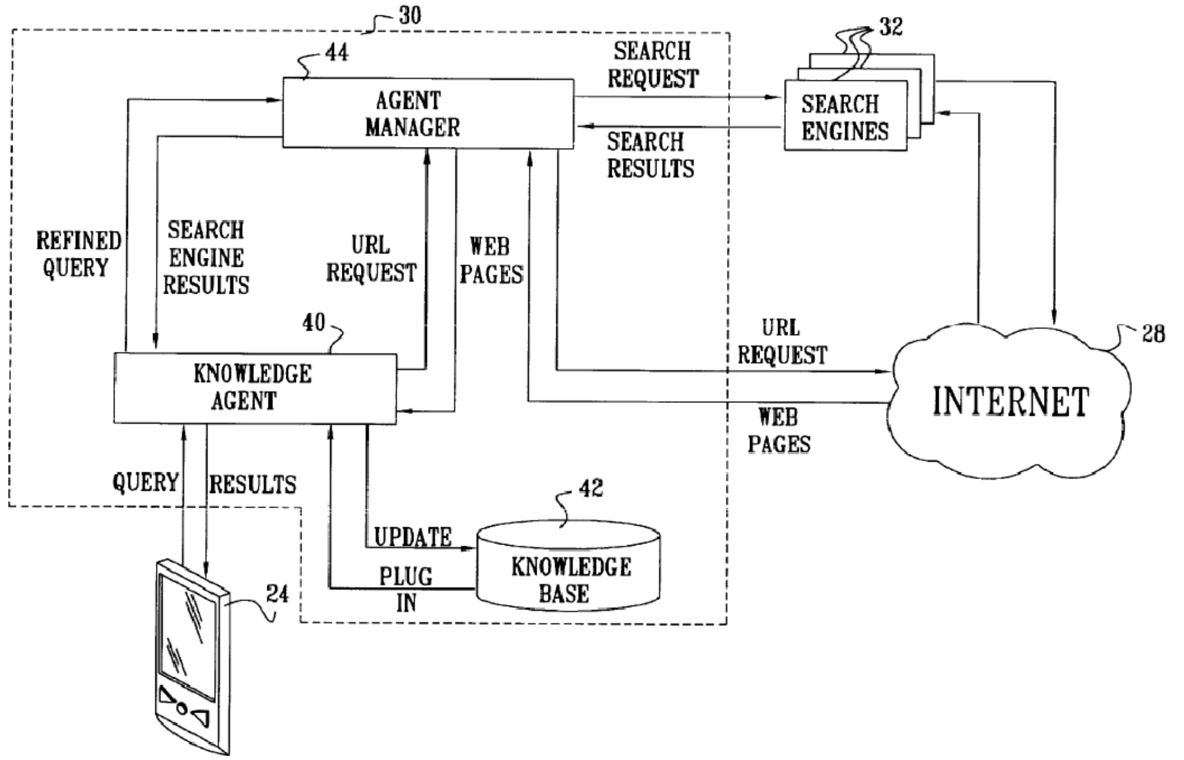
Baker generates a "rank signal 30," "represent[ing] the likelihood…that a choice word 20 is substantially similar to" the word being recognized as a function of the "context matching and acoustic model matching" probabilities. EX1008, 23:22-27. Baker selects the choice word with the highest rank signal 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.

D. Aridor (EX1010)

Aridor is a U.S. Patent entitled "Information Search Using Knowledge Agents" filed on June 6, 2000, and issued on October 21, 2003. EX1010, Cover. Aridor is thus prior art under pre-AIA §102(a) and (e).

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 ¶¶111-113. 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, such as AltaVista, Google or Hotbot,” 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 ¶¶111-113.

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 ¶¶62-64.

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 ¶¶65-67.

VI. CLAIM CONSTRUCTION

The Board construes claims "using the claim construction standard that would be used" in district courts. 37 C.F.R. § 42.100(b). This Petition establishes the prior art meets each claim limitation under any reasonable construction—no express construction necessary.[1]

---

[1] Petitioner reserves the right to argue alternative constructions in other proceedings.

VII. GROUND 1:

A. Motivation to Combine

1. Ross and Ittycheriah

A POSITA would have been motivated to use Ittycheriah's speech recognition method to implement the details of Ross's "speech engine." Lopresti ¶¶116-122. Ross uses a "speech engine" to "convert speech into text strings that can be utilized by software applications on a computer." EX1005, 1:7-22. Ross identifies a "speech engine," but recognizes that off-the-shelf speech engines could be used to convert speech to text. EX1005, 1:16-21, 4:37-38 ("In a preferred embodiment, the speech engine 22 is ViaVoice™ from IBM®."). Indeed, the provisional application from which Ross depends, states that "[w]hile we certainly expect to make use of IBM's ViaVoice speech engine, we would be able to easily replace it with a better alternative if one came along." EX1016, 3. Thus, a POSITA would have been motivated to look for and rely on available speech recognition methods for Ross's "speech engine," including Ittycheriah (also developed by IBM). Lopresti ¶¶116-119.

A POSITA would have also been motivated to implement Ross's speech engine using Ittycheriah to quickly recognize speech (faster than classical speech engines with its cache database) with its cache database of recently used keywords and probability scores. *see* Section IV.B (Ittycheriah Overview). Lopresti ¶¶120-

122. A POSITA would have recognized that there were multiple design choices to improve the accuracy and speed of speech recognition, including utilizing recently used keywords and probability scores. Lopresti ¶¶120-122. 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 ¶¶120-122.

2. Ross and Ross018

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 ¶¶123-127. Ross and Ross018 share Provisional Application No. 60/261,372. EX1005, Cover; EX1009, Cover. 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 ¶124.

The shared provisional explains that the two patents are complementary, with the technology recited in 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 ¶127. 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 ¶¶123-127.

B. Reasonable Expectation of Success

A POSITA would have had a reasonable expectation of success in combining Ross, Ittycheriah, and Ross018. Lopresti ¶¶128-131. POSITA would have used Ittycheriah's speech recognition process to implement the details of Ross's speech engine 22, as shown below in Figure 2, which Ross acknowledges is any commercially-available speech recognition engine:

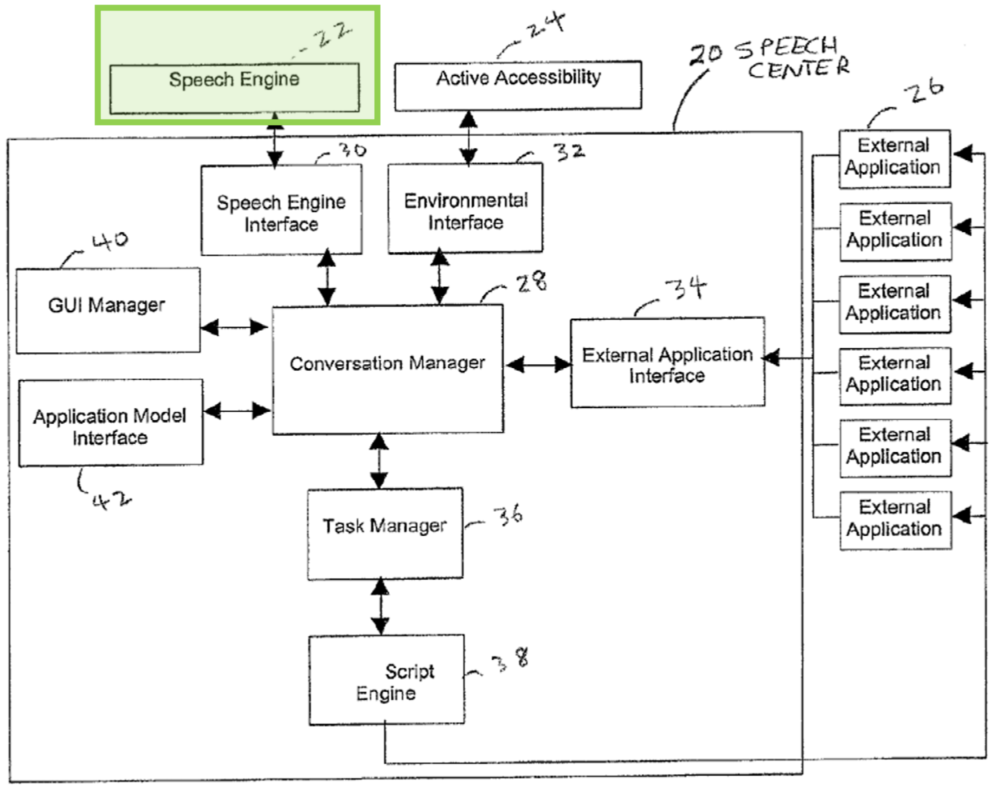


Fig. 2

EX1005, Fig. 2 (annotated)

This is particularly true since IBM developed both Ross and Ittycheriah. Indeed, a POSITA would have understood that Ittycheriah's speech recognition would have been compatible with Ross since Ittycheriah discloses that its decoded keywords can be used with external speech-enabled applications to perform their respective automatic functions. EX1007, 4:63-5:3; Lopresti ¶¶129-130.

In addition, Ross018 is an explicit complement to Ross, and is thus compatible with Ittycheriah for the same reasons. EX1016, 8. A POSITA would 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 ¶131.

C. Independent Claim 5

1. Preamble

To the extent limiting, Ross discloses the preamble. Lopresti ¶¶132-140. 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, Abstract. Ross thus teaches *a method for processing natural language speech utterances* and the speech-enabled applications are *context-specific domain agents.*

Ross discloses a "speech center" that processes *natural language speech utterances*. EX1005, 5:34-38; Lopresti ¶135. Ross discloses examples of its methods responding to natural language requests, such as, "[i]f the speech center **20** hears a phrase such as 'print the first message' or 'print the first appointment,' the context manager **50** can readily figure out the intended target application **26** for the uttered sentence." EX1005, 8:25-33. Ross also 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, 2:25-31 ("[T]he user asks what is the current stock price for a particular company and obtains the stock price from stock monitoring service."), 2:31-36 ("[T]he user makes a request to take a letter, and a word processing application starts up (i.e., is launched by the speech center)."). A POSITA would have understood that these voice commands are natural language because they use phrases in human-understandable language. Lopresti ¶¶135-137.

Ross's method responding to natural language speech utterances is illustrated below in Figure 1, which shows receipt by a processor of "input spoken utterance 14," which is a "voice command or other speech input from a user of a computer system":

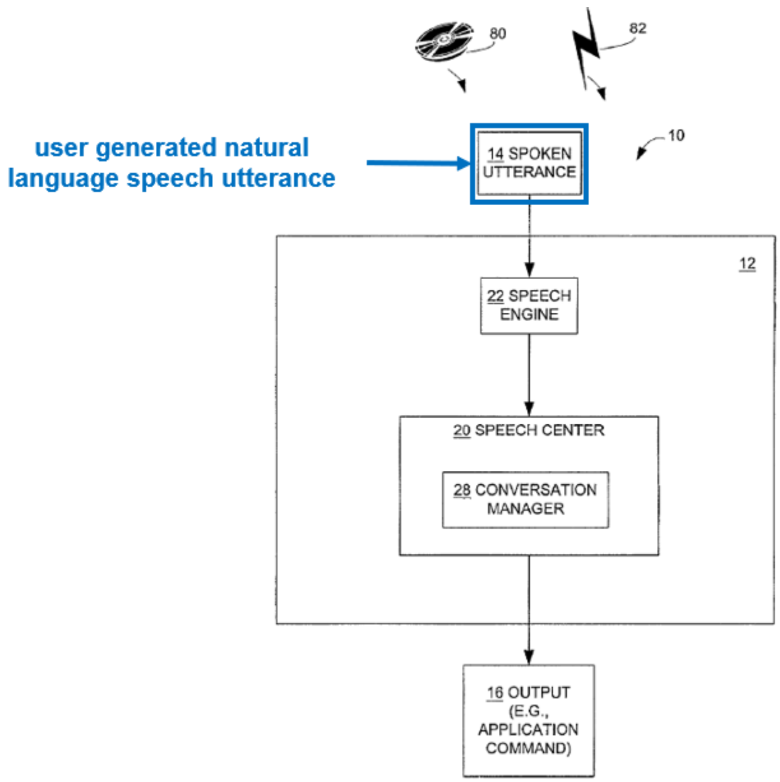
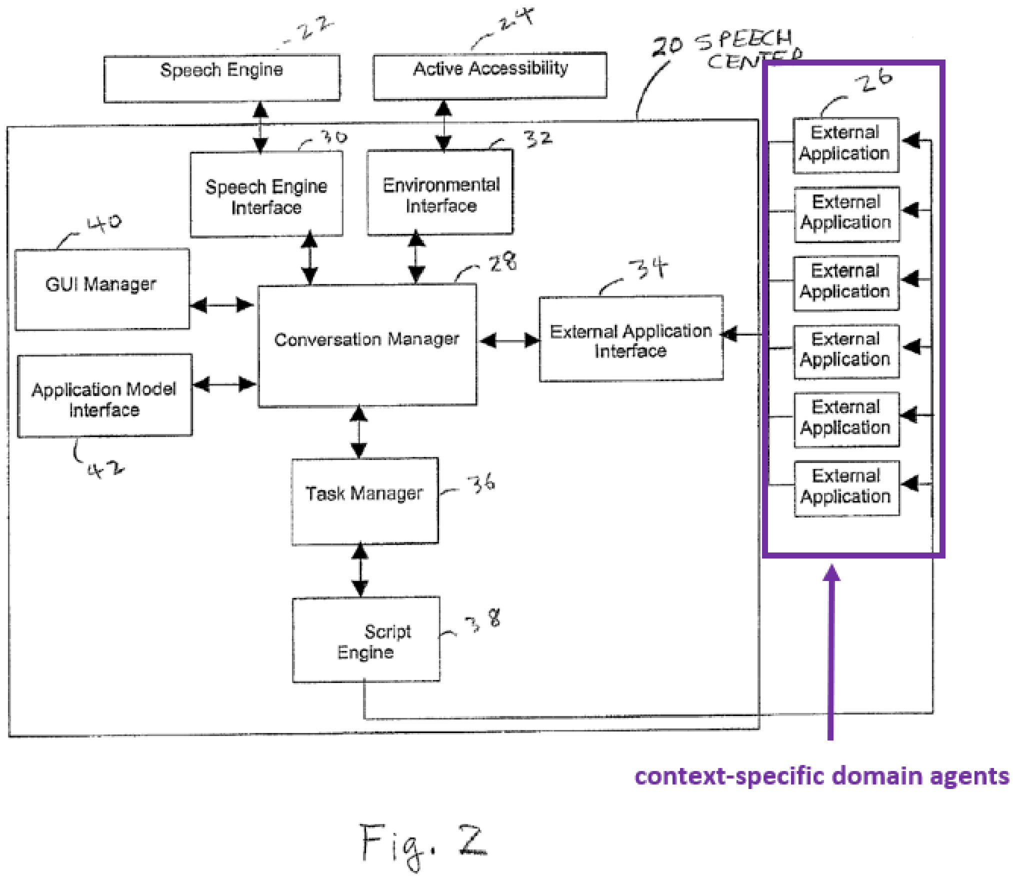


FIG. 1

EX1005, Fig. 1 (annotated)

EX1001, 3:41-53; Lopresti ¶¶137-138.

The method is responsive to the user-generated natural language speech because it produces an output 16, which is "a command or other input based on the spoken utterance 14 directed to the speech enabled external application 26." EX1005, 3:41-53; Lopresti ¶138. The external applications are each a *context-specific domain agent* that respond to the user's spoken utterance, as shown below as item 26 in Figure 2:



EX1005, Fig. 2 (annotated)

EX1001, 2:22-36; Lopresti ¶¶138-139; *See also* limitations 5[d], 5[e] (discussing the *domain agents*). Specifically, Ross "maintains a context list of speech enabled applications" that the user has accessed, and teaches "deciding the target application for a particular utterance" based in part on the determined context. EX1005, Abstract, 2:5-8. Ross teaches that the "associated application 26 is the one that should be targeted to receive the corresponding command" when its parse is accepted by the grammar of a specific speech-enabled external application. Ross EX1005, 8:33-36; Lopresti ¶¶138-140.

2. 5[a]: "receiving…"

Ross discloses this limitation. Lopresti ¶¶141-144; *see* Section VII.C.1. Ross discloses that "[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)," which is *receiving the user generated natural language speech utterance*. EX1005, 3:45-48; Lopresti ¶¶141-144. Ross discloses that the input speech is received by a "speech center," with a processor, which is *a speech unit coupled to a processing device*. EX1005, 3:41-45 ("[C]omputer system 10 includes a digital processor 12 which hosts and executes a speech center system 20 and speech engine 22 in working memory."). A POSITA would understand a "voice command" is *a request*. EX1005, 2:22-36, 3:45-48, 9:40-43 ("[T]he spoken utterance including a command that is able to be associated with multiple applications…."); Lopresti ¶144.

3. 5[b]: "recognizing…"

Ross in view of Ittycheriah discloses this limitation. Lopresti ¶¶145-151. As explained above in Sections VII.A and VII.C.2, Ittycheriah acts as the speech engine in Ross that is used 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. The speech engine is *coupled to the processing device*. EX1005, 3:41-45 ("[C]omputer system 10 includes a digital processor 12 which hosts and executes a speech center system 20 and speech engine 22….”); Lopresti ¶146.

Ittycheriah recognizes words and phrases contained in a received utterance using a "cache database" of "keywords…frequently used by a user, as well as score information attributable to the keywords." EX1007, 1:55-58. Ittycheriah defines "keyword" to "include such *words and phrases contained in the utterance* as names, commands, and sentences which include both functional and non-functional words." 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." EX1007, 3:62-64, 4:48-53; Lopresti ¶¶147-148.

By 2002, a POSITA would understand that such Ittycheriah's cache database and large-scale language database of "words and phrases" that include "names, commands, and sentences" is *one or more dictionary and phrases tables*. Lopresti ¶¶139-150; EX1026 (defining "dictionary" as "a list of words stored in machine-readable form for reference"); EX1025 (defining "dictionary" as "a list (as of phrases, synonyms, or hyphenated instructions) stored machine-readable form); EX1019, 73; Lopresti ¶¶149-150. For example, IBM's ViaVoice™ product defined the term "vocabulary" to be "a list of words provided with the product or obtained separately which can be recognized correctly when spoken by a user." EX1017, 107. The ViaVoice™ product also allowed users to add phrases. EX1017, 58. Accordingly, a POSITA reading Ittycheriah would understand that its cache database and large-scale language database would be understood as a "dictionary and phrase tables," as claimed in the '006 patent. Lopresti ¶¶150-151.

4. 5[c]: "parsing…"

Ross in view of Ittycheriah discloses this limitation. Lopresti ¶¶152-165. The '006 patent explains that a *parser* "transforms [recognized words and phrases] into complete commands and questions." EX1001, 11:19-21. A POSITA would understand that Ross's "speech center" includes a *parser* as part of its "context manager," because it transforms recognized text into queries for the external applications. Lopresti ¶¶152-165. Ross's "speech center," including its "context manager" is *coupled to the processing device*. EX1005, 3:41-45 ("[C]omputer system 10 includes a digital processor 12 which hosts and executes a speech center system 20[.]"); Lopresti ¶154.

As discussed in Section VII.C.3 (Ground 1, 5[b]), the output of Ittycheriah's is at least one "decoded keyword…representative of the input speech segment." EX1007, 2:5-7. This "decoded keyword" (i.e., the to be *parsed information*) thus includes *one or more recognized words and phrases*, including "names, commands, and sentences which include both functional and non-functional words." EX1007, 2:16-19; Lopresti ¶¶155-156. A POSITA would understand that the "decoded keyword" includes *information relating to the utterance*, because it is derived from the spoken input. Ittycheriah's "decoded keyword" therefore corresponds to the "recognition message" in Ross because it is a "speech representation recognized by the speech engine 22." EX1005, 7:1-2; EX1007, 4:63-53; Lopresti ¶157 .

A POSITA would have thus found it obvious to use Ittycheriah's "decoded keyword" as a recognition message in Ross's to *determine a meaning associated with the utterance and a context associated with the request contained in the utterance*. Lopresti ¶¶158-165. Ross's Figure 5 below, illustrates the process Ross uses to determine a context for the at least one request using a recognition message:

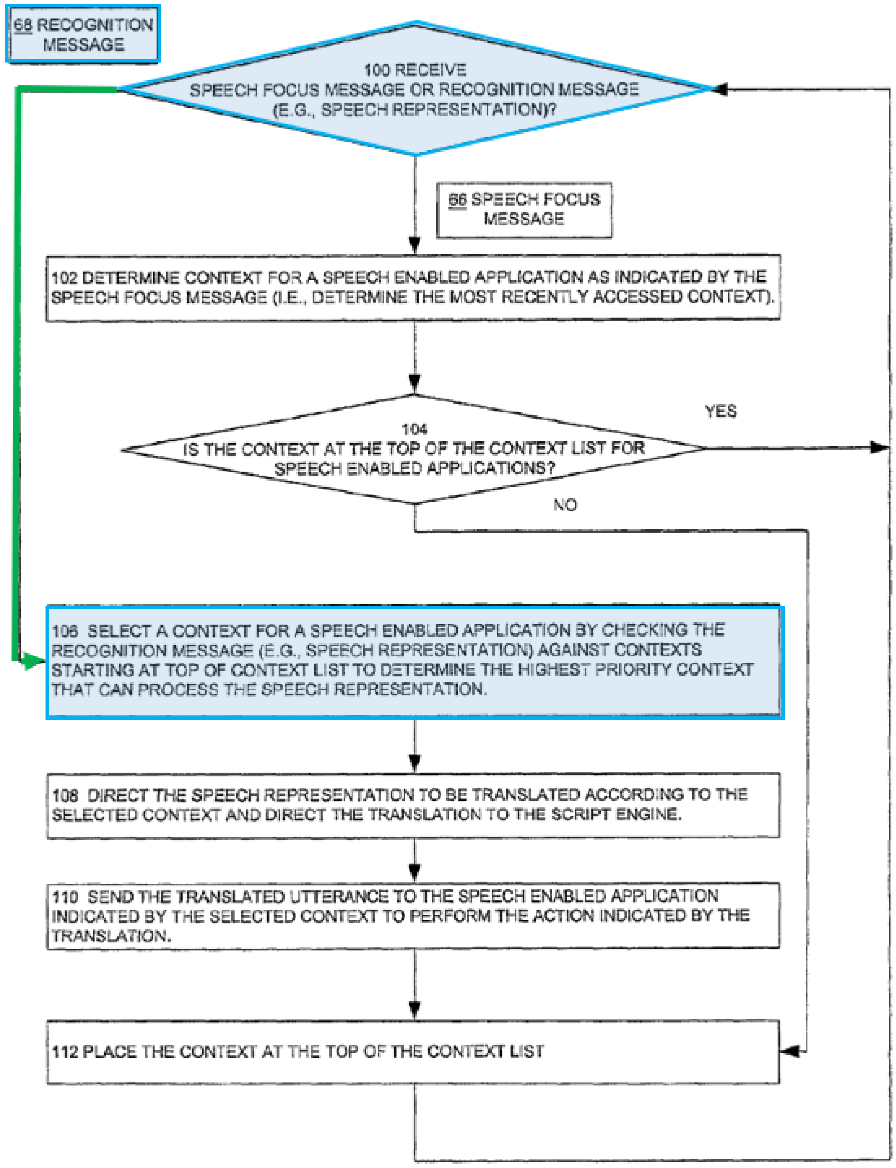


FIG. 5

EX1005, Fig. 5 (annotated)

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. 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 ¶¶159-161.

Ross describes how determining a meaning and the context for the request *is associated with the request contained in the utterance*. Lopresti ¶¶162-165. 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 grammar:

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

EX1005, 7:60-66. 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 ¶¶162-165.

5. 5[d]: "formulating…"

Ross in view of Ross018 discloses this limitation. Lopresti ¶¶166-169. While Ross discloses formulating the request in accordance with a grammar used by a domain agent, it does not provide details. Ross018 does. *Id.* Ross018 "is basically an expansion" of Ross's conversation manager and uses an additional semantic analysis and syntax manager. EX1016, 8; EX1009, Abstract ("The syntax manager combines the ontology, lexicon, and syntax definitions to generate a grammatic specification" of a domain). Combining Ross with the expanded parsing steps of Ross018 therefore discloses each of the formulating steps. Lopresti ¶¶166-169.

a. 5[d.i]: "determining…"

Ross in view of Ross018 discloses this limitation. Lopresti ¶¶170-185. Ross discloses a *domain agent associated with the determined context*. The '006 patent 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, 2:41-42 ("Any application that supports a robust automation user interface can participate."). A POSITA would thus understand a "domain agent" as claimed to include an autonomous executable software module specific to each application area. Lopresti ¶¶171-175.

Ross's "external applications 26" that "communicate with the speech center **20**, a speech engine **22**, and active accessibility module **24**," each have a defined grammar, "along with associated translations for a script engine **38**." EX1005, 4:16-21, 6:29-31. Each of these "external applications" is configured to respond to queries and/or commands within a particular domain, as they are "speech-enabled applications" that respond to speech. EX1001, Abstract, 2:22-36; Lopresti ¶172. Ross further 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. Based on these disclosures in Ross, a POSITA would thus understand that Ross's external applications are autonomous executables triggered by the script commands issued by Ross's task manager. Lopresti ¶¶48-51, 172-175. Accordingly, each of Ross's external, speech-enabled applications is a *domain agent* as claimed in the '006 patent. *Id.* This also aligns with Appendix B to PO's Infringement Contentions, that identifies external applications (weather and news) accessed via application program interfaces (APIs) as "domain agents." EX1014, 14-17.

Ross018 adds to Ross's "conversation manager" (i.e., *the parser*) 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 ¶¶176-177. According to Ross018, each speech-enabled external application (i.e., *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, Ross018 discloses that after an utterance is recognized, "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 ¶¶176-177.

Second, 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*. Lopresti ¶¶178-185. Specifically, Ross018 discloses that its 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" are such things as the "action," the object or "patient" of the action ("the thing acted upon by an action"), and the "destination." EX1009, 10:5-15; *see id.* at 7:35-40; Lopresti ¶¶178-179. These values 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, 7:41-51; Lopresti ¶¶180-183. Indeed, by 2002, a POSITA would understand and expect that grammars accepted by speech-enabled applications by 2002 would include both required and optional values. Lopresti ¶¶183-185.

The '006 patent 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 Ross-018 would be inclusive of all values contained in the utterance, a POSITA would understand such triples would include both all optional and required values. EX1005, 9:49-10:35; Lopresti ¶185. These attribute and object values in Ross018 align with how "values" is used in the '006 patent as criteria that describe different attributes, such as "gender" and "occupation" as personal profile values. EX1001, 14:32-50, 18:12-16; Lopresti ¶185. Accordingly, a POSITA would understand that Ross018's determination of attribute and object values teach *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 ¶¶183-185.

b. 5[d.ii]: "extracting…"

Ross in view of Ross018 discloses this limitation. Lopresti ¶¶186-191. The '006 patent uses the terms "criteria," "parameters" and "values," somewhat interchangeably. *See* EX1001, 11:65-67, 14:15-19.[2] However, a POSITA would understand that "criteria" can refer to the attributes of a request (e.g., location) and a parameter can refer to the attribute's value (e.g., Virginia). Lopresti ¶187.

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 ¶¶188-191. In generating its "propositions," which are the set of attribute-object-value triples, Ross018 converts the recognized utterance into frame representations that serve as input to the reasoning facility. EX1009, 9:59-62; Lopresti ¶¶188-189. The one or more "criteria" of the '006 patent would include the one or more "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 shown in the below frame structure:

[2] PO points to the same evidence for "values," "criteria," and "parameters" in its contentions without differentiating between these terms. EX1014, 18-25.

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

EX1009, 9:59-10:35; Lopresti ¶¶189-190.

Ross018 develops its attribute-object-value triples in *using procedures sensitive to the determined context*. Lopresti ¶¶189-190. Namely, as discussed with respect to Section VII.C.4 (Ground 1, 5[c]), prior to processing the initial semantic representation, Ross's context manager 58 determines that the recognized utterance is associated with a context of a specific external application. Ross018 further discloses that its frame structures (from which the attribute-object-value triples are extracted) are based in part on the "domain model," which includes the concepts, grammatic specifications, and semantic specifications of one or more 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 ¶¶189-190.

c. 5[d.iii]: "inferring…"

Ross in view of Ross018 discloses this limitation. Lopresti ¶¶192-199. Ross018 discloses a "reasoning facility" that "performs the reasoning process for the conversation manager." EX1009, 10:36-37. As Ross018 explains, using the command "make this column green," the initial attribute-object-value triples are able to "make much explicit," "[t]he reasoning facility 52 must still determine which column to operate upon." EX1009, 10:31-35. A POSITA would thus understand that Ross018's reasoning facility must identify *one or more further criteria and one or more further parameters associated with the request*. Lopresti ¶¶193-194.

Ross018 accomplishes this with an *inference engine* that *uses a dynamic set of prior probabilities or fuzzy possibilities*. Lopresti ¶¶195-199. 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 ¶195. 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.[3] If a match is found, "[t]hese become new goals for the inference engine of the reasoning facility 52 to achieve." EX1009, 10:57-58. A POSITA would thus understand that Ross018's reasoning facility *infers one or more further criteria and one or more further parameters associated with the request*. Lopresti ¶¶195-197. For example, Ross018 describes a speech utterance of "Create an appointment with Mark at 3 o'clock tomorrow," followed by the utterance "Change that to 4 o'clock." EX1009, 11:8-10. The speech center would be required to infer what "appointment object whose time attribute is changing" as an additional criteria and parameter. EX1009, 11:10-14. Using the logic of the inference engine (*fuzzy possibilities*), Ross018's reasoning facility selects the proper referent (of the appointment) based on the conversational record (e.g., recency) and the goal structure of the conversation as discussed above. EX1009, 11:14-16. Lopresti ¶¶195-197.

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

---

[3] The '006 patent does not define *fuzzy possibilities*. PO's contentions suggest that it includes any logic to infer criteria. EX1014, 24-25. Petitioner thus applies PO's broad understanding here.

facility." EX1009, 10:59-64. These goals are based on the "conversation record," 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 conversational record is constantly evolving, a POSITA would understand the logic of the inference agent is dynamic. Lopresti ¶199.

d. 5[d.iv]: "transforming…"

Ross in view of Ross018 discloses this limitation. Lopresti ¶¶200-205. Ross018 discloses that its 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. This is because Ross018's reasoning facility, in generating its propositions (the attribute-object-value triples) "can contain embedded script function calls" for external applications (i.e., *domain agents*). A POSITA would thus understand 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 ¶¶201-204. 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 ¶¶204-205.

Moreover, a POSITA would have found it obvious based on Ross018 that the script call or command (i.e., 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 above in Section VII.C.5.d (Ground 1, 5[d.iv]). Ross018 discloses that formulating its response to a recognized utterance, "[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 that in formulating the script call or command (i.e., the *token*), all of the required values and optional values would be included or Ross018 would prompt the user for more information, including for additional values. Lopresti ¶205.

6. 5[e]: "processing…"

Ross discloses this limitation. Lopresti ¶¶206-207. Ross provides several examples of *processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance*. Ross describes using an external application (*domain agent*), to process an utterance about a "next appointment" and "obtain[ing] information from a calendar program." EX1005, 2:22-30 (utterance about current stock price obtains the stock price from a stock monitoring service); *id.* at 2:31-36 (launching a word processing application in response to a request to take a letter), 8:47-9:6 (print requests); *id.* at 9:44-50 ("launching the selected speech enabled application as needed and performing the function of the command."). Lopresti ¶¶206-207.

7. 5[f]: "presenting…."

Ross discloses this limitation. Lopresti ¶¶208-210. As detailed above in Section VII.C.6 (Ground 1, 5[e]) once a response to the utterance is generated by the domain agent, its results are presented to the user. Alternatively, to the extent that this is not considered to be "via the speech unit," Ross018 additionally discloses presenting the response via a user interface. EX1009, 12:11-46 ("speech engine 22 converts the response output…[to] be provided to the user through a speaker of the computer system."); Lopresti ¶¶208-210.

VIII. GROUND 2:

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 based on relevance. Lopresti ¶¶212-216. 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 understand that the Internet provided a robust source of information. Lopresti ¶¶213-214. Accordingly, a POSITA would have been motivated to look for and rely on Internet searching mechanisms to aid in retrieving and processing responses to a user's queries (such as Aridor, also developed by IBM). Lopresti ¶¶213-214.

A POSITA would understand that Aridor's metasearch functionality would 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 ¶¶214-215. 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 set forth in Section VII.C.6-7 (Ground 1, 5[e]-5[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 (question or command) to find the most relevant information. Lopresti ¶¶214-215. A POSITA would also be 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 ¶216.

A. 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 information sources to the Internet. Lopresti ¶¶217-219. 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 ¶¶217-219.

B. Independent Claim 6

1. Preamble

To the extent limiting, Ross in view of Aridor discloses this limitation as set forth above in the preamble to claim 5, and below. Section VII.C.1; Lopresti ¶221.

2. 6[a]: "receiving…"

Ross discloses this limitation for the same reasons as 5[a]. Section VII.C.2. A POSITA would further understand a request in Ross *includes at least one command and at least one question*. 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 ¶222.

3. 6[b]: "parsing…"

Ross discloses this limitation for the same reasons as 5[c]. Section VII.C.4; Lopresti ¶223.

4. 6[c]: "formulating…;"

Ross discloses this limitation for the same reasons as 5[d]. Section VII.C.5; Lopresti ¶224.

5. 6[d]: "processing…"

Ross in view of Aridor discloses this limitation for the same reasons as 5[e], in addition to the below. Section VII.C.6; Lopresti ¶225.

a. 6[d.i]: "directing…"

Ross in view of Aridor discloses this limitation. Lopresti ¶¶226-229. Ross describes requests that include *at least one question and at least one command* (Section VIII.C.2, Ground 2, 6[a]). Ross, however, is explicit on how its domain agents process and return results to the user's query, including via the Internet. Adding Aridor's metasearch to Ross discloses that *at least one command* is *directed to one or more local or remote processing devices to execute* the command. 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," which can "any suitable general purpose computer processor having a connection to network." EX1010, 8:1-12; Lopresti ¶¶226-229

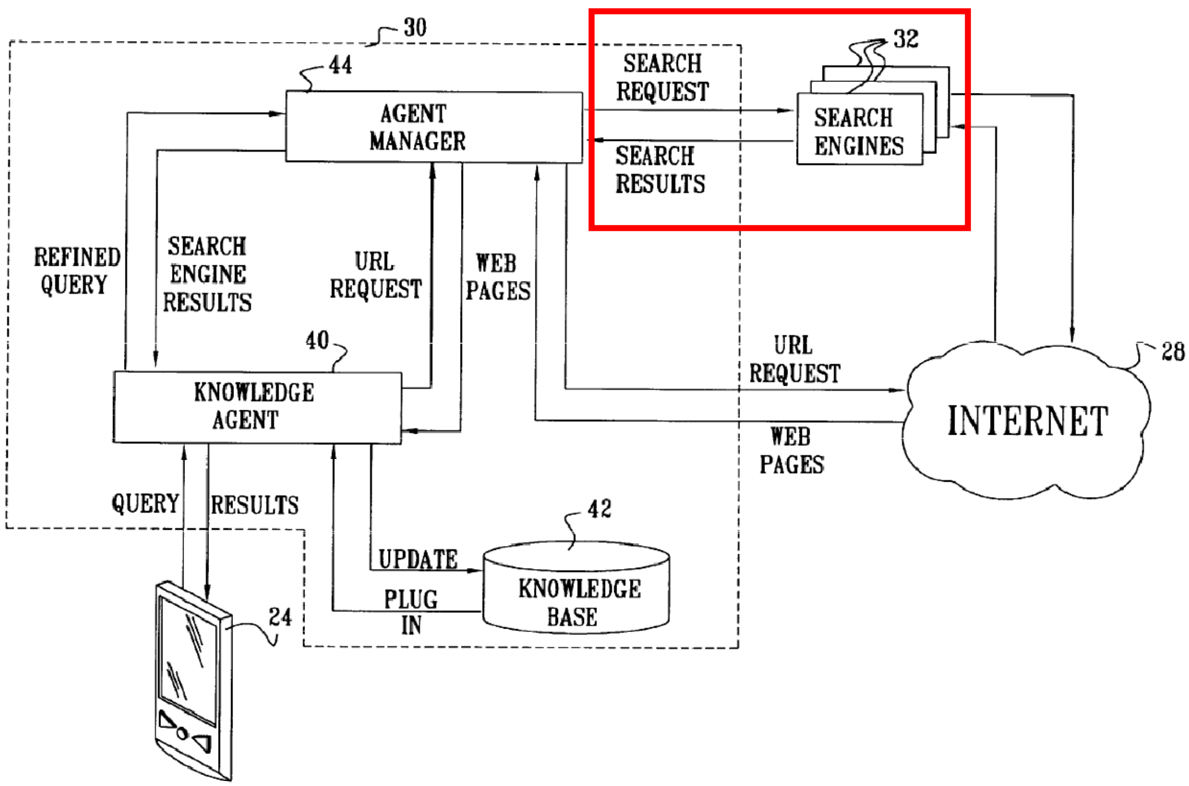
b. 6[d.ii]: "querying…"

Ross in view of Aridor discloses this limitation. Lopresti ¶230. As discussed as to 6[a], Ross in combination with Aridor allows for 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 VIII.C.5.a; Lopresti ¶230.

i. 6[d.ii.A]: "submitting…"

Ross in view of Aridor discloses this limitation. Lopresti ¶¶231-235. 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 VIII.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 ("[T]he agent manager performs a meta-search, by first invoking several general-purpose search engines specified by user 22, such as AltaVista, Google or Hotbot[.]"), Fig. 2. This is also shown in Figure 2 of Aridor, in which a search request is submitted to multiple search engines:

FIG. 2



EX1010, Fig. 2 (annotated)

A POSITA would understand that a metasearch that sends the same search request multiple times to different search queries *submits duplicate queries to distinct information sources*. Lopresti ¶¶231-235.

ii. 6[d.ii.B]: "asynchronously receiving…"

Ross in view of Aridor discloses this limitation. Lopresti ¶¶236-238. Aridor discloses that 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 understand that each user-selected search engine sends results *asynchronously*. Lopresti ¶¶237-238. For example, Aridor discloses that it employs general-purpose search engines such as AltaVista, Google or Hotobot, which are separate search engines that operate independently from one another, and in doing so, operate asynchronously. EX1010, 8:52-59; Lopresti ¶¶237-238.

iii. 6[d.ii.C]: "scoring…"

Ross in view of Aridor discloses this limitation. Lopresti ¶¶239-243.

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*. Following receipt of the metasearch results, 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." EX1010, 11:19-34. "Textual similarity scores 68 are computed to measure the 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 ¶¶240-241.

Aridor *uses a dynamic set of prior probabilities or fuzzy possibilities* to score the relevance of the results. Lopresti ¶242. In computing both the "textual score" and the "link topology score," Aridor employs the use of 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 thus understand that the history scores represent *a prior probability or fuzzy possibility* that the site will provide a relevant result to a user's query. Lopresti ¶242. 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 ¶243.

(a) 6[d.ii.C.1]: "determining…"

Ross in view of Aridor discloses this limitation. Lopresti ¶¶244-246. Upon receiving the formatted script call from Ross, Aridor's knowledge agent processes the question as a text query (i.e., a "keyword-based query such as those typically submitted to a general-purpose Web search engine."). EX1010, 9:42-49. Aridor does so by creating 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 understand that in creating the query profile thus 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 (i.e., the requested information). Lopresti ¶246.

(b) 6[d.ii.C.2]: "applying one or more scraping criteria…"

Ross in view of Aridor discloses this limitation. Lopresti ¶¶247-251. Aridor discloses how the query is submitted to "one or more general purpose search engines," the results are collected, which Aridor calls the "root set 54 of sites 26." EX1010, 10:18-23. Aridor then adds to the "root set," sites already known to the knowledge agent (the "knowledge base") as having relevant and quality information and backward and forward sites to those in the returned results, that collectively form a "final site list" of potential results. EX1010, 10:40-11:18; Lopresti ¶248.

Aridor then *applies one or more scraping criteria* to the *asynchronously received* results returned from the sites on 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. For textual similarity, Aridor scrapes the words in each page and divides the words into three grounds: "Strong text" (title words or large fonts); "medium text" (highlighted or bold text or small font headers); or regular texts. EX1010, 11:46-51. Aridor then applies weights to each type of word and computes a "query textual score," and, in a similar way, a "domain score" based on known, relevant documents in the knowledge base. EX1010, 12:13-60. The scores are "normalized and combined to create the overall textual similarity score 68 for each site[.]" EX1010, 12:60-62; Lopresti ¶¶249-250.

Aridor also applies a *scraping criteria* to the site list to compute a "link topology score," based on the "anchor text" (language a pointing page uses to describe a destination page…as the search results page of the general-purpose search engine) and "anchor links" (the links that connect 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 those added to root sites within the site list, the topology score has greater significance in identifying an answer to the query (the *value for the determined token*). EX1010, 14:11-23; Lopresti ¶251.

(c) 6[d.ii.C.3]: "evaluating…"

Ross in view of Aridor discloses this limitation. Lopresti ¶¶252-255. Aridor's "textual score" and "link topology score" both use *a dynamic set of prior probabilities or fuzzy possibilities* by way of "domain textual scores" and "anchor links," computed using Aridor's dynamic knowledge base. Section VIII.C.5.iii (Ground 2, 6[d.ii.C]). Once the one or more values are extracted, 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 ("[T]he algorithm takes into account textual affinity to the particular query and to the domain of interest, as well as topological information for finding the most 'authoritative' pages."), 14:56-15:11; Lopresti ¶¶253-54. A POSITA would understand 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 ¶¶254-255.

(d) 6[d.ii.C.4]: "selecting…"

Ross in view of Aridor discloses this limitation. Lopresti ¶¶256-258. 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 ¶¶256-258.

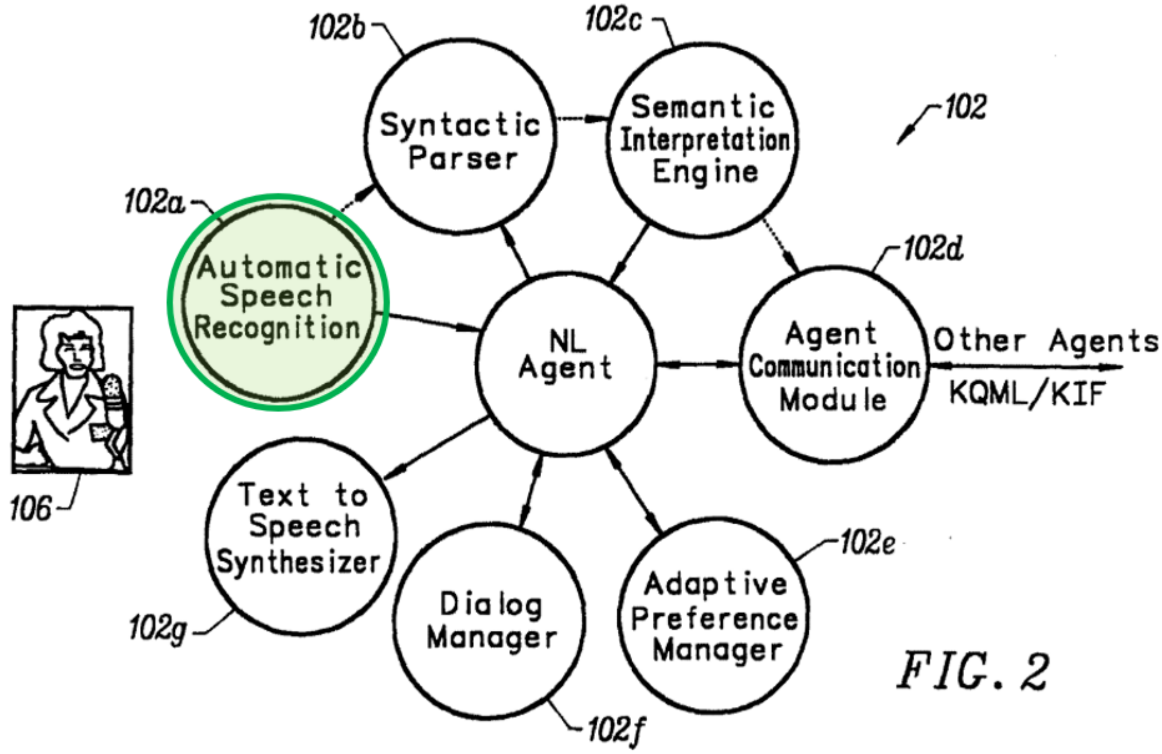
6. 6[e]: "presenting…."

Ross discloses this limitation for the same reasons as 5[f]. Section VII.C.7; Lopresti ¶259.

IX. GROUND 3:

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 ¶¶261-269. Hartono discloses that speech is input into an "automatic speech recognition" module, shown below (item 102a) in Figure 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 thus be motivated to look for available speech recognition systems to use to with Hartono. Lopresti ¶¶261-264.

Baker is such a speech recognition system developed by Dragon Systems, Inc., a well-known developer of speech recognition software. EX1008, Cover; Lopresti ¶¶43, 47, 265-266. A POSITA would understand that 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 ¶266. Hartono also employs the Government and Binding approach to its parsing, which is based on the "implied syntactic structure in English language sentences." EX1006, 12:1-10. Baker's speech recognition system that 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 ¶267.

A POSITA would also be 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 ¶¶268. This allows Baker to avoid building data-intensive statistical language models or data-training sets. EX1008, 3:46-4:56 (describing the 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 be motivated to use Baker's faster processing in speech recognition to more quickly respond to specific user requests. Lopresti ¶268-269.

A. Reasonable Expectation of Success

A POSITA would have had a reasonable expectation of success in combining Hartono's and Baker's teachings. Lopresti ¶¶270-272. Hartono's automatic speech recognition system 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 ¶¶270-272.

B. Independent Claim 5

1. Preamble

To the extent limiting, Hartono discloses the preamble. Lopresti ¶274. Hartono discloses a "method for receiving commands and/or adaptively outputting results using a natural language speech interface[.]" EX1006, 8:8-10, 1:19-23. A POSITA would understand that Hartono *processes natural language speech utterance with context-specific domain agents* that Hartono refers to as task or application agents. EX1006, 4:20-25 ("[T]he natural language interface for a computer system includes a natural language agent adapted to receive and interpret the natural language user input and to output an output command and at least one application agent adapted to receive and further interpret the output command…and to output an executable instruction to an application program."), 5:16-24, 10:13-20; *see also* Section IX.C.5.a (Ground 3, 5[d.i]).

Figure 1 of Hartono illustrates the relationship between the natural language agent (102), domain agents (104a through 104d), and external applications (112):

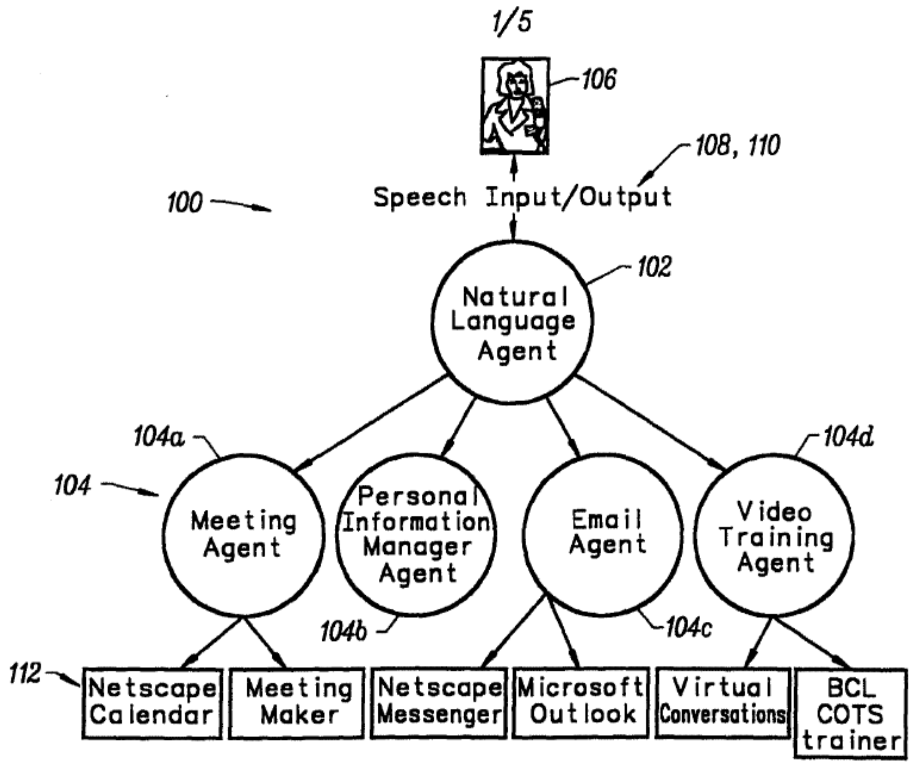


FIG. 1

EX1006, Fig. 1, 9:5-10 ("As is generally shown, the user 106 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 class task agents 104a-d."); Lopresti ¶¶277-282.

2. 5[a]: "receiving…"

Hartono discloses this limitation. Lopresti ¶¶278-282. 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; *id.*, 1:20-23, 8:8-10, 4:20-25, 5:11-12, 5:17-23. Hartono's teaches 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; Lopresti ¶¶279-280.

A POSITA would understand a "command" to be *a request* as claimed in the '006 patent. Lopresti ¶¶274-282. For example, Hartono discloses "'compose a letter to John Smith,' 'please being drafting a letter to John Smith,' or 'can you create a letter for my friend, John Smith?'" EX1006, 29:2-6, 25:20-23; Lopresti ¶¶281-282.

3. 5[b]: "recognizing…"

Hartono combined with the speech recognition system of Baker does disclose this limitation. Lopresti ¶¶283-287. Hartono recognizes that "[a]utomatic speech recognition systems" were readily and commercially available off-the-shelf and that "any suitable off the shelf speech recognition systems may be used as the automatic speech recognition system 102a." EX1006, 11:4-12.

As discussed in Section IV.E, Baker *uses information in one or more dictionary and phrase tables* to recognize *one or more words and phrases contained in the utterance*. Lopresti ¶¶285-287. Specifically, Baker teaches that "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 passages "to determine the likelihood that the choice phrase represents the utterance being analyzed." EX1008, 21:19-47, 21:49-22:6; Baker's "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 ¶285.

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 ("Grolier's Encyclopedia"), 16:8-12 (Encyclopedia Brittanica); Lopresti ¶¶286-287.

4. 5[c]: "parsing…"

Hartono discloses this limitation. Lopresti ¶¶288-294. Once Baker recognizes the utterance, Hartono discloses inputting the recognized words or phrases into a "natural language syntactic parser 102b" and "natural language semantic interpreter 102c" *(parser)*. Lopresti ¶289. Hartono discloses first syntactically parsing the natural language 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 ¶290.

Hartono's semantic interpreter then applies "context sensitive methodologies" to the syntactic parse to *determine a meaning associated with the utterance and a context associated with the request contained in the utterance*. EX1006, 13:13-21; Lopresti ¶¶291-294. Hartono "takes the syntactic parse of the spoken language request and maps it to a generic concept frame," such as shown 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 *the one or more recognized words or phrases* to determine a meaning and a context, by using recognized verbs as "action-concepts" and recognized 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 ("The system then determines which application context should be responsible for interpreting and executing that command concept.");

Lopresti ¶¶292-294.

5. 5[d]: "formulating…"

Hartono discloses this limitation. As detailed below, Hartono teaches each of the following steps included in formulating the request. Lopresti ¶295.

a. 5[d.i]: "determining…"

Hartono discloses this limitation. Lopresti ¶¶296-303. Like the '006 patent, Hartono "utilizes an agent-based architecture" that uses "application class task agents for each class of applications," as illustrated below in Figure 1:

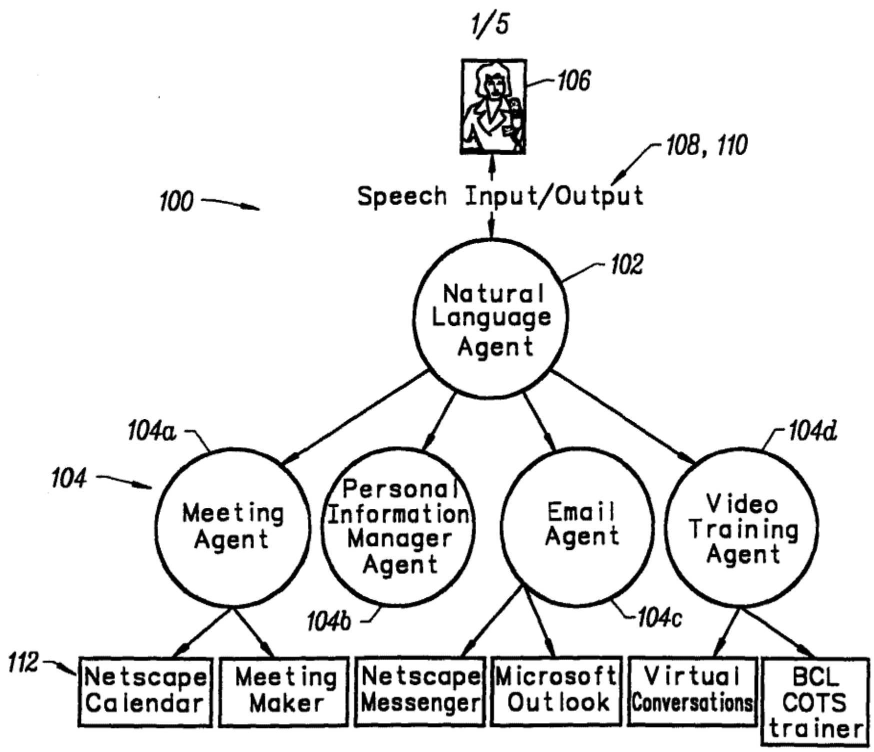


FIG. 1

EX1006, Fig 1, 8:10-13, 9:9-14 ("Examples of the task agents 104a-d shown in FIG. 1 are meeting agent 104a, personal information manager agent 104b, email agent

104c, and voice training agent 104d."). A POSITA would understand that Hartono's "task agents" (or "application class agents") correspond to the claimed *domain agent* in the '006 patent because they are software modules specific to an application or application area. EX1006, 23:19-20 ("Each application class agent 104 preferably works with a single class of applications 112 that have similar conceptual operations."), 28:7-29 (Table IV listing "task agents"); Lopresti ¶¶297-298. Hartono's task agents are autonomous executables as each agent "includes a process-specific execution module for each application." EX1006, 30:15-20 ("For example, the word processing task agent may include an execution module for MICROSOFT WORD"), 36:7-9; Lopresti ¶¶298-299.

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 ¶¶300-302. 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 ¶302. A POSITA would thus understand that 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 ¶¶301-303.

b. 5[d.ii]: "extracting…"

Hartono discloses this limitation. Lopresti ¶¶304-309. The '006 patent uses the terms "criteria," "parameters" and "values," as claimed in the '006 patent somewhat interchangeably. *See* EX1001, 11:65-67, 14:15-19.[4] However, a POSITA would generally understand that "criteria" can refer to the attributes of a request (e.g., location) and a parameter can refer to the value of that attribute (e.g., Virginia). Lopresti ¶305.

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 ¶¶306-308. Specifically, Hartono "takes the syntactic parse of the spoken language request and maps it to a generic concept frame," thus

---

[4] PO points to the same evidence for "values," "criteria," and "parameters" in its contentions without differentiating between these terms. EX1014, 18-25.

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 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 ("The natural language interface parses the user's input and semantically maps it into a knowledge concept structure."); Lopresti ¶¶306-308.

c. 5[d.iii]: "inferring…"

Hartono discloses this limitation. Lopresti ¶¶309-313. 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 understand Hartono's probability list to be a *set of prior probabilities or fuzzy possibilities*. Lopresti ¶¶310-313. 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 ¶311.

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 ¶¶312-313. 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 understand 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 ¶¶312-313.

d. 5[d.iv]: "transforming…"

Hartono discloses this limitation. Lopresti ¶¶314-317. Hartono teaches 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 (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 ("[T]ask application wrappers…map the complex application tasks to vendor-specific executable tasks."), 24:21-25:3; Lopresti ¶¶316-317. Hartono's "application specific operations" to be executed by the application would be 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 ¶¶316-317. A POSITA would thus understand that by using, via the task agents, the task application wrapper to translate the parsed utterance into a set of application specific operations to be executed by the application, Hartono discloses *transforming the one or more extracted criteria, the one or more extracted parameters, the one or more inferred criteria, and the one or more inferred parameters into one or more tokens having a format compatible with the grammar used by the domain agent.* Lopresti ¶¶315-317.

6. 5[e]: "processing…"

Hartono discloses this limitation. Lopresti ¶¶318-319. Hartono discloses that 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 output an executable instruction to an application program."), 4:20-25, 5:4-15, 5:21-25; 36:7-9; Lopresti ¶¶318-319.

7. 5[f]: "presenting…."

Hartono discloses this limitation. Lopresti ¶¶320-323. 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 teaches several different applications that are used to execute 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 (*the formulated request*) and present the results to the user. EX1006, 5:23-24, 9:13-25; Lopresti ¶¶320-323.

X. GROUND 4:

A. Motivation to Combine

A POSITA would have been motivated to add Aridor's metasearch functionality as one of Hartono's application class agents (*domain agents*) to collect information from the Internet and to filter and rank the results based on relevance. Lopresti ¶¶325-329. Although Hartono describes that its domain agents use the Internet, Hartono does not detail how it performs such Internet searches. EX1006, 29 (Table IV), 39:1-13. Accordingly, a POSITA would have been motivated to look for and rely on Internet searching mechanisms to aid in retrieving and processing responses to a user's queries (such as Aridor). Lopresti ¶¶325-326.

A POSITA would also understand that Aridor's metasearch functionality would be compatible with Hartono, as Aridor operates by using domain-based knowledge agents. Lopresti ¶327. Combining Aridor with Hartono 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 Hartono invoke Aridor's metasearch functionality. EX1010, 3:53-59; Lopresti ¶327. A POSITA would further be motivated to add to Hartono the "ranking and filtering" features of Aridor that process the results of the user's query (question or command) to find the most relevant information. Lopresti ¶328. In addition, a POSITA would be motivated to use Aridor's metasearch functionality because its knowledge agents are able 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 ¶¶329.

A. Reasonable Expectation of Success

Aridor expands on Hartono's ability to process natural language commands and questions by adding metasearch functionality to relevant results from the Internet. Lopresti ¶¶330-332. 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 ¶¶330-332.

B. Independent Claim 6

1. Preamble

To the extent limiting, Hartono in view of Aridor discloses this limitation as discussed in the preamble to claim 5, as well as below. Section IX.C.1; Lopresti ¶¶333-334.

2. 6[a]: "receiving…"

Hartono discloses this limitation for the same reasons as 5[a]. Section IX.C.2. Hartono provides an example of a request contained *in the utterance* that includes *at least one command* and *at least one question*, describing the request "can you create a letter for my friend, John Smith?" that includes both a question (can you create?) and a command (create a letter to John Smith). EX1006, 29:2-6; Lopresti ¶335.

3. 6[b]: "parsing…"

Hartono discloses this limitation for the same reasons as 5[c]. Section IX.C.4; Lopresti ¶336.

4. 6[c]: "formulating…"

Hartono discloses this limitation for the same reasons as 5[d]. Section IX.C.5; Lopresti ¶337.

5. 6[d]: "processing…"

Hartono in view of Aridor discloses this limitation for the same reasons as 5[e], in addition to the below. Section IX.C.6; Lopresti ¶338.

a. 6[d.i]: "directing…"

Hartono discloses this limitation. Lopresti ¶¶339-341. A POSITA would understand that for the application class agent (*domain agent*) in Hartono to perform the command on a computer (*execute the at least one command*), the command must be d*irected to a local or remote processing device*. EX1006, 33:21-35:3 (describing

Hartono's implementation on a computer that includes a multi-processor system); Lopresti ¶¶339-341.

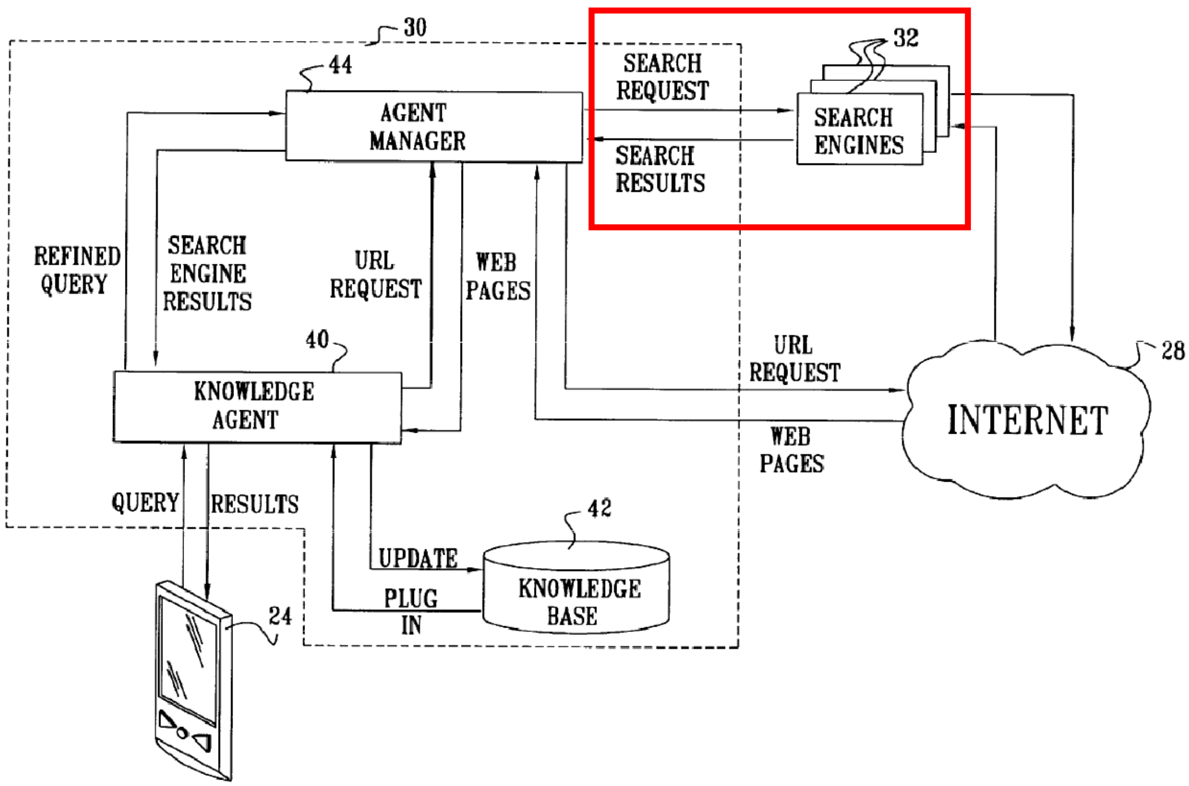
b. 6[d.ii]: "querying…"

Hartono discloses this limitation. Lopresti ¶¶342-343. 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, 29 (Table IV), 39:1-13 (listing task agents for each class of application programs, including include local (such as addresses) and network (such as internet retail sales) sources). For example, Table IV shows how certain task agents (*domain agents*) will locate information over the Internet or local information sources, such as an address book. EX1006, 29 (Table IV); Lopresti ¶¶342-343.

i. 6[d.ii.A]: "submitting…"

Hartono in combination with Aridor discloses this limitation. Lopresti ¶¶344-347. Aridor's 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. In performing the search, Aridor's agent manager and knowledge agent refines a query and" passes the refined query to a number of search engines" (*submits multiple asynchronous queries to the one or more local or network information sources*). EX1010, 4:10-16, 3:33-35, 8:22-28, 8:45-59 ("[T]he agent manager performs a meta-search, by first invoking several general-purpose search engines…such as AltaVista, Google or Hotbot[.]"); Lopresti ¶¶345-346. This is shown in Figure 2 of Aridor, where a search request is submitted to multiple search engines:

FIG. 2



EX1010, Fig. 2 (annotated)

A POSITA would thus understand that a metasearch that sends the same search request multiple times to different search queries are *duplicate queries submitted to distinct information sources*. Lopresti ¶¶346-347.

ii. 6[d.ii.B]: "asynchronously receiving…"

Hartono in combination with Aridor discloses this limitation. Lopresti ¶¶348-350. Aridor discloses that the user's query, after being refined the knowledge agent "submit[s] 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 ("Whenever the agent performs a search in response to a user-specified query, the result is a list of candidate sites that the agent has collected, examined and ranked."), Fig. 2 (showing search results received from search engines). A POSITA would understand that each user-selected search engine sends results *asynchronously*. Lopresti ¶350. For example, Aridor discloses that it employs search engines such as AltaVista, Google or Hotobot, which are separate engines that operate independently from one another, and in doing so, operate asynchronously. EX1010, 8:52-59; Lopresti ¶350.

iii. 6[d.ii.C]: "scoring…"

Hartono in view of Aridor discloses this limitation. Lopresti ¶¶351-359. Ardior describes that once it receives the metasearch results, they "rank[ed] and filter[ed]" by the knowledge agent. EX1010, 8:56-57. Aridor's 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." EX1010, 11:19-34; Lopresti ¶352. Specifically, "[t]extual similarity scores 68 are computed to measure the relevance of the pages retrieved both to the specific query and to the agent's specialized domain," and topological aspects are scored based 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 ¶¶352-354 .

Aridor *uses the dynamic set of prior probabilities or fuzzy possibilities* to score the relevance of the results. Lopresti ¶¶355-359. In computing both the "textual score" and the "link topology score," Aridor employs the use of 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 a assigned a history score, which reflects the site's relevance to the domain through the course of the use of the agent," such that high history scores are assumed to have higher quality results. EX1010, 14:44-55. A POSITA would thus understand that the history scores associated with prior sites represent *a prior probability or fuzzy possibility* that the site will provide a relevant result to a user's query. Lopresti ¶¶355-358. 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 ¶359.

(a) 6[d.ii.C.1]: "determining…"

Hartono in view of Aridor discloses this limitation. Lopresti ¶¶360-363. Aridor's knowledge agent processes the question as a text query (i.e., a "keyword-based query such as those typically submitted to a general-purpose Web search engine."). EX1010, 9:42-49. Specifically, 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 thus understand that the process of creating the query profile *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 spoken utterance for which values are sought (i.e., the requested information). Lopresti ¶¶360-363.

(b) 6[d.ii.C.2]: "applying one or more scraping criteria…"

Hartono in view of Aridor discloses this limitation. Lopresti ¶¶364-368. Once it submits the query to "one or more general purpose search engines," Aridor's knowledge agent collects the results from those search engines, which Aridor calls the "root set 54 of sites 26." EX1010, 10:18-23; Aridor then adds to the "root set," sites already known to the knowledge agent (the "knowledge base") as having relevant and quality information and backward and forward sites to those in the returned results, that collectively form a "final site list" of potential results. EX1010, 10:40-11:18; Lopresti ¶365.

Aridor then *applies one or more scraping criteria* to the *asynchronously received* results returned from the sites on the "site list" to *extract one or more values for the determined tokens*. Specifically, 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; Lopresti ¶¶366-368. For textual similarity, Aridor scrapes the words in each page and divides the words into three grounds: "Strong text" (title words or large fonts); "medium text" (highlighted or bold text or small font headers); or regular texts. EX1010, 11:46-51. Aridor then applies weights to each type of word and computes a "query textual score," and, in a similar way, a "domain score" based on known, relevant documents in the knowledge base. EX1010, 12:13-60. The scores are "normalized and combined to create the overall textual similarity score 68 for each site[.]" EX1010, 12:60-62; Lopresti ¶¶366-367.

Aridor also applies a *scraping criteria* to the site list to compute a "link topology score," based on the "anchor text" (language a pointing page uses to describe a destination page…as the search results page of the general-purpose search engine) and "anchor links" (the links that connect 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 those added to root sites within the site list, the topology score has greater significance in identifying an answer to the query (the *value for the determined token*). EX1010, 14:11-23; Lopresti ¶368.

(c) 6[d.ii.C.3]: "evaluating…"

Hartono in view of Aridor discloses this limitation. Lopresti ¶¶369-372. 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. Section X.C.5.b.iii (Ground 4, 6[d.ii.C]). Once the one or more values are extracted using the "textual score" and "link topology score," 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 ("[T]he algorithm takes into account textual affinity to the particular query and to the domain of interest, as well as topological information for finding the most 'authoritative' pages."); 14:56-15:11; Lopresti ¶371. A POSITA would understand 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 ¶372.

(d)6[d.ii.C.4]: "selecting…"

Hartono in view of Aridor discloses this limitation. Lopresti ¶¶373-375. Aridor evaluates the extracted criteria using a ranking algorithm in which the "textual score" and "link topology score" "are normalized and summed 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 ¶¶373-375.

6. 6[e]: "presenting…."

Hartono in view of Aridor discloses this limitation for the same reasons as 5[f]. Section IX.C.7. Once the search results are "downloaded to the device," Hartono would be able to present those results to the user. Lopresti ¶376.

XI. 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 '006 patent is assigned Dialect, LLC ("PO"), which is currently asserting the '006 patent in the following concurrent litigation filed on February 5, 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).

Petitioner has also filed, at substantially the same time that this Petition was filed, petitions for *inter partes* review against related family members asserted in the concurrent litigation: U.S. Patent No. 7,398,209 and U.S. Patent No. 9,734,825.

The '006 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

Petitioner provides the following designation of counsel:

Lead Counsel	Lisa K. Nguyen (Reg. No. 58,018) Paul Hastings LLP 1117 S. California Avenue Palo Alto, CA 94303 Telephone: 650.320.1800 Fax: 650.320.1900 Email: PH-META-DialectIPR@paulhastings.com

Back-Up Counsel	David Tennant (Reg. No. 48,362) Paul Hastings LLP 2050 M St., N.W. Washington, DC, 20036 Telephone: 202.551.1700 Fax: 202.551.1705 Email: PH-META-DialectIPR@paulhastings.com
Back-Up Counsel	Diane Ghrist (*pro hac vice* to be filed) Paul Hastings LLP 2050 M St., N.W. Washington, DC, 20036 Telephone: 202.551.1700 Fax: 202.551.1705 Email: PH-META-DialectIPR@paulhastings.com
Back-Up Counsel	Rachel Wu Hankinson (*pro hac vice* to be filed) 525 South Flower Street Twenty-Fifth Floor Los Angeles, CA 90071 Telephone: 213.683.6112 Fax: 213.627.0705 Email: PH-META-DialectIPR@paulhastings.com

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 ’006 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.

XII. CONCLUSION

For these reasons, Petitioner respectfully requests institution.

Dated : August 15, 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.

CLAIM LISTING APPENDIX

5. A method for processing natural language speech utterances with context-specific domain agents, comprising:

[a] receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;

[b] recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables;

[c] parsing, at a parser coupled to the processing device, information relating to the utterance to determine a meaning associated with the utterance and a context associated with the request contained in the utterance, wherein the parsed information includes the one or more recognized words or phrases;

[d] formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context, wherein formulating the request in accordance with the grammar used by the domain agent includes:

5[d.i] determining one or more required values and one or more optional values associated with formulating the request in the grammar used by the domain agent;

5[d.ii] 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, wherein the parser extracts the one or more criteria and the one or more parameters using procedures sensitive to the determined context;

5[d.iii] inferring one or more further criteria and one or more further parameters associated with the request using a dynamic set of prior probabilities or fuzzy possibilities; and

5[d.iv] transforming the one or more extracted criteria, the one or more extracted parameters, the one or more inferred criteria, and the one or more inferred parameters into one or more tokens having a format compatible with the grammar used by the domain agent, wherein the one or more tokens include all the required values and one or more of the optional values associated with formulating the request in the grammar used by the domain agent;

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

[f] presenting the generated response to the utterance via the speech unit.

6. A method for processing natural language speech utterances with context-specific domain agents, comprising:

[a] receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request, wherein the request includes at least one command and at least one question;

[b] parsing, at a parser coupled to the processing device, information relating to the utterance to determine a meaning associated with the utterance and a context associated with the request contained in the utterance;

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

[d] processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance, wherein processing the formulated request with the domain agent includes:

6[d.i] directing the at least one command to one or more local or remote processing devices to execute the at least one command; and

6[d.ii] querying one or more local or network information sources to resolve information requested in the at least one question, wherein querying the one or more local or network information sources further includes:

6[d.i.a] submitting multiple asynchronous queries to the one or more local or network information sources, wherein the multiple queries include one or more duplicate queries submitted to distinct information sources;

6[d.i.b] asynchronously receiving multiple results to the multiple asynchronous queries from the one or more local or network information sources; and

6[d.i.c] scoring a relevance associated with the multiple asynchronously received results using a dynamic set of prior probabilities or fuzzy possibilities to determine one or more best responses to the at least one question, wherein scoring the relevance associated with the multiple asynchronously received results to determine the one or more best responses includes:

6[d.i.c.i] determining one or more tokens required to formulate a response to the at least one question;

6[d.i.c.ii] applying one or more scraping criteria to the multiple asynchronously received results to extract one or more values for the asynchronously received results;

6[d.i.c.iii] evaluating the one or more extracted values using the dynamic set of prior probabilities or fuzzy possibilities to resolve ambiguous, incomplete, or conflicting information associated with the one or more extracted values; and

6[d.i.c.iv] selecting one or more best values for the one or more required tokens from the one or more evaluated values, wherein the one or more best responses include the one or more best values for the one or more required tokens; and

[e] presenting the generated response to the utterance via the speech unit, wherein the response presented via the speech unit includes results associated with the one or more local or remote processing device executing the at least one command and the one or more best responses to the at least one question.

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. 8,015,006** exclusive of the parts exempted as provided in 37 C.F.R. §42.24(a), contains 13,961 words and therefore complies with the type-volume limitations of 37 C.F.R. §42.24(a).

Dated : August 15, 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. 8,015,006 AND EXHIBITS EX1001-1026** are being served on August 15, 2025 via Federal Express overnight mail on counsel of record for U.S. Patent No. 8,015,006 as addressed below:

David A. 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:

Mark D. Siegmund (msiegmund@cjsjlaw.com)
CHERRY JOHNSON SIEGMUND JAMES, PC
Bridgeview Center
7901 Fish Pond Rd., 2nd Floor
Waco, TX 76710
Telephone: (254) 732-2242

Dated : August 15, 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.