

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

META PLATFORMS, INC.,
Petitioner,

v.

DIALECT, LLC
Patent Owner

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

**DECLARATION OF DANIEL P. LOPRESTI PH.D. IN SUPPORT OF
PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 7,398,209**

TABLE OF CONTENTS

I.	INTRODUCTION-----	1
II.	EXPERIENCE, QUALIFICATIONS, AND STATUS AS AN INDEPENDENT EXPERT-----	2
III.	MATERIALS REVIEWED AND CONSIDERED -----	12
IV.	UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES-----	12
	A. Anticipation-----	13
	B. Obviousness -----	13
	C. Priority and Written Description Priority -----	15
V.	TECHNOLOGY OVERVIEW -----	16
	A. Natural Language Speech Recognition and Spoken Dialogue Systems -----	17
	B. Asynchronous Programming -----	22
	C. Meta-searching -----	23
VI.	THE '209 PATENT -----	25
	A. Overview -----	25
	B. Prosecution History -----	26
	C. The Challenged Claims -----	27
VII.	LEVEL OF SKILL IN THE ART AND PERSPECTIVE APPLIED IN THIS DECLARATION-----	27
VIII.	OVERVIEW OF THE PRIOR ART -----	29
	A. Ross (Ex. 1005) -----	29

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 7,398,209

B.	Ittycheriah (Ex.1007) -----	31
C.	Ross018 (Ex. 1009)-----	34
D.	Hartono (Ex. 1006) -----	35
E.	Baker (Ex. 1008) -----	38
F.	Aridor (Ex. 1010) -----	43
IX.	CLAIM CONSTRUCTION -----	45
X.	OPINIONS ON GROUND 1:-----	45
A.	A POSITA Would Have Been Motivated to Combine Ross and Ittycheriah-----	45
B.	A POSITA Would have Had a Reasonable Expectation of Success -----	48
C.	Independent Claim 1 -----	50
1.	Preamble: “A method responsive to a user generated natural language speech utterance, comprising:” -----	50
2.	Limitation 1[a]: “receiving the user generated natural language speech utterance, the received user utterance containing at least one request;” -----	53
3.	Limitation 1[b]: “maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance;” -----	54
4.	Limitation 1[c]: “recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;” -----	62
5.	Limitation 1[d]: “parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request	

	contained in the utterance based on one or more keywords contained in the recognized words and phrases;” -----	64
6.	Limitation 1[e]: “selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context;” -----	68
7.	Limitation 1[f]: “formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;” -----	71
8.	Limitation 1[g]: “invoking the selected domain agent to process the formulated request; and” -----	72
9.	Limitation 1[h]: “presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.” -----	73
D.	Dependent Claim 4: “The method according to claim 1, wherein determining a context for the at least one request contained in the utterance includes:” -----	74
1.	Limitation 4[a]: “matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords, the dynamically updated set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities” -----	75
2.	Limitation 4[b]: “scoring one or more possible contexts for the matched keywords using one or more of a profile of the user, data content associated with the domain agents, or a context stack of one or more recent contexts; and” -----	77

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 7,398,209

3.	Limitation 4[c]: “determining the context by selecting one or more of the possible contexts having a score satisfying a predetermined confidence level.”-----	79
XI.	OPINIONS ON GROUND 2:-----	80
A.	A POSITA Would Have Been Motivated to Combine Ross, Ittycheriah, and Ross018-----	80
B.	A POSITA Would Have Had a Reasonable Expectation of Success -----	81
C.	Claim 6: “The method according to claim 1, wherein formulating the at least one request in accordance with the grammar used by the selected domain agents includes:” -----	83
1.	Limitation 6[a]: “determining one or more required values and one or more optional values for a request in the grammar used by the selected domain agent”-----	84
2.	Limitation 6[b]: “extracting one or more criteria and one or more parameters from the keywords contained in the recognized words and phrases using procedures sensitive to the determined context;” -----	90
3.	Limitation 6[c]: “inferring one or more criteria and one or more parameters using the dynamic set of prior probabilities or fuzzy possibilities; and”-----	92
4.	Limitation 6[d]: “transforming the extracted and the inferred criteria and parameters into a format compatible with the grammar used by the selected domain agent, the transformed criteria and parameters including all of the required values and one or more of the optional values.” -----	94
XII.	OPINIONS ON GROUND 3:-----	96
A.	A POSITA Would Have Been Motivated to Combine Ross, Ittycheriah and Aridor -----	96

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 7,398,209

B.	Reasonable Expectation of Success -----	98
C.	Claim 7: “The method according to claim 1, wherein the at least one request contained in the utterance includes at least one question and at least one command, wherein the invoked domain agent processes the formulated request by:”-----	99
1.	Limitation 7[a]: “directing the at least one command to local or remote processing devices; and” -----	99
2.	Limitation 7[b]: “querying local or network information sources to resolve information requested in the at least one question.” -----	100
D.	Claim 8: “The method according to claim 7, wherein querying the local or network information sources to resolve the information request in the at least one question includes:”-----	101
1.	Limitation 8[a]: “submitting a plurality of asynchronous queries to the information sources, the plurality of queries including a plurality of duplicate queries submitted to distinct information sources;” -----	101
2.	Limitation 8[b]: “a synchronously receiving a plurality of results of the queries from the information sources; and” -----	103
3.	Limitation 8[c]: “scoring a relevance of each of the asynchronously received results using the dynamic set of prior probabilities or fuzzy possibilities to determine one or more best responses to the at least one question”-----	104
E.	Claim 9: “The method according to claim 8, wherein determining the one or more best responses to the at least one question includes:”-----	106
1.	Limitation 9[a]: “determining one or more tokens required to formulate a response to the question;” -----	106

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 7,398,209

2.	Limitation 9[b]: “extracting one or more values for the determined tokens from the asynchronously received results, the one or more values extracted by applying one or more scraping criteria to the asynchronously received results;” -----	107
3.	Limitation 9[c]: “evaluating the extracted values using the dynamic set of prior probabilities or fuzzy possibilities, wherein evaluating the extracted values includes resolving ambiguous, incomplete, or conflicting information; and” -----	109
4.	Limitation 9[d]: “selecting one or more best values for the determined tokens from the evaluated values, wherein the one or more best responses include the one or more selected best values.” -----	110
XIII. OPINIONS ON GROUND 4:-----		111
A.	A POSITA Would Have Been Motivated to Combine Hartono and Baker -----	111
B.	A POSITA Would have Had a Reasonable Expectation of Success -----	114
C.	Independent Claim 1 -----	114
1.	Preamble: “A method responsive to a user generated natural language speech utterance, comprising:” -----	114
2.	Limitation 1[a]: “receiving the user generated natural language speech utterance, the received user utterance containing at least one request;” -----	115
3.	Limitation 1[b]: “maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance;” -----	117
4.	Limitation 1[c]: “recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;” -----	124

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 7,398,209

5.	Limitation 1[d]: “parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases;” -----	126
6.	Limitation 1[e]: “selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context;” -----	128
7.	Limitation 1[f]: “formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;” -----	131
8.	Limitation 1[g]: “invoking the selected domain agent to process the formulated request; and.” -----	132
9.	Limitation 1[h]: “presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.” -----	133
D.	Dependent Claims-----	134
1.	Claim 4: “The method according to claim 1, wherein determining a context for the at least one request contained in the utterance includes:” -----	134
2.	Claim 6: “The method according to claim 1, wherein formulating the at least one request in accordance with the grammar used by the selected domain agents includes:” -----	144
3.	Claim 7: “The method according to claim 1, wherein the at least one request contained in the utterance	

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 7,398,209

includes at least one question and at least one
command, wherein the invoked domain agent
processes the formulated request by:” ----- 152

XIV. OPINIONS ON GROUND 5:----- 155

A. A POSITA Would Have Been Motivated to Combine
Hartono, Ittycheriah, and Aridor ----- 155

B. A POSITA Would Have Had a Reasonable Expectation of
Success ----- 156

C. Claim 8: “The method according to claim 7, wherein
querying the local or network information sources to
resolve the information request in the at least one question
includes:”----- 157

1. Limitation 8[a]: “submitting a plurality of
asynchronous queries to the information sources, the
plurality of queries including a plurality of duplicate
queries submitted to distinct information sources;” ----- 157

2. Limitation 8[b]: “a synchronously receiving a
plurality of results of the queries from the
information sources; and” ----- 160

3. Limitation 8[c]: “scoring a relevance of each of the
asynchronously received results using the dynamic
set of prior probabilities or fuzzy possibilities to
determine one or more best responses to the at least
one question”----- 161

D. Claim 9: “The method according to claim 8, wherein
determining the one or more best responses to the at least
one question includes:”----- 163

1. Limitation 9[a]: “determining one or more tokens
required to formulate a response to the question;” ----- 163

2. Limitation 9[b]: “extracting one or more values for
the determined tokens from the asynchronously

received results, the one or more values extracted by
applying one or more scraping criteria to the
asynchronously received results;”----- 164

3. Limitation 9[c]: “evaluating the extracted values
using the dynamic set of prior probabilities or fuzzy
possibilities, wherein evaluating the extracted values
includes resolving ambiguous, incomplete, or
conflicting information; and” ----- 166

4. Limitation 9[d]: “selecting one or more best values
for the determined tokens from the evaluated values,
wherein the one or more best responses include the
one or more selected best values.” ----- 167

XV. CONCLUSION ----- 168

I. INTRODUCTION

1. I, Daniel Lopresti Ph.D., have been retained by Meta Platforms, Inc. (“Meta”) to provide an analysis of the scope and content of U.S. Patent No. 7,398,209 relative to the state of the art at the time of the earliest application to which the ’209 patent claims priority. My analysis relates to claims 1, 4, 6, 7, 8, and 9. I have also been retained to provide analysis regarding what a person of ordinary skill in the art (“POSITA”) would have understood at the time of the earliest application underlying the ’209 patent.

2. This declaration summarizes the opinions I have formed to date. I reserve the right to modify or amend my opinions, if necessary, based on further review and analysis of information that I receive subsequent to the filing of this declaration, including in response to positions taken by Dialect, LLC (“Dialect”) or its experts that I have not yet seen, including any secondary considerations evidence that Dialect or its expert may consider and present.

3. It is my opinion that Claims 1, 4, 6, 7, 8, and 9 of the ’209 patent are invalid based on the following grounds.

Ground 1	Claims 1 and 4 are obvious over Ross (Ex. 1005) in view of Ittycheriah (Ex. 1007)
Ground 2	Claim 6 is obvious over Ross (Ex. 1005) and Ittycheriah (Ex. 1007) in view of Ross018 (Ex. 1009)

Ground 3	Claims 7-9 are obvious over Ross (Ex. 1005) and Ittycheriah (Ex. 1007) in view of Aridor (Ex. 1010)
Ground 4	Claims 1, 4, 6, and 7 are obvious over Hartono (Ex. 1006) and Baker (Ex. 1008)
Ground 5	Claims 8-9 are obvious over Hartono (Ex. 1006) and Baker (Ex. 1008) in view of Aridor (Ex. 1010)

II. EXPERIENCE, QUALIFICATIONS, AND STATUS AS AN INDEPENDENT EXPERT

4. My curriculum vitae (“CV”) is Exhibit 1004.

5. I am being compensated at the rate of \$500 per hour for my work on this declaration. My fee is not contingent on the outcome of any matter or on any of the technical positions I explain in this declaration.

6. I have no financial interest in Meta. I have been informed that Dialect claims ownership of the ’209 Patent. I have no financial interest in Dialect.

7. I am currently a tenured Full Professor in the Department of Computer Science & Engineering (“CSE”) of the P.C. Rossin College of Engineering and Applied Science at Lehigh University in Bethlehem, Pennsylvania and have been since May 2009.

8. I have been a faculty member in the CSE Department since 2003. From July 2009 through June 2019, I also served as Chair of the CSE Department, except for July 2014 to June 2015, during which I served as the Interim Dean of the P.C.

Rossin College of Engineering and Applied Science at Lehigh University. In July 2015, I was named Director of the Lehigh Data X strategic initiative focused on interdisciplinary data science, a role I held until June 2020. From 2003 to 2009, I was an Associate Professor in the CSE Department.

9. I have taught, conducted research, and worked on various aspects of computer science and engineering, including aspects of computer science and engineering related to speech recognition, such as search algorithms used for speech recognition, parallel algorithms, speech synthesis, speech biometrics, and other areas.

10. I received a B.A. in Mathematics and Engineering Science from Dartmouth College in 1982, an M.S.E. in Electrical Engineering and Computer Science from Princeton University in 1983, an M.A. in Electrical Engineering and Computer Science from Princeton University in 1984, and a Ph.D. in Computer Science from Princeton University in 1987.

11. The courses that I have taught at Lehigh include Big Data Analytics, Artificial Intelligence Theory and Practice, AI for Social Good, Software Product Management, Issues in Cybersecurity, Data Mining, Research Methods, Operating System Design, and Breadth of Computing. In my Big Data Analytics, AI, and Data Mining courses I have covered a range of techniques from artificial intelligence and machine learning. The Breadth of Computing course was an introductory course

which covered the operation of computer hardware, software, networks, and systems. Topics covered in this course included introductions to designing websites, writing computer programs, computer architecture, operating systems, computer networks, databases, security, privacy, and social, ethical, and professional issues. As a part of this class, I taught the students HTML and JavaScript programming.

12. From December 2002 to June 2003, I held a courtesy appointment as a Visiting Research Scientist at the Palo Alto Research Center, and from February 2003 to June 2003 I also held courtesy appointments as a Research Fellow in the Information Security Institute at Johns Hopkins University, and as a Visiting Research Associate Professor at Rensselaer Polytechnic Institute.

13. From October 1997 to December 2002, I was a Member of Technical Staff at Bell Labs/Lucent Technologies, conducting research in the areas of computer security, document image analysis, optical character recognition, and pen computing, and where I also collaborated with colleagues on the development of text-to-speech (“TTS”) technologies. I also worked on audio CAPTCHAs, speech biometrics, and handwriting biometrics.

14. From June 1991 to October 1997, I was a Principal Scientist at the Matsushita Information Technology Laboratory in Princeton, NJ, conducting

research in the areas of optical character recognition, pen computing, document analysis, and document processing systems.

15. From September 1986 to May 1991, I was an Assistant Professor of Computer Science at Brown University, conducting research in parallel computing and computer-aided design of VLSI. During this time I also taught courses related to VLSI design, parallel computing, discrete math, and computational molecular biology.

16. From 2009 through 2013, I served as Chair of Technical Committee 11 of the International Association for Pattern Recognition (“IAPR”). Technical Committee 11 is known as the “Reading Systems” committee and its goal is to understand and develop systems that can analyze media containing character symbols and return an encoded representation of the text content and structure.

17. From 2018 through 2021, I served as Treasurer of IAPR, and in 2021, I was elected President of IAPR. In 2022, I transitioned to the role of Past-President and held that position until 2024. I served as a member of the IAPR Executive Committee from 2018 through 2024.

18. From 2015 through 2025, I served as an appointed member on the Council for the Computing Community Consortium (“CCC”) of the Computing Research Association (“CRA”). I assumed the role of Vice Chair of CCC effective July 2020 and became Chair of CCC in July 2022. I served in the position of Chair

Emeritus of CCC from July 2024 through June 2025 and remain involved in some ongoing CCC and CRA activities. The role of the CCC is to catalyze the United States computing research community to pursue high-impact research, conduct activities that strengthen the research community, articulate compelling research visions, and align those visions with key national and global challenges.

19. While on the CCC Council, I served on, and also co-chaired, a number of Task Forces, including: Intelligent Infrastructure; Cybersecurity and Cybercrime; Technology for a Connected and Distributed Life; Security, Integrity, and Trust; NextGen AI; and Grand Challenges in Computing. I also served on the Roadmap for Artificial Intelligence Research Steering Committee, which addressed processing of speech and dialog systems, among other AI topics. *A 20-Year Community Roadmap for Artificial Intelligence Research in the United States* (2019), available online at <https://cra.org/ccc/wp-content/uploads/sites/2/2019/08/Community-Roadmap-for-AI-Research.pdf>. I also helped organize the recent CCC Workshop on the Future of Information Retrieval Research in the Age of Generative AI, which was held in Washington, DC in July 2024.

20. Since 2014, I have served as an Editor-in-Chief of the International Journal of Document Analysis and Recognition which is a scholarly journal published by Springer-Nature and generally concerns all areas related to document analysis and recognition. This includes research dealing with computer recognition

of characters, symbols, text, lines, graphics, images, handwriting, signatures, as well as automatic analyses of the overall physical and logical structures of documents, with the ultimate objective of a high-level understanding of their semantic content. Over a period of more than 20 years, I have served as a technical program co-chair for many of the major international conferences in the fields of document analysis and pattern recognition research.

21. Since 2011, I have also served as an Area Editor for Computer Vision and Image Understanding which is a scholarly journal published by Elsevier. The focus of Computer Vision and Image Understanding is the computer analysis of pictorial information, including all aspects of image analysis from the low-level, iconic processes of early vision to the high-level, symbolic processes of recognition and interpretation.

22. My areas of expertise encompass the fields of document analysis techniques which combine both computer vision and natural language understanding, optical character recognition technology, other artificial intelligence techniques (e.g., machine learning and computer vision), computer and biometric security techniques, electronic voting systems, and research reproducibility.

23. I have authored over 175 publications in the computer science field, including publications relating to the algorithms that are central to the computer processing of speech, including the design of custom computer hardware for parallel

algorithms, speech synthesis, speech in security applications, and speech biometrics.

I am a named inventor on 24 United States patents. I have authored and co-authored many publications involving search algorithms and dynamic programming, text-to-speech, computer processing of speech, Hidden Markov Models, and related concepts. My Ph.D. thesis was titled “Discounts for dynamic programming with applications in VLSI processor arrays” (1987), available at <https://www.proquest.com/docview/303617084?pq-origsite=gscholar&fromopenview=true>, which dealt with both the dynamic programming algorithms that are core to speech recognition, and also the parallel implementation of such algorithms.

24. While they are fully set forth in my CV, some examples of my publications include:

- (1) “A Text-to-Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions,” M. Lee, D. Lopresti, and J. Olive, *Proceedings of the Fourth ISCA Workshop on Speech Synthesis*, August-September 2001, Perthshire, Scotland, pp. 75-80.
- (2) “Human Interactive Proofs for Spoken Language Interfaces,” D. Lopresti, C. Shih and G. Kochanski, *Proceedings of the Workshop on Human Interactive Proofs*, January 2002, Palo Alto, CA, pp. 30-34.

- (3) "Towards Speech-Generated Cryptographic Keys on Resource-Constrained Devices," F. Monroe, M. Reiter, Q. Li, D. Lopresti, and C. Shih, *Proceedings of the Eleventh USENIX Security Symposium*, August 2002, San Francisco, CA, pp. 283-296.
- (4) "A Reverse Turing Test Using Speech," G. Kochanski, D. Lopresti, and C. Shih, *Proceedings of the Seventh International Conference on Spoken Language Processing*, September 2002, Denver, CO, pp. 1357-1360.
- (5) "A Text-to-Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions," M. Lee, D. Lopresti, and J. Olive, *International Journal of Speech Technology*, vol. 6, no. 4, October 2003, pp. 347-356.
- (6) "Speech Biometric Mapping for Cryptographic Key Generation," K. Inthavisas and D. Lopresti, *Biometric Technology for Human Identification VIII (SPIE Defense, Security, and Sensing)*, April 2011, Orlando, FL, pp. 80291P-1 - 80291P-12.
- (7) "Attacks on Speech Biometric Authentication," K. Inthavisas and D. Lopresti, *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICCV 2011)*, July 2011, Las Vegas, NV, pp. 310-316.

- (8) "Speech Cryptographic Key Regeneration Based on Password," K. Inthavisas and D. Lopresti, Proceedings of the International Joint Conference on Biometrics (IJCB 2011), October 2011, Arlington, VA, 7 pages, DOI: 10.1109/IJCB.2011.6117553.
- (9) "Secure Speech Biometric Templates for User Authentication," K. Inthavisas and D. Lopresti, *IET Biometrics*, vol. 1, no. 1, pp. 46-54, 2012, DOI: 10.1049/iet-bmt.2011.0008.
- (10) I am a named inventor in 24 United States patents, including:
- (11) "Methods and Apparatus for Defending Against Telephone-based Robotic Attacks Using Random Personal Codes," H. Baird, J. Bentley, D. Lopresti, and S.Y. Wang, U.S. Patent No. 7,978,831, issued July 2011.
- (12) "Methods and Apparatus for Defending Against Telephone-based Robotic Attacks Using Contextual-based Degradation," H. Baird, J. Bentley, D. Lopresti, and S.Y. Wang, U.S. Patent No. 8,005,197, issued August 2011.
- (13) "Methods and Apparatus for Defending Against Telephone-based Robotic Attacks Using Permutation of an IVR Menu," H. Baird, J. Bentley, D. Lopresti, and S.Y. Wang, U.S. Patent No. 8,005,198, issued August 2011.

25. I have also been a member of the Institute of Electrical and Electronics Engineers (“IEEE”) since 1982. I became an IEEE Senior Member in 2003, and an IEEE Life Member in 2024. During my career, I have authored or co-authored approximately 18 refereed papers or proceedings that were published in IEEE technical publications.

26. I have also been active in other IEEE activities. For example, I served as program co-chair for the Eleventh International Conference on Document Analysis and Recognition in 2011, whose proceedings were published on the IEEE Explore website. I have received two awards from IEEE organizations. I was an Associate Editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence* between 2005 and 2010. I have been a member of conference program committees for several IEEE conferences, I have served as an invited reviewer for a number of different IEEE journals and several IEEE conferences, all as listed in my CV.

27. I am also a member of several other scientific and research associations, and have conducted various activities for them, attended their conferences, and published papers and proceedings in their technical publications. These include the Association for Computing Machinery (“ACM”) and the Association for the Advancement of Artificial Intelligence (“AAAI”).

28. In formulating my opinions, I have relied upon my training, knowledge, and experience in the relevant art.

III. MATERIALS REVIEWED AND CONSIDERED

29. My opinions in this declaration are based on working in the field of computer science and engineering, particularly on topics relating to natural language speech recognition, including my teaching, research, and consulting. I have an established understanding of the relevant field at the relevant time frame, and I have an understanding of the skill set, capabilities, and knowledge of a POSITA as of June 2002.

30. My opinions are also based on investigation and study of the patent at issue, its file history, and the prior art. In the course of forming my opinions, I have reviewed the exhibits submitted with the Petition, which are specifically identified in Appendix A.

31. I may rely upon these materials and/or additional materials to rebut arguments raised by Dialect. Further, I may also consider additional documents and information in formulating any necessary opinions including documents that may not yet have been provided to me.

IV. UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES

understand that statutory and judicially created standards must be considered to determine the validity of a patent claim. I am not an attorney and,

consequently, will offer no opinion on the law itself. My understanding of the pertinent law is described in this section and is the result of explanations provided by counsel. I have applied this understanding in my analysis.

33. I understand that a patent claim is unpatentable if it is obvious in view of the prior art. I further understand that the frame of reference for determining whether a claim is obvious is from the perspective of a POSITA at the time of invention. I have been asked to assume that the effective priority date is June 3, 2002. I have not been asked to determine whether this date is appropriate.

A. Anticipation

34. I have been informed that for a patent claim to be anticipated by the prior art, each and every limitation of the claim must be found, expressly or inherently, in a single prior art reference as recited in the claim. I have been informed that a claim limitation not expressly found in a prior art references is inherent if the prior art necessarily functions in accordance with, or includes, the claim limitation. Mere probability that a limitation is included is not sufficient to establish inherency.

B. Obviousness

35. In analyzing obviousness in light of the prior art I have been informed that it is important to understand the scope of the claims, the level of skill in the relevant art, the scope and content of the prior art, the differences between the prior

art and the claims, and any objective indicia of non-obviousness (also called secondary considerations).

36. I have been informed that a patent claim is unpatentable for obviousness if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA to which said subject matter pertains. I have been informed that obviousness may be based on one reference or a combination of references. I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.

37. I have been informed that when a patented invention is a combination of known elements, the Board must determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue by considering the teachings of prior art references, the effects of demands known to people working in a field or present in the marketplace, and the background knowledge possessed by a POSITA.

38. I have been informed that the Supreme Court has recognized several rationales for combining references or modifying a reference to show obviousness of the claimed subject matter. I understand that several of these rationales are: (1) combining prior art elements according to known methods to yield predictable

results; (2) simple substitution of one known element for another to obtain predictable results; (3) use of a known technique to improve a similar device (method or product) in the same way; (4) applying a known technique to a known device (method or product) ready for improvement to yield predictable results; (5) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; (6) and some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

39. I have been informed that in order to prove that a claimed invention is not patentable for obviousness, a petitioner must (1) identify the differences between the claim and particular disclosures in the prior art references, singly or in combination, (2) specifically explain how the prior art references could have been combined in order to arrive at the subject matter of the claimed invention, and (3) specifically explain why a POSITA would have had reasons to so combine the prior art references.

C. Priority and Written Description Priority

40. I have been informed that to claim priority to a prior application, each application in the chain that leads to an earlier application must comply with the written description requirement. I have been informed that in order to satisfy the written description requirement, the specification of the provisional application must

contain a written description of the invention and the manner and process of making and using it, in such full, clear, concise, and exact terms, that it would enable a person of ordinary skill in the art to practice the invention claimed in the non-provisional application. I have been informed that the provisional application must describe the invention claimed in the non-provisional application such that a person of skill in the art can clearly conclude that the inventor had invented the claimed invention by the provisional application filing date.

V. TECHNOLOGY OVERVIEW

41. In this section I provide overviews of technologies and prior art pertinent to the '209 patent and to the state of the art as of 2002. In later sections, I address in detail how the prior art discloses and renders obvious the recitations of the challenged claims.

42. The prior art references described below and discussions of what was known to a person of ordinary skill in the art provide a general description of the state of the art at the time of the invention, provide additional motivation to modify the primary references evaluated in the grounds of invalidity with the knowledge of a person of ordinary skill in the art or other references cited herein, and provide support for why a person of ordinary skill in the art would have had a reasonable expectation of success in combining the teachings of the references. The additional

prior art references identified in this section are exemplary in nature and are included to provide factual support for my opinions.

A. Natural Language Speech Recognition and Spoken Dialogue Systems

43. Natural language speech recognition refers to the process in which a computer system converts spoken language into written text. This enables interaction and communication between human and computers through speech and voice. Natural language speech recognition methods have been used for decades. By the mid-1990s, several companies including IBM and Dragon Systems, had already been selling speech recognition systems off-the-shelf as ways to process user-generated natural language speech into machine readable text to be used by various applications. *See generally* Ex. 1017 (IBM ViaVoice 1998 User Manual), Ex. 1018 (IBM ViaVoice 2002 User Manual), Ex. 1023 (Dragon Systems Naturally Speaking 5 User's Guide).

44. These speech recognition systems used statistical probabilities, language modeling, and context-recognition procedures to recognize speech. By 2002, one of the most common acoustic models for speech recognition that used probabilities was a concept referred to as the hidden Markov model ("HMM"). Ex. 1019, 22. As Federick Jelenick, a well-known researcher in the field of speech recognition wrote, "[a] *Markov chain* is a very important and well-studied concept of probability theory. It deals with a class of random processes that incorporate a

minimum amount of memory without actually being memoryless.” Ex. 1019, 22. Essentially, HMMs apply probabilities to recognize words based on how they transition between one acoustic data point to the next and the probability of a specific transition.

45. By 2001, IBM’s ViaVoice product was already incorporated into an IBM “Personal Speech Assistant,” which was a “complete speech recognition, speech synthesis, and dialog management system.” Ex. 1021, 1, 3. It employed HHM states modeled by a Gaussian mixture to look at the likelihood between different transitions to recognize phrases. Ex. 1021, 3. Using these statistical probabilities, a user could speak into the Personal Speech Assistant, which would be received by a dialog manager. Ex. 1021, 4. After speech recognition is performed, the dialogue manager would convert the utterance into events using a vocabulary stored in a database. Ex. 1021, 2, 4. The dialog manager would then send the “associated event and data” to “the target application.” Ex. 1021, 4. For example, natural language utterance around appointments could be sent to a speech-enabled date book application, such that queries or commands for new appointments “can be transcribed directly into the calendar as text[.]” Ex. 1021, 4.

46. Importantly, they also routinely used dictionaries and phrase tables as reference vocabularies to identify words. As Frederick Jelenick also noted, “[e]very

speech recognizer is based upon a vocabulary of finite size.” Ex. 1019, 62. These “vocabularies” contain lists of words and phrases, that are often specific to a speaker:

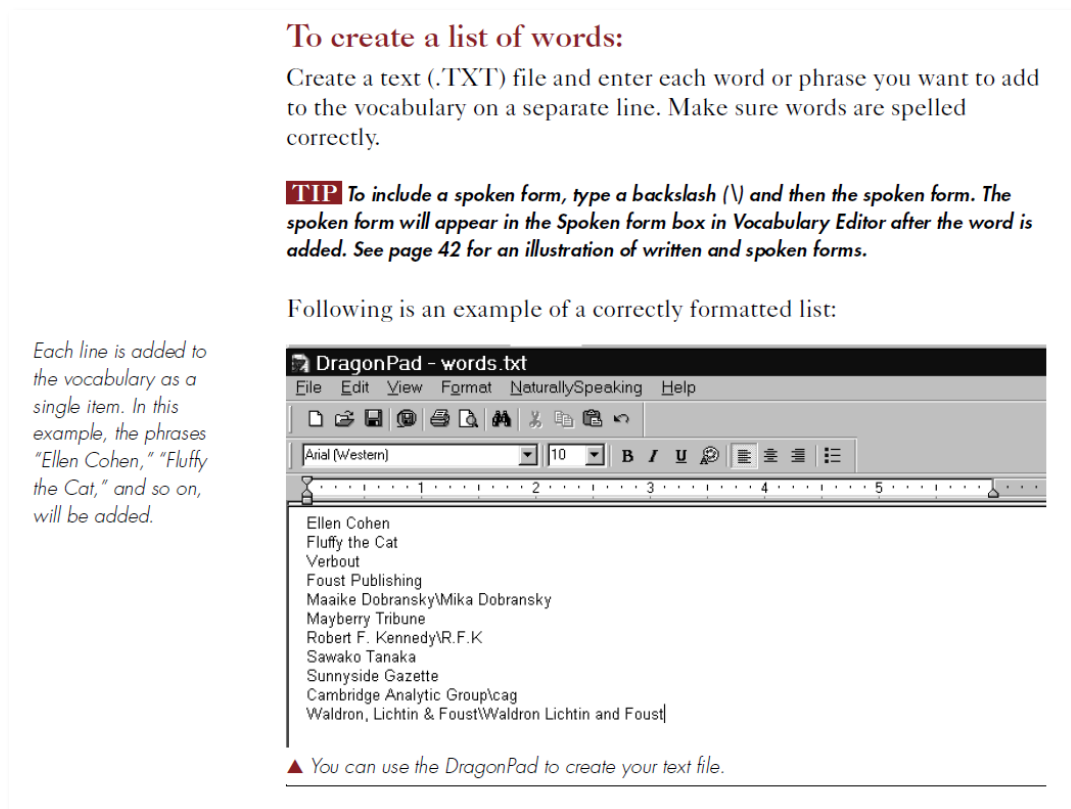
A vocabulary is particular to a speaker and his task in at least four different aspects. First, choice of words reflects the speaker’s habits of expression that are related, for example, to his level of education. His usage is also conditioned by the general domain of discourse (e.g., data processing, musicology, medical reports, etc.) that calls for a variety of technical terms, clichés, and such. Furthermore, the speaker has available those names, addresses, and other expressions peculiar to his current interests. Finally, he requires words specific to the topic of the particular document he is composing.

A large, fixed vocabulary can at most provide for the speaker’s habitual usage and for the discourse domain, but a completely satisfactory coverage can be attained only through a user’s active, though limited, participation. In particular, although he can be asked to indicate whether a proper noun is a personal or company name, location, etc., and even to annotate technical terms, he cannot be expected to provide any grammatical feature classification of ordinary English words. Information about the latter should either be contained in some large stored source lexicon, or, better still, be extracted from the actual syntactic/semantic function that these words play in the previously transcribed text.

Ex. 1019, 63. These vocabularies were often dynamic in nature, as a way to personalize the vocabulary to the speaker. *Id.* (“A crude method of personalization could be devised by maintaining during dictation a dynamically varying vocabulary consisting at any given moment of the last L different words used. The resulting *dynamic coverage* is then defined as the asymptotic probability that the word spoken next already belongs to the vocabulary.”). In other words, a vocabulary built from the speakers most recently used words would be the most probable words the speaker

would use again in follow-on spoken inputs. Ex. 1023, at 47-65 (describing the dynamic vocabulary used by Dragon Systems that is customized by a user either manually or through adding previously recognized words to a vocabulary).

47. In the below example, it is shown how a vocabulary consists of both individual words and phrases:



Ex. 1023, 57.

48. By 2002, there were several different types of spoken dialogue systems that allowed users to interact with computer-based applications, including agent-based systems. See Ex. 1020, 3. Agent-based systems in particular were designed to allow for more complex communications between the system, user, and

underlying application. Such systems would often use various different knowledge sources to process spoken dialogue including a dialogue history between the system and a domain model, with specific information about the domain. Ex. 1020, 6. An agent-based system is a broad concept, but it generally includes interactions between two agents, “each of which is capable of reasoning about its own actions and beliefs, and sometimes also about the actions and beliefs of the other agent.” Ex. 1020, 5.

49. For example, an application itself can act as an agent if it interacts with another system agent—like conversation or dialogue manager—to respond to a user’s spoken utterance. Such systems can be seen in U.S. Patent No. 7,127,402, in which computer programs themselves are “agents”:

In the approach of the present invention, the speech center system 20 has an explicit model of the world (e.g., domain model 70) which will serve as a foundation for language understanding and reasoning. Some of the basic concepts that the speech center system 20 models using the domain model 70 are:		10
		15
Things	A basic category that includes all others	
Agents	Animate objects, people, organizations, computer programs	
Objects	Inanimate objects, including documents and their sub-objects	
Locations	Places in the world, within the computer, the network, and within documents	20
Time	Includes dates, as well as time of day.	
Actions	Things that agents can do to alter the state of the world	
Attributes	Characteristics of things, such as color, author, etc.	
Events	An action that has occurred, will occur, or is occurring over a span of time.	25

Ex. 1012, 7:8-25

Ex. 1012, 8:12-16 (“The speech center 20 has a rudimentary built-in notion of what an ‘action’ is. An ‘action’ is something that an agent can do in order to achieve some change in the state of the word (e.g., known to the speech center 20 and an application 26).”).

B. Asynchronous Programming

50. In the field of computing (in particular, software engineering), it is generally assumed that when there are multiple applications present in a system they will behave asynchronously. This means that each application will execute at its own rate, independent of the execution of other applications running on the same system. Ex. 1024, 3 (“asynchronous” defined as “[P]ertaining to, being, or characteristic of something that is not dependent on timing. For example, asynchronous communications can start and stop at any time instead of having to match the timing governed by a clock” (Microsoft Computer Dictionary 4th Ed. (1999))).

51. The alternative – synchronous execution – actually requires substantially more program code and effort to implement, as the applications must be designed to communicate with each other at various phases during their execution to stay in sync, and if one application gets too far ahead of the other it will be paused (“blocked”) until the other application has had a chance to catch up. Such

coordination can be complex and potentially error-prone, so it is avoided when it is not necessary for the correct operation of the applications. We can tell that applications execute asynchronously when there is no indication of, or need for, step-by-step coordination between the applications. An application that simply returns its final result when it is done is still operating asynchronously.

C. Meta-searching

52. In its early days, the Internet experienced explosive growth. It quickly became impossible for a user to benefit from the totality of content available online that was relevant to a question they might have or a task they wanted to perform. More and more search engines were developed to help users navigate an increasingly complex information landscape. To anyone who tried the various alternatives available for search, it was apparent that different search engines would produce different results, even for an identical query. No single search engine dominated all the others: depending on a user's specific information needs, one search engine might be more effective than its competitors, but it would not always be the same one. While it was possible to conduct a search by manually invoking several engines and browsing through the results from each, this clearly placed a significant extra burden on the user.

53. The concept of meta-search was created to automate the process of querying multiple search engines and integrating their results in a way that produced

more a comprehensive response to a user's information needs. Using a meta-search engine, a user asks just one query and the engine determines which other search engines to engage. Because the secondary search engines are stand-alone systems fielded by different, competing vendors, they run autonomously. The meta-search engine receives the results from those searches and integrates them before returning a response to the user. Going beyond simply reporting all of the hits returned, a meta-search engine could combine and reformulate the results, recognizing that information repeatedly returned by different search engines might be more authoritative and valuable to the user. If one search engine misses an important resource – it has a “blind spot” – that will not hurt the user because other search engines are likely to know about that resource. A meta-search engine could also employ the output from one search engine to refine its query to another, allowing it to investigate a topic more thoroughly. In all these ways, a meta-search engine acts as an agent working on behalf of the user to seek out and aggregate as much information as possible for the benefit of the user.

54. In the case of a speech interface, meta-search offers even more benefits because it can produce a more comprehensive response to a query while requiring less back-and-forth interaction between the user and the system.

VI. THE '209 PATENT

A. Overview

55. I understand that the '209 Patent claims priority to a provisional application U.S. Provisional Application No. 60/384,388 filed on June 3, 2002.

56. The '209 patent describes well-known methods for natural language processing of speech that allows users to use natural language to process a request. The '209 Patent recognizes that natural language processing and speech recognition was well known by June 3, 2002. Ex. 1001, 1:38-43. The '209 Patent provides for an “integrated approach” to natural language processing that “makes maximum use of context, prior information, domain knowledge, and user specific profile data to achieve a natural environment for one or more users making queries or commands in multiple domains,” by essentially combining the admittedly well-known systems and methods. [[*Id.*, 1:65-2:1. 9:57-61]].

57. Specifically, the '209 Patent describes high-level steps to process natural language speech. Ex. 1001 , Figs. 4A, 4B. The '209 Patent details that upon receipt of a natural language “utterance,” the words and phrases contained in the “utterance” are “recognized,” by using “one or more dictionary and phrase tables.” Ex. 1001 , 24:27-32. Once recognized, using a “parser,” the words and phrases are used “to determine a meaning” of the utterance using “criteria and/or parameters of the properly query and/or command.” Ex. 1001, 24:50-25:1, Fig 6. The '209 Patent

then describes selecting a “domain agent” based on the determined meaning, i.e., “an autonomous executable that receives, processes, and responds to requests associated with the determined context.” Ex. 1001 , cl. 1. The utterance “request” is then formulated “in accordance with a grammar used by the selected domain agent,” and the domain agent is then able to process the request. *Id.*; Ex. 1001, 25:29-43. The results of the processed request are then presented to the user. Ex. 1001, cl. 1.

B. Prosecution History

58. I understand that the original claims of the '209 Patent were repeatedly rejected over the prior art. In particular, I understand that the Examiner suggested more specific subject matter “including fuzzy probabilities and specific methods of variable substitutions, which might not be as common as what the broader claims were directed to, but made no representation that the subject matter would be allowable.” Ex. 1002, 211-212.

59. I understand that the Applicant cancelled all remaining claim and presented new claims 44-57 after an Examiner interview. Ex. 1002, 211-212, 230-230-242.

60. I understand that on May 21, 2008, the Examiner explained that “the prior art of record does not disclose or reasonably suggest recognizing words using information from phrase tables in combination with the limitations of parsing to determine a meaning based on keywords, selecting a domain agent, and formulating

a request in accordance with a grammar used by a selected domain agent.” Ex. 1002, 341-344.

61. I understand that the Examiner found that U.S. Patent No. 6,523,061 to Halverson *et al.*, “omits a grammar used by a domain agent associated with the determined context and one or more dictionary and phrase tables.” Ex 1002, 342.

62. I understand that the examiner stated that “[w]hile it is known to recognize words based on a vocabulary definition by a dictionary for speech recognition, the prior art of record does not disclose or reasonably suggest additionally using phrase tables for speech recognition.” Ex. 1002, 342.

C. The Challenged Claims

63. The '209 Patent has 14 claims, and I understand that only six of these claims, claims 1, 4, 6, 7, 8, and 9, are challenged. The challenged claims generally cover a method of processing natural language.

VII. LEVEL OF SKILL IN THE ART AND PERSPECTIVE APPLIED IN THIS DECLARATION

64. I understand that certain issues relating to validity must be judged from the perspective of a person of ordinary skill in the relevant art, as I discussed below.

65. In determining the level of ordinary skill, I have been asked to consider, for example, the types of problems encountered in the field, prior solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the educational level of active workers in the field. In my opinion,

it would have taken at least two years of experience in the field for a person to become familiar with the problems encountered in the field and the prior and current solutions to those problems.

66. Taking those factors into consideration, person of ordinary skill in the art in June 2002 (which I assume is the earliest possible priority date of the '209 patent) 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.

67. My opinions do not turn on the precise definition, and the challenged claim would be unpatentable from the perspective of any reasonable person of ordinary skill in the art.

68. I believe that I would have qualified as a person of at least ordinary skill in the art as of the earliest claimed date of June 3, 2002, which appears on the face of the '209 patent. I believe that I have a sufficient level of knowledge, experience, and education to provide an expert opinion in the field of the '209 patent.

69. For the purpose of this Declaration, and unless otherwise noted, my statements and opinions, such as those regarding my experience and understanding

of a person of ordinary skill in the art generally (and specifically related to the references I consulted) reflect the knowledge that existed in the field as of June 3, 2002. This is true even if stated in the present tense.

VIII. OVERVIEW OF THE PRIOR ART

A. Ross (Ex. 1005)

70. Ross describes a system that includes a “speech center” that “coordinates speech services for a number of speech-enabled applications performing on a computer system,” including the “task of deciding the target application for a particular utterance by a user.” Ex. 1005, Abstract, 2:3-9, Fig. 1.

71. When the system receives a speech input from the user, Ross uses a speech engine to convert spoken commands (i.e., requests) “into text strings that can be utilized by software applications,” i.e., a “speech representation.” Ex. 1005, 6:62-7:2. Ross recognizes that by 2001, speech engines were already well known and commercially available including ViaVoice™ from IBM and NaturallySpeaking Professional from Dragon Systems.

72. 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.” Ex. 1005, 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:

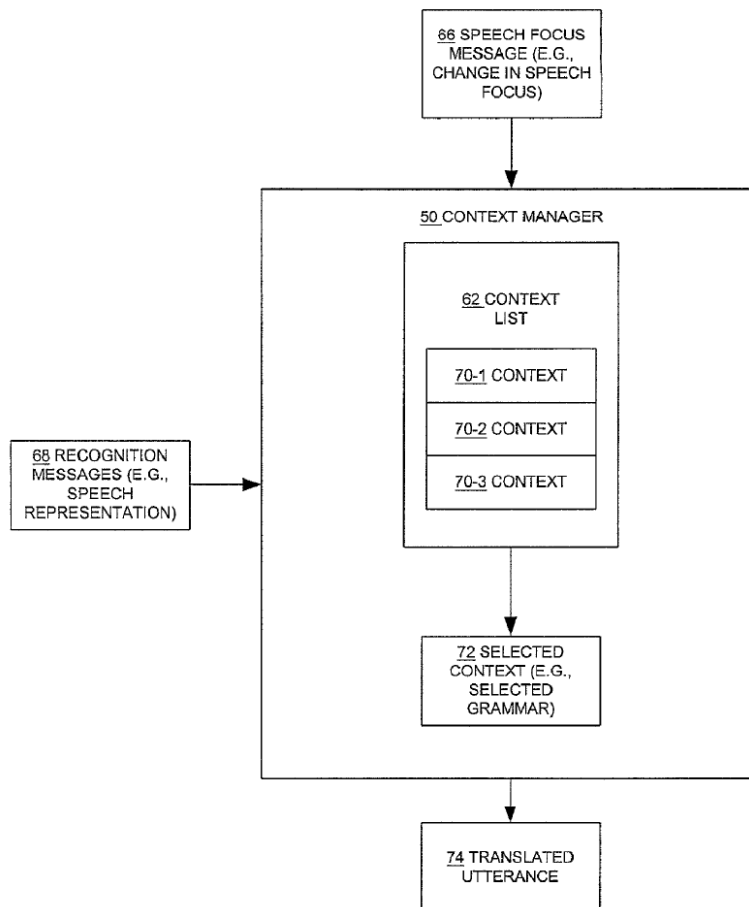


FIG. 4

Ex. 1005, Fig. 4, 2:47-49, 3:1-3, 6:47-61, 8:46-9:13.

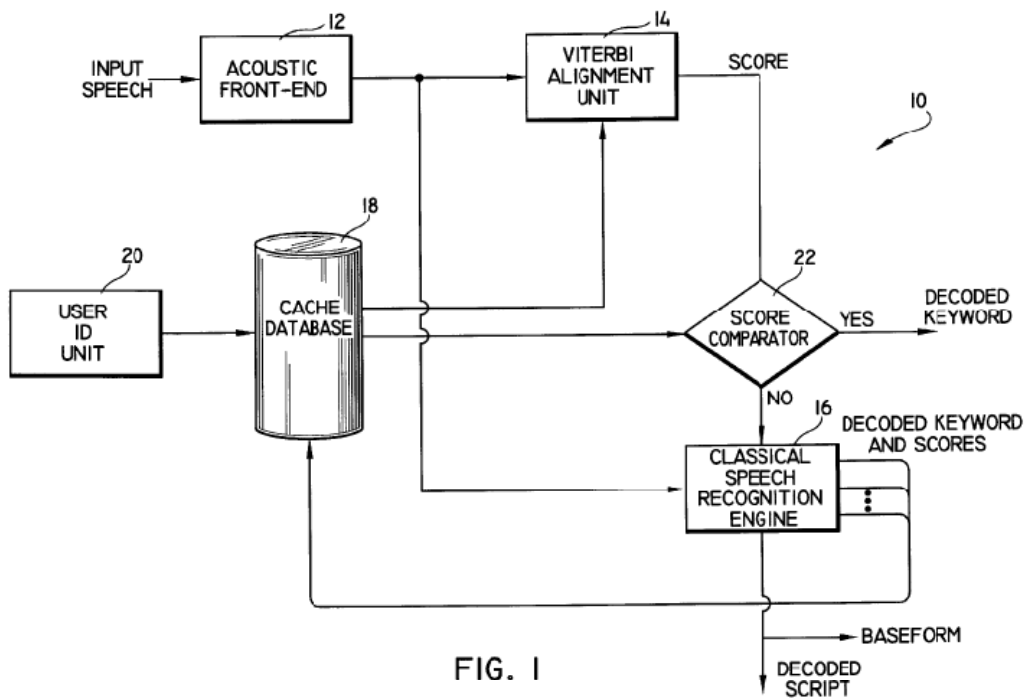
73. In this way, Ross maintains a set of prior probabilities of possible contexts “biased to send commands to the most recently accessed application,” that is continuously being “reordered.” Ex. 1005, 8:47-53; *see also* 5:58-61.

74. Ross then 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 based on this matching (i.e., relevance scoring). Ex. 1005, 7:23-26.

75. 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. Ex. 1005, 7:27-28.

B. Ittycheriah (Ex.1007)

76. Ittycheriah discloses the details of the *speech engine*. Ittycheriah discloses a speech recognition engine for “performing speech recognition on speech segments frequently input by a user,” as shown below in Figure 1:



Ex. 1007, Fig. 1, Abstract, 1:11-15, 1:51-54

77. 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[.]” Ex. 1007, 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.” Ex. 1007, 2:63-66.

78. Ittycheriah uses a set of prior probabilities and possibilities to perform speech recognition by way of Hidden Markov Models (HMMs), explaining that once speech is input into its system:

[T]he acoustic features extracted in acoustic front-end 12 are subject to probabilistic matching technique using Hidden Markov models

(HMMs) in both the alignment unit 14 and the engine 16... It is to be understood that Hidden Markov modeling provides different scores for different hypotheses. Examples of such scores are rank (i.e., the ranking of a decoded word), fast match scores (i.e., likelihoods of a simplified scheme), detailed match scores (i.e., the likelihoods of a detailed scheme), language modeling scores (i.e., probability according to a language model) and/or any combination of these scores.

Ex. 1007, 3:56-4:4.

79. Ittycheriah's compares the "resulting scores" to "scores which were previously stored in the cache database 18 pertaining to the particular user are presented to the score comparator 22 wherein the scores from unit 14 are compared to the stored scores." Ex. 1007, 4:33-43. "If the new scores are within a margin of, the same as, or larger than the stored scores corresponding to one of the decoded keywords in the cache, then the keyword is accepted," and the decoded keyword in the cache is outputted from the score comparator. Ex. 1007, 4:43-48.

80. If the new scores are "not within a margin of error, or are less than any of the stored scores, then the decoded keywords in the cache are rejected and the input speech segment is subjected to the complete vocabulary of the classical speech recognition engine," from which a decoded script (i.e., a keyword) is outputted. Ex. 1007, 4:48-53. The decoded keyword and associated scores from the classical

speech engine (and its vocabulary) are then added to the cache database. Ex. 1007, 4:54-57.

81. Ittycheriah's vocabulary in its "classical speech engine" contains a "larger scale language database." Ex. 1007, 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." Ex. 1007, 1:55-58, 2:16-19, 2:47-49.

82. A person of ordinary skill in the art would thus 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 '209 patent.

C. Ross018 (Ex. 1009)

83. Ross018 discloses the details of transforming the natural language utterance into a grammar of a given external application or domain. Ross018 builds on the teaching of Ross, by detailing a "semantics analysis module" and "syntax manager," that use a domain model to process spoken utterances "into a set of frames (i.e., internal representation of the spoken utterance)." Ex. 1009, Abstract.

84. As Ross018's domain model consists of classes that "define commonly used concepts such as people, organizations, objects, actions, events, time, location,

and number, along with base grammatical constructs that define ways that these items can be combined into sentences.” Ex. 1009, 2:13-17.

85. After a spoken utterance is recognized, Ross018’s semantics analysis module processes the utterance “using a grammatic specification 90 based on the domain model 70 to produce an initial semantic representation based on the recognized utterance. Ex. 1009, 11:61-65. Ross018 accomplishes this by producing a “frame structure based on the recognized spoken utterance” that is used to “provide[] a set of propositions.” Ex. 1009, 12:4-10. These “propositions” are composed of attribute-object-value triples, that include criteria, parameters, and values associated with the spoken utterance. Ex. 1009, 10:5-30.

86. Using these propositions, Ross018 generates a “goal” that is then further used to generate a response, including “an action to be executed, such as a script call or command to an external application.” Ex. 1009, 12:11-20.

D. Hartono (Ex. 1006)

87. Hartono discloses an agent-based system of processing natural language utterances. Hartono discloses a “method for receiving commands, executing received commands and adaptively interacting with the user using a natural language interface.” Ex. 1006, 1:20-23.

88. To that end, 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. Ex. 1006, 4:20-25.

89. Hartono illustrates its use of various application agents in Figure 1, below:

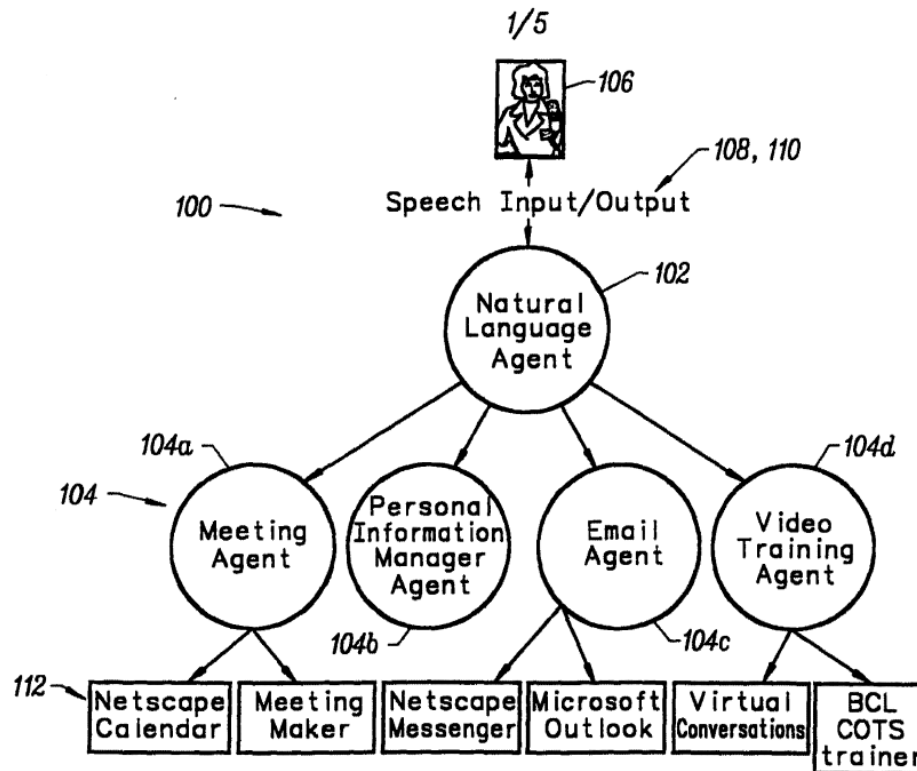


FIG. 1

Ex. 1006, Fig. 1, 7:6-9

90. Hartono initially processes input speech using “[a]ny suitable off the shelf speech recognition system.” Ex. 1006, 11:4-12.

91. 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.” Ex. 1006, 12:11-13:10.

92. Hartono then 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

Ex. 1006, Table I

93. To complete this mapping, Hartono’s semantic interpretation engine uses a “reverse grammar generation mechanism” that “includes a list or vector for each word and corresponding probabilities for each word in the list.” Ex. 1006, 14:10-20.

94. 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. Ex. 1006, 15:4-12.

95. Using both the action-concept (verb) and topic-concepts (noun/object), Hartono determines the context of the natural language request by “determin[ing] which task agent should handle the request.” Ex.1006, 15:13-19.

96. Hartono communicates with various task agents (also called “application class agents”), which “works with a single class of applications...that have similar conceptual operations.” Ex. 1006, 23:15-23, 28:1-3 (“Each task agent thus serves as the common user interface for the corresponding class of applications under the assumption that each application within the class accomplishes the same or generally overlapping set of tasks.”).

97. Based on the determine 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.” Ex. 1006, 24:14-25:3.

98. The command within the natural language request is then executed by the selected application. Ex. 1006, 4:20-25, 5:4-8, 5:11-15, 29:21-22, 42:11-12.

E. Baker (Ex. 1008)

99. I understand that Baker is a U.S. Patent entitled “Systems and Methods for Word Recognition” filed on June 7, 1995 and issued on October 21, 1997.

100. In my opinion, Baker describes systems and methods for “detect[ing] words within portions of speech signals and compar[ing] these detected words to

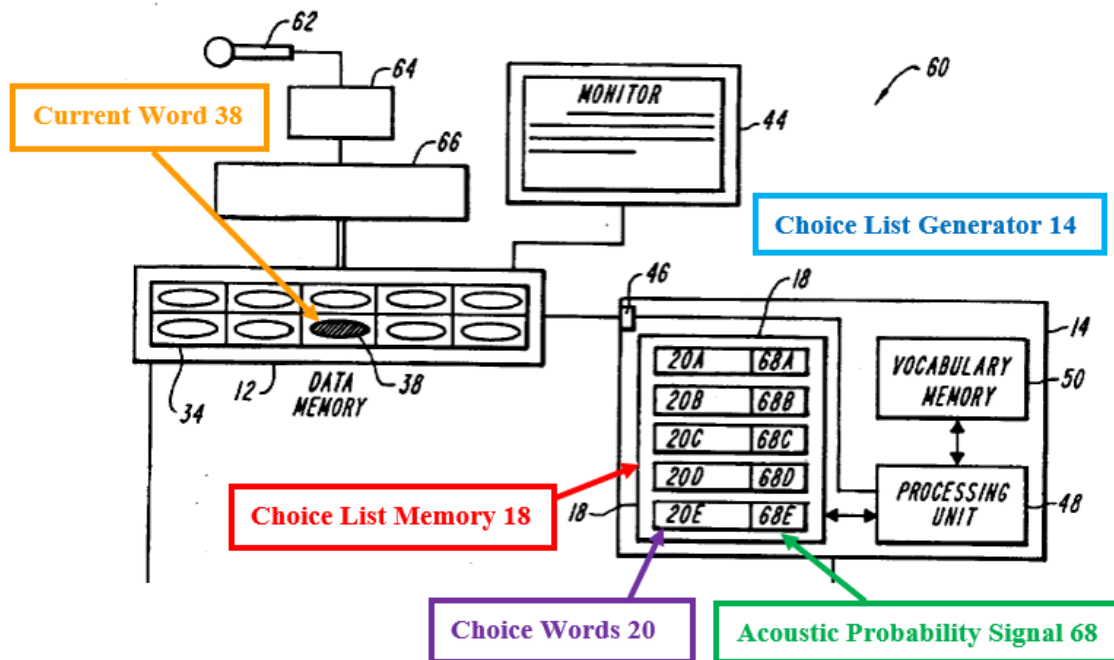
acoustic models of known words, or sequence of acoustic models of known words.”

Ex. 1008, 18:13-16.

101. Upon detection of a “speech signal,” Baker compares the detected speech to “acoustic models of known words, or sequence of acoustic models of known words.” Ex. 1008, 18:12-16.

102. “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.” Ex. 1008, 18:16-19. Words with a probability signal indicating a “high degree of certainty” are deemed recognized (referred to as “context words”). Ex. 1008, 18:16-19. However, words with a “low degree of certainty” indicate that the word has not been confidently identified (referred to as the “current word”). Ex. 1008, 18:22-24.

103. For those “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:



Ex. 1008, Fig. 2 (excerpted and annotated)

104. Baker explains that each “choice word 20A-E” has a corresponding “acoustic probability signals 68A-E” that represent “the degree of certainty that the word 20 associated with the acoustic model is substantially representative of the current word 38 being recognized.” Ex. 1008, 18:30-35, 20:30-46 (“The probability signals 68A-68E can represent the likelihood that the associated word choice 20A-20E is substantially representative of the current word 38 being recognized.”), 22:55-64.

105. Baker also discloses generating a “word list” of words that “have a semantic association with the choice word 20.” Ex. 1008, 14:57-69. The “semantic association” for each choice word (which Baker also refers to as “passage words”

(labeled as 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.” Ex. 1008, 20:55-58 (“The word list generator 16 can access via a transmission path the reference database 40 and can identify information in the reference database 40 that relate to the topic, theme, or subject of a choice word 20.”).

106. Baker next “perform[s] a matching function to calculate the number of matched words” between those word recognized by the speech analysis unit 66 with high confidence (the “context words”) with the words in the “word list” (the “passage words”). Ex. 1008, 21:19-47. By looking at the “relative frequency of matches between the “context words” and the “passage words,”

107. Baker calculates a “probability signal 28 representative of the likelihood that the choice word or phrase 20 substantially represents the utterance currently being recognized.” Ex. 1008, 21:19-47, 21:49-22:6, Figs, 2-4.

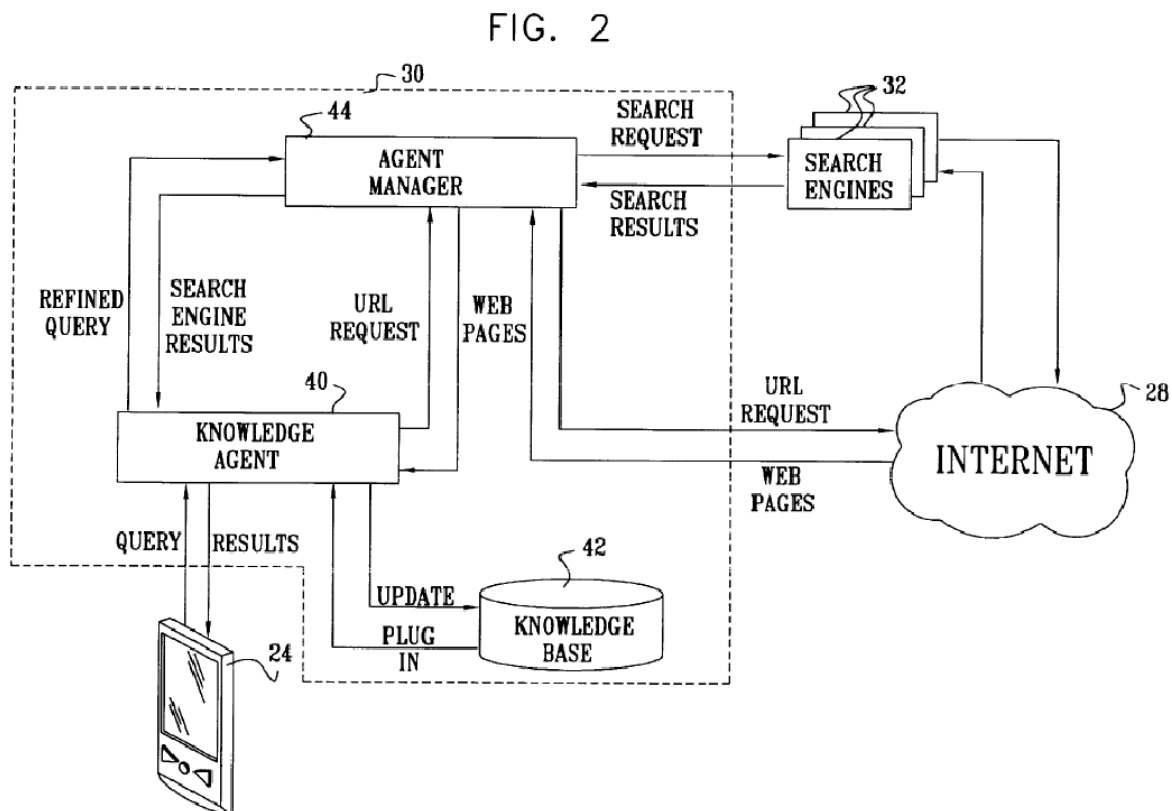
108. Baker then generates a “ranking signal 30” that “represents the likelihood...that a choice word 20 is substantially similar to” the word being recognized” as a function of both the “context matching and acoustic model matching” probabilities. Ex. 1008, 23:22-27. Baker selects the choice word with the highest rank signal and supplants the unrecognized word in the text string with

the choice word as a way of recognizing the received utterance. Ex. 1008, 12:54-62, 17:23-27 (“[T]he word recognition system 10 can select the choice word 20 with the highest probability signal 28 or rank signal 30 to substitute for the current word 38 currently being recognized.”).

109. Baker recognizes that by 1997 it was conventional for natural language speech recognition systems to “rely on data intensive language models to identify” words, and these models are essentially statistical probability models that include a set of hypotheses as to what words are likely to appear in association with other words.” Ex. 1008, 2:1-12, 2:36-47 (discussing the n-gram language model to identify similar sequences of words), 17:11-15 (“However, it will be apparent to one of ordinary skill in the art of word recognition that many known techniques can be employed to generate the probability signal 28. Any of these methods, or variations thereof, can be used without departing from the scope of the present invention.”), 21:65-22:6 (“Techniques for generating the probability signal given two constraints...are well known in the field of mathematics and any of the techniques capable of generating a probability signal given a set of constraints, including the maximum entropy technique, can be practiced with the present invention without departing from the scope thereof.”).

F. Aridor (Ex. 1010)

110. Aridor describes methods for searching “a corpus of documents,” including documents found on the Internet by performing searches over multiple search engines, and filtering and ranking the retrieved responses. Figure 2 of Aridor shows the general structure of its system:



Ex. 1010, Abstract, Fig. 2

111. Aridor uses “knowledge agents” to perform these searches, each of which “specializes in a domain.” Ex. 1010, 3:38-52. Aridor states that “[t]he 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.” Ex. 1010, 8:23-3. Aridor also states:

In terms of user interaction, the knowledge agent acts as an intermediary between the user and one or more Web search engines, preferably managing the entire search process for the user. For each search, the user chooses the knowledge agent that has the relevant specialization, typically a knowledge agent that the user has initiated and used in previous searches.

Ex. 1011, 3:53-59. Aridor’s knowledge agents also “make use of persistent, acquired knowledge in a domain that is defined and redefined by a user.” Ex. 1011, 3:65-4:2, 14:44-15:19.

112. 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.” Ex. 1010, 8:45-59, 9:6-14, 11:19-23. Aridor summarizes how it ranks and filters the results as follows, using both textual and topological information scraped from the retrieved results:

The knowledge agent analyzes the initial search results and then retrieves additional pages pointing to and from these pages according to their relevance to the query and to the domain of interest. The knowledge agent applies a ranking algorithm to this expanded set of pages. Preferably, the 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.

Ex. 1011, 4:16-24, 11:19-34 (summarizing scoring based on textual and topological aspects), 11:35-12:67 (textual similarity scoring), 13:1-14:23 (topological scoring), Figs. 4-5.

IX. CLAIM CONSTRUCTION

113. I have applied the plain meaning of the terms and phrases in the claims in my analysis, as a POSITA would have understood those terms and phrases at the relevant time (around 2002). I understand that the Board will construe the claims using the same claim construction standard that would be used in district courts. My declaration will establish that the prior art meets each of the claim limitations under any reasonable construction. I have not been asked to consider any express constructions.

X. OPINIONS ON GROUND 1:

114. In my opinion, claims 1 and 4 are rendered obvious by Ross (Ex. 1005) in view of Ittycheriah (Ex. 1007).

A. A POSITA Would Have Been Motivated to Combine Ross and Ittycheriah

115. In my opinion, a person of ordinary skill in the art would have been motivated implement the details of the Ross “speech engine” using Ittycheriah’s speech recognition method because Ross teaches its speech engine could use any

off-the-shelf speech recognition system, including those available from IBM, like the Ittycheriah system.

116. Ross discloses that its “speech center,” shown below as item 20 in Figure 2, is distinct from the “speech engine,” which is used to “convert speech into text strings that can be utilized by software applications on a computer.” Ex. 1005, 1:7-22.

117. Although Ross does not disclose the details of the “speech engine,” Ross teaches that off-the-shelf speech engines could be used to accomplish this step. For example, Ross teaches that “[i]n a preferred embodiment, the speech engine 22 is ViaVoice™ from IBM®.” Ex. 1005, 4:37-38; *see also* 1:16-21 (“Examples of speech recognition products that convert speech into text strings that can be utilized by software applications on a computer system include the ViaVoice™ from IBM®, Armonk, N.Y., and NaturallySpeaking Professional from Dragon Systems, Newton, Mass.”). Indeed, the provisional patent 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.”

118. Accordingly, a person of ordinary skill in the art would have been motivated to look for and rely on available speech recognition methods to use as Ross’s “speech engine,” including Ittycheriah’s speech recognition engine which was also developed by IBM.

119. A person of ordinary skill in the art would have also been motivated to implement the Ross speech engine using Ittycheriah as a way to quickly complete the speech recognition process by way of its cache database of recently used keywords and probability scores. At the time, there were multiple design choices to improve the accuracy and speed of speech recognition, and utilizing recently used keywords and probability scores were among those known design choices. These features would have advanced the objectives of Ross, which includes an objective that “avoids the time, expense, and footprint of duplicating speech services.” Ex. 1005, 2:14-16.

120. In addition, a person of ordinary skill in the art would have been motivated to use Ittycheriah’s speech recognition system as Ross’s speech engine because Ittycheriah is able to complete its speech recognition faster than classical speech engines by way of its cache database that stores recently-recognized keywords and associated probabilities. *See supra*, Section VIII.B (Ittycheriah Overview).

121. Because it is desirable for speech-enabled requests and command (like those inputted into Ross) to be processed quickly, a person of ordinary skill in the art would have found it desirable to use Ittycheriah’s cache database to “reduce the amount of hypotheses to consider in the decoding process.” Ex. 1007, 2:13-19, 2:58-63 (“Generally, the present invention provides for storing data in a cache and

utilizing such data in order to quickly accept or reject keywords, i.e., names, commands or sentences, without impacting the overall recognition process and therefore, without adversely affecting recognition performance.”).

B. A POSITA Would have Had a Reasonable Expectation of Success

122. In my opinion, a person of ordinary skill in the art would have had a reasonable expectation of success in combining the teachings of Ross and Ittycheriah.

123. Ross teaches that its speech engine 22, as shown below in Figure 2, could be any commercially-available speech recognition engine. Accordingly, a person of ordinary skill in the art would have used the software algorithms of Ittycheriah’s speech recognition process to implement the details of speech engine 22:

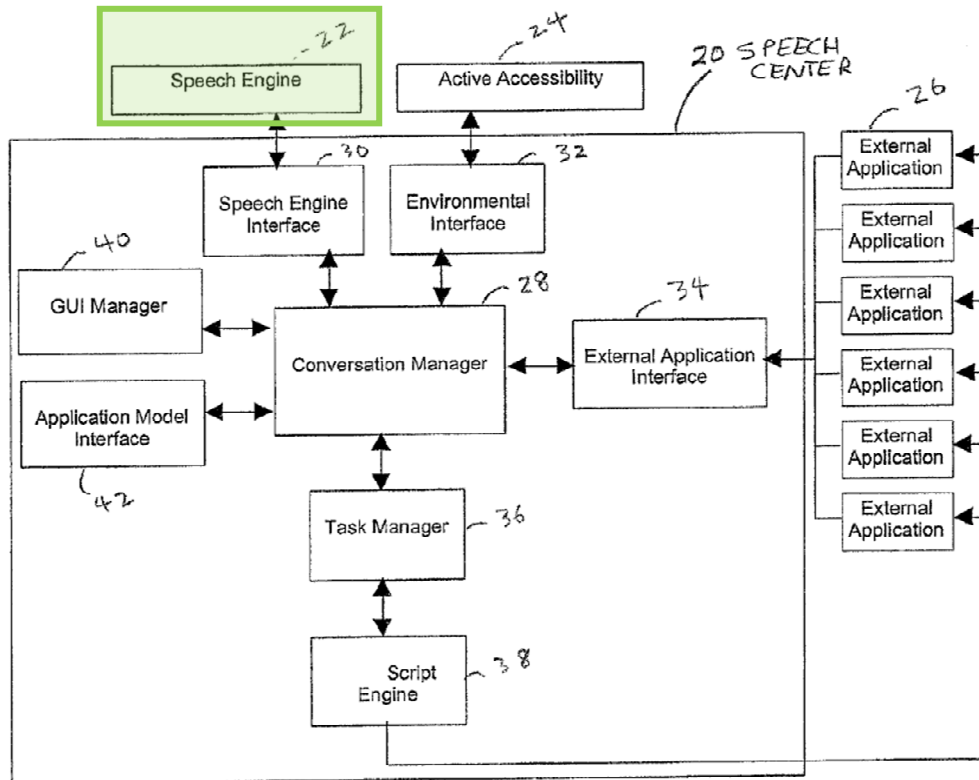


Fig. 2

Ex. 1005, Fig. 2 (annotated)

This is particularly true since both Ross and Ittycheriah were developed by IBM.

124. A POSITA would have also understood that Ittycheriah's speech recognition system would be compatible with Ross, as Ittycheriah discloses that the system's decoded keywords and scripts could be used to direct speech requests to external speech-enabled applications to perform their respective automatic functions, just like in Ross. Ex. 1007, 4:63-5:3 ("It is to be appreciated that the output of the comparator 22 (decoded keyword) and the output of the engine 16

(decoded script) may be connected to the actual phone system or navigation system [speech recognition applications] such that the phone or navigation systems are responsive to the decoded keyword or script and thus perform the respective automatic functions (i.e., dialing or navigating).”).

125. A person of ordinary skill in the art 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 (e.g., known systems for speech recognition and processing natural language requests) according to known methods (e.g., known methods of 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 adding knowledge of a user’s domain, plans, tasks, and processes in parsing the user’s natural language utterance.

C. Independent Claim 1

126. In my opinion, Ross discloses the Independent Claim.

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

127. It is my opinion that Ross discloses the preamble to the extent limiting

128. As I discussed above in Section VIII.A (Ross Overview), Ross discloses a method for “determining a speech-enabled application to receive a

spoken utterance in a multi-context speech enabled environment.” Ex. 1005, 2:43-46, Abstract.

129. To be more specific, Ross discloses a “speech center” that is responsive to user generated natural language speech via a graphical user interface. Ex. 1005, 5:34-38. Ross discusses several examples of its methods responding to natural language requests, for example, “[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.” Ex. 1005, 8:25-33.

130. Additionally, Ross describes that when “the user asks whether the user has any new messages,” he or she “receives an answer supplied by a message or mail application.” Ex. 1005, 2:25-31

In another example, the user asks what is the current stock price for a particular company and obtains the stock price from a stock monitoring service. Furthermore, the user asks whether the user has any new messages and receives an answer supplied by a message or mail application. Applications do not even have to be running (i.e., currently performing on a computer system). For example, the user makes a request to take a letter, and a word processing application starts up (i.e., is launched by the speech center) if it is not already running.

Ex. 1005, 2:25-36

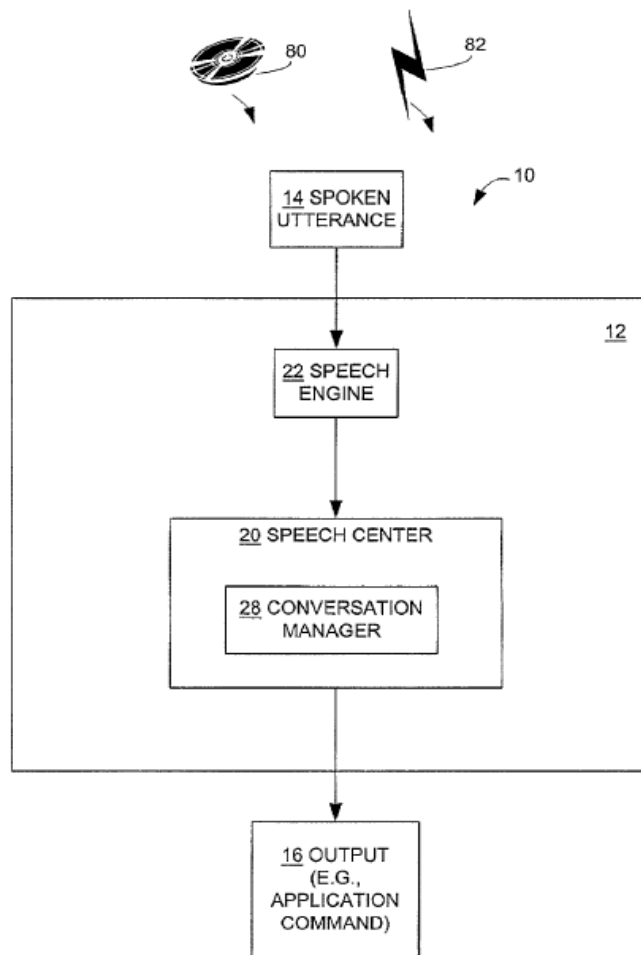


FIG. 1

Ross, Fig. 1

Ex. 1005, 3:41-53.

131. The method that Ross discloses is responsive to the user-generated natural language speech because it produced an output 16, which is “a command or other input based on the spoken utterance 14 directed to the speech enabled external application 26[.]” Ex. 1005, 3:41-53. The external application then presents to the user an answer to the spoken utterance. Ex. 1005, 2:22-36.

132. Thus, it is my opinion that a person of ordinary skill in the art would have understood that Ross discloses a method for “determining a speech-enabled application to receive a spoken utterance in a multi-context speech enabled environment.” [[Ex. 1005, 2:43-46, Abstract.]].

2. Limitation 1[a]: “receiving the user generated natural language speech utterance, the received user utterance containing at least one request;”

133. In my opinion, Ross renders obvious this limitation.

134. As I discussed for the preamble, Section X.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*. Ex. 1005, 3:45-48.

135. A person of ordinary skill in the art would have understood and agreed that a “voice command” is *at least one request*. Ex. 1005, 3:45-48, 9:40-43 (“[T]he spoken utterance including a command that is able to be associated with multiple applications, any subset of which may or may not be running[.]”); *see* Ex. 1005, 2:22-36 (describing various utterances such asking about the user’s “next appointment” or “current stock price for a particular company,” each containing “at least one request”).

136. Thus, it is my opinion that a person of ordinary skill in the art would have understood that Ross discloses *receiving the user generated natural language speech utterance*. Ex. 1005, 3:45-48.

3. Limitation 1[b]: “maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance;”

137. In my opinion, Ross in view of Ittycheriah renders obvious this limitation.

138. While the '209 Patent is not clear which of its claimed steps are considered a “stage of processing the received user utterance,” the specification discusses the use of “prior probabilities” and “fuzzy possibilities” as it relates to “recognizing step” 1[c] (Ex. 1001, 17:9-21) and “parsing step” 1[d] (Ex. 1001, 24:63-25:9).

139. In my opinion, a person of ordinary skill in the art would not have considered any of the other claimed steps related to a “stage of processing the received user utterance,” but steps associated with selecting and invoking a speech-enabled application. As such, I will address this limitation as it relates to limitation 1[c] and 1[d] here. However, to the extent prior probabilities or fuzzy possibilities are required for other limitations, I will address those separately in the section for the respective limitations. It is my opinion that a person of ordinary skill in the art

would have understood that those prior probabilities or fuzzy possibilities would likewise be maintained as part of the proposed combinations.

140. In my opinion, a person of ordinary skill in the art would have found that the combination of Ross with Ittycheriah renders the “maintaining” step 1[b] applies to the “recognizing” step 1[c] obvious.

141. As discussed above as to motivation to combine (Section X.A), Ross discloses well-known “speech recognition products” that could “convert speech into text strings,” such as the ViaVoice™ from IBM® and NaturallySpeaking Professional Dragon Systems. Ex. 1005, 1:17-22; *see also* Ex. 1017, 1019, 1023. An example of a speech recognition product can be found in Ittycheriah which also happens to have been developed by IBM® as well. Ex. 1007, 1:10-15 (“The present invention relates to speech recognition systems.”); *id.* at cover (assignee International Business Machines Corporation). *See also* Ex. 1017, 1019.

142. Ittycheriah, also developed by IBM, discloses methods for “performing speech recognition on speech segments frequently input by a user.” Ex. 1007, Abstract. As part of Ittycheriah’s speech recognition method, Ittycheriah discloses “storing data in a cache database,” in which the cache “contains the word history of a keyword (name, command or sentence) as well as the likelihood (score) obtained when the word was added to the cache.” Ex 1007, 2:58-66.

143. Ittycheriah explains that the “likelihood score” is created as part of the “Viterbi alignment unit,” using a “probabilistic pattern matching technique using Hidden Markov Models (HMMs)” which “provid[es] different scores for different hypotheses.” Ex. 1007, 3:54-60.

144. Ittycheriah provides different examples of how those “likelihood scores” are defined using prior probabilities:

It is to be understood that Hidden Markov modeling provides different scores for different hypotheses. Examples of such scores are rank (i.e., the ranking of a decoded word), fast match Scores (i.e., the likelihoods of a simplified Scheme), detailed match Scores (i.e., the likelihoods of a detailed Scheme), language modeling Scores (i.e., probability according to a language model) and/or any combination of these Scores.

Ex. 1007, 3:64-4:4

145. A person of ordinary skill in the art would have recognized that by June 2002, HMMs were well-known and ubiquitous in the field of speech recognition. *See* Ex. 1019 (“A *Markov chain* is a very important and well-studied concept of probability theory. It deals with a class of random processes that incorporate a minimum amount of memory without actually being memoryless.”).

146. The Hidden Markov Model uses probability distributions to determine the maximum likely sequence of transitions between a starting state of an input and the probability of a given output. *See, e.g., id.* at 17-25 (describing how transitions

sequences based on probability distributions are calculated by HMMs). *See* Section VIII.B (Overview of Ittycheriah).

147. Therefore, it would have been well known to a person of ordinary skill in the art that Ittycheriah's cache database that stores "likelihood scores" for keywords *maintains a set of prior probabilities or fuzzy possibilities* that is *usable* to recognize words or phrases in the received utterance.

148. A person of ordinary skill in the art would have understood Ittycheriah's cache database, specifically the set of prior probabilities that the database maintains, to be *dynamic*.

149. Ittycheriah describes its cache database as being created from and constantly updated by way of the classical speech recognition engine, labeled as item 16 below in Figure 1:

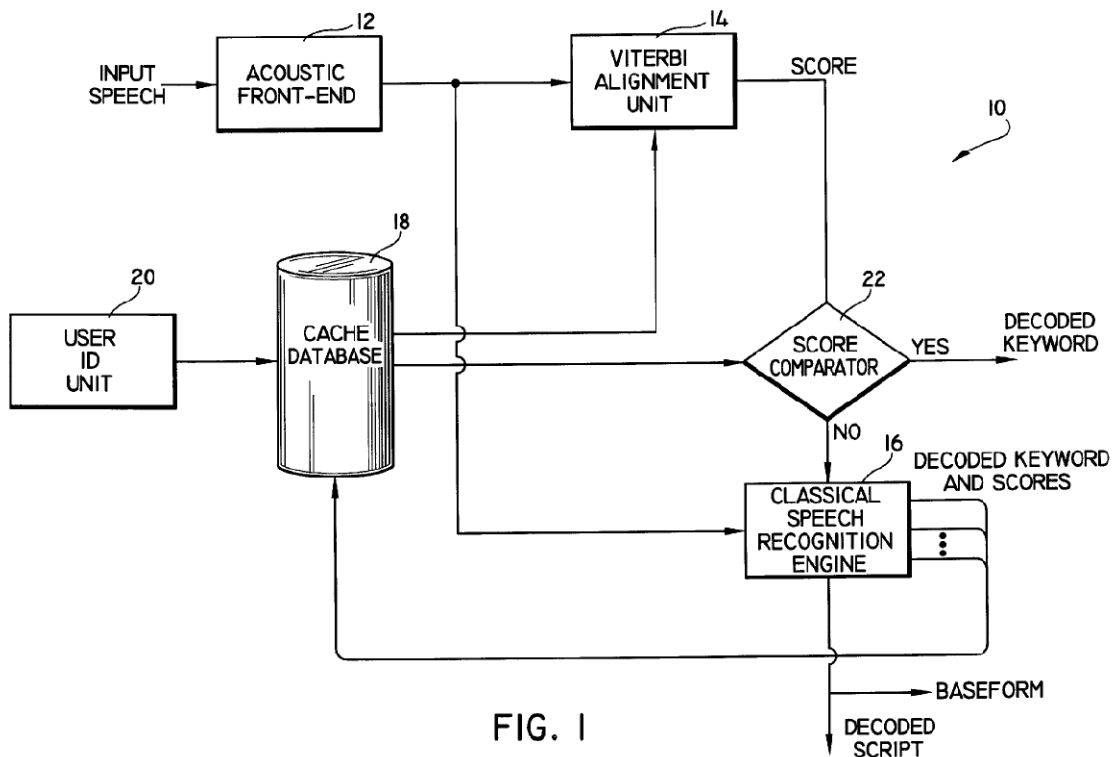


FIG. 1

Ex. 1007, Fig. 1

150. As explained in Ittycheriah:

[W]hen the same user enters another speech segment, the following operation is performed. First, each decoded keyword in the cache database 18 corresponding to the same user is considered as a hypothesis with respect to the newly entered Speech Segment. This evaluation is done by the Viterbi alignment unit 14. The resulting scores from such comparison are then presented to the Score comparator 22. Also, the Scores which were previously Stored in the cache database 18 pertaining to the particular user are presented to the Score comparator 22 wherein the Scores from unit 14 are compared to the Stored Scores. If the new scores are not within a margin or, or are less than any of the

stored scores [in the cache database], then the decoded keywords in the cache are rejected and the input speech segment is subjected to the complete vocabulary of the classical speech recognition engine 16 whereby a decoded script is generated.

Ex. 1007, 4:33-53.

151. Ittycheriah then goes on to explain how the cache database receives and processes updates, (“Subsequently, the decoded word(s) resulting from the recognition performed by the engine is then preferably added to the cache along with associated scores.”); 8:3-5 (claim 22): “After the input speech is decoded by the engine 16, the decoded word and associated scores are provided to the cache database 18 where they are stored as new keyword and related scores for that particular user. In this way the cache stores the latest and most frequently used keywords that a user may input into the system.” Ex. 1007, 4:54-59, 3:16-18.

152. A person of ordinary skill in the art would have understood that due to Ittycheriah’s cache database constantly being updated with “the latest and most frequently used keywords that a user may input,” the cache database would be *dynamic*.

153. In my opinion, a person of ordinary skill in the art would have found that the combination of Ross with Ittycheriah renders the “maintaining” step 1[b] applies to the “parsing” step 1[d] obvious.

154. As I explained above in detail in Section VIII.B. (Overview, Ittycheriah) and further below with respect to 1[c], Ittycheriah’s “cache database” uses a dynamic set of HMM derived probabilities to create a “likelihood score” during the initial “recognizing” step. And in response, Ittycheriah outputs a decoded keyword that Ross’s conversation manager then uses to determine a meaning (including determining a context) of the utterance.

155. A person of ordinary skill in the art would have understood that the combination of Ross and Ittycheriah maintains the same set of prior probabilities for step 1[c] as it does for step 1[d].

156. Moreover, Ross alone discloses this limitation in an additional way. Ross *maintains* an “ordered list of applications (e.g., a context list 62),” which is *a set of prior probabilities* that “may prioritize the contexts based on the access characteristic.” Ross, 3:1-2; *id.* 2:47-49. More specifically, the Ross system uses a “conversation manager” that “combines an analysis of the utterance with information on the state of the desktop and remembered context from previous recognitions to determine the intended target of the utterance.” Ross, 5:50-60. This would have been a well-known technique at the time to a person of ordinary skill in the art.

157. For example, as shown below, Ross uses “recency of relevant access to the context” (i.e., a relevance score) as an “access characteristic” such that Ross “is biased to send commands to the most recently accessed application”

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

Ex. 1005, 8:46-53

158. As I discussed previously, Ross’s context list is also *dynamic* as it is constantly being reordered “when (1) “an application **26** gains window focus, it will move to the head of the list 62” and/or (2) “an application **26** which is not the top priority application **26** is chosen as the target for a speech command, the application **26** indicated by the selected context **72** will move to the head of the list **62**.” Ex. 1005, 6:53-61; 7:9-15, 34-35.

159. Thus, it is my opinion that a POSITA would have understood that Ross in view of Ittycheriah discloses *maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance.*

4. Limitation 1[c]: “recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;”

160. In my opinion, Ross in view of Ittycheriah renders obvious this limitation.

161. As I explained previously in Sections X.A (Motivation to Combine) and X.C.3 (limitation 1[b]), 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.” Ross, 6:12-15, 7:12.

162. Ittycheriah recognizes words and phrases contained in a received utterance using “keywords” stored in a “cache database” that are “frequently used by a user, as well as score information attributable to the keywords.” Ittycheriah, 1:55-58. Ittycheriah defines the term “keyword” to “include such *words and phrases* as names, commands, and sentences which include both functional and non-functional words.” Ittycheriah, 2:16-19.

163. A person of ordinary skill in the art would have understood that Ittycheriah 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.” Ittycheriah, 3:62-64, 4:48-53.

164. By 2002, a person of ordinary skill in the art would have understood that Ittycheriah's cache database and large scale language database of "words and phrases" that include "names, commands, and sentences" is both a dictionary and phrases table (Ex. 1026 (defining "dictionary" as "a list of words stored in machine-readable form for reference"); Ex. 1025 (defining "dictionary" as "a list (as of phrases, synonyms, or hyphenated instructions) stored machine-readable form); Ex. 1019, 73 ("Every speech recognizer is based upon a vocabulary...that includes only the words that can appear in the transcribed text.")).

165. As an additional example, IBM's ViaVoice™ product defined the term "vocabulary" as used in its speech recognition product to be "a list of words provided with the product or obtained separately which can be recognized correctly when spoken by a user." Ex. 1017 (ViaVoice User Manual 2002). The ViaVoice™ product also allowed users to include phrases, rather than individual words. Ex. 1017, 58 (ViaVoice™ User Manual 2002).

166. A person of ordinary skill in the art would have understood that Ittycheriah's classical speech recognition engine contains a usable "dictionary and phrase tables," as claimed in the '209 patent. A person of ordinary skill in the art would have understood that the cache database and large-scale language database as used in Ittycheriah was well known by 2002.

167. Thus, it is my opinion that a POSITA would have understood that Ross in view of Ittycheriah discloses *recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables.*

5. Limitation 1[d]: “parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases;”

168. In my opinion, Ross in view of Ittycheriah renders obvious this limitation.

169. As I discussed previously in Section X.C.3, the output of Ittycheriah’s speech recognition engine is at least one “decoded keyword...representative of the input speech segment.” Ittycheriah 2:5-7. It stands to reason that this “decoded keyword” contains *one or more keywords contained in the recognized words and phrases*, including “names, commands, and sentences which include both functional and non-functional words.” Ittycheriah, 2:16-19.

170. A person of ordinary skill in the art would have understood Ittycheriah’s “decoded keyword” to corresponds to the “recognition message” in Ross, which is output by the speech engine because it is a “speech representation recognized by the speech engine 22.” Ross, 7:1-2; Ittycheriah, 4:63-53 (“It is to be appreciated that the output of the comparator 22 (decoded keyword) and the output

of the engine 16 (decoded script) may be connected to [external applications] such that the [external applications] are responsive to the decoded keyword or script and thus perform their respective automatic functions.”).

171. A person of ordinary skill in the art would have thus found it obvious to use Ittycheriah’s “decoded keyword” as a recognition message in Ross’s context manager to *determine a meaning of the utterance by determining a context for the at least one request contained in the utterance*. Ross’s Figure 5 below, illustrates the process in Ross used to determine a context for the at least one request using a recognition message:

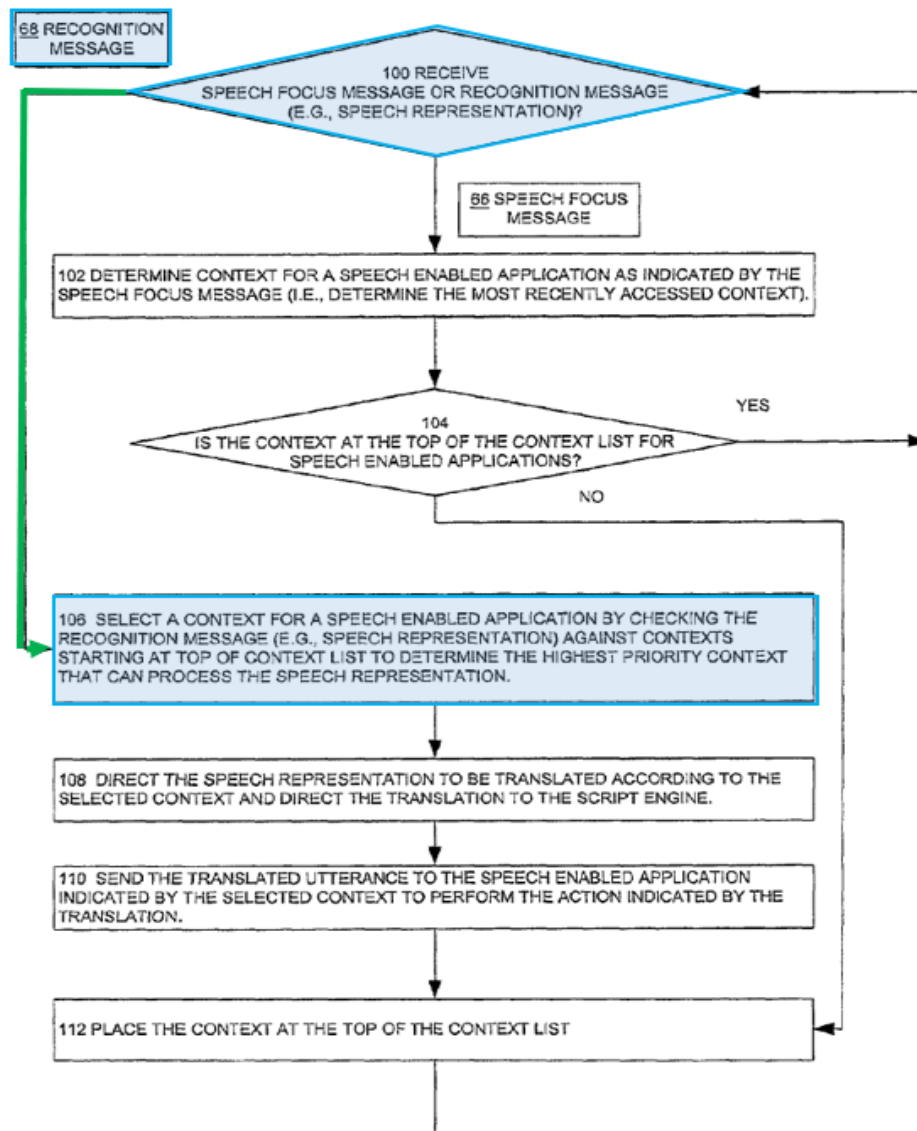


FIG. 5

Ross, Fig. 5 (annotated)

Ex. 1005, 7:17-21; 7:23-26 (“[T]he context manager 50 uses the grammar associated with each context 70 to determine if the recognition message 68 is capable of being processed by that grammar.”).

172. Ross then goes on to describe how determining a context for the request contained in the input speech (corresponding to the “utterance”) *is based on one or more keywords in the request*. For example, Ross discloses an “electronic mail application” that accepts keywords and phrases from the user such as “Open the first message,” “Create a message,” “Send this,” and “Print it,” based on the following sample grammar:

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

Ex. 1005, 7:60-66.

173. Ross uses a Backus Naur Form grammar in the above embodiment. Ex. 1005, 9:14-16. However, Ross recognizes that:

In other embodiments, context 70 may include another type of grammar, lists of keywords or phrases, hashed lists of words, or other suitable approaches to allow identification of a phrase as acceptable to a speech-enabled application 26.

Ex. 1005, 9:16-20.

174. A person of ordinary skill at the time would have therefore understood that Ross determines a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases.

175. Thus, it is my opinion that a POSITA would have understood that Ross in view of Ittycheriah discloses *parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases.*

6. Limitation 1[e]: “selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context;”

176. In my opinion, Ross renders obvious this limitation.

177. The '209 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.” Ex. 1001, 2:50-59. A person of ordinary skill in the art would have understand a “domain agent” to include autonomous executable software module specific to each application area. *See also* Section V.A (Overview of Natural Language Speech Recognition and Spoken Dialogue Systems).

178. The external applications in Ross *receive, process, and respond to requests based on the determined meaning* of the request (i.e., based on the determined context). Ex. 1005, 2:22-35. Ross discloses “external applications 26” that “communicate with the speech center **20**, a speech engine **22**, and active

accessibility module 24.” Ex. 1005, 4:16-21. Each of these external applications have a defined grammar, “along with associated translations for a script engine 38.”

Ex. 1005, 6:29-31. As Ross explains:

The speech center can allow cross-application speech. In one example, a user is working with a spreadsheet application, but then asks about the user’s next appointment and obtains the information from a calendar program. In another example, the user asks what is the current stock price for a particular company and obtains the stock price from a stock monitoring service. Furthermore, the user asks whether the user has any new messages and receives an answer supplied by a message or mail application. Applications do not even have to be running (i.e., currently performing on a computer system). For example, the user makes a request to take a letter, and a word processing application starts up (i.e., is launched by the speech center) if it is not already running.

Ex. 1005, 2:22-36.

179. Similarly, Ross discloses a task manager that “provides the capability to proceed with multiple execution requests simultaneously, to queue up additional script commands for busy applications 26, and to track the progress of execution.”

Ex. 1005, 5:1-5.

180. Therefore, a person of ordinary skill in the art would have understood Ross’s external, speech-enabled applications are *domain agent(s)* as claimed in the ‘209 patent. The external applications in Ross are autonomous executables triggered

by the script commands issued by Ross's task manager. This would have been well known at the time to a person of ordinary skill in the art.

181. Ross further discloses *selecting at least one domain agent based on the determined meaning*. Specifically, Ross “maintains a context list of speech enabled applications” that the user has accessed and teaches “deciding the target application (i.e., selecting a domain agent) for a received utterance based on the determined meaning. *See* Section X.C.5 (limitation 1[d]). After Ross determines the context for the received utterance, the system discloses selecting an application (i.e., a domain agent).

182. As an example, Ross teaches that if only one grammar for a speech-enabled external application accepts the received utterance, “that associated application 26 is the one that should be targeted to receive the corresponding command.” Ex. 1005, 8:33-36.

183. As explained below, Ross further selects the domain agent based on “recency of access” when more than one grammar of an external application is capable of accepting the received utterance. Ex. 1005, 8:37-40.

If the sentence is “print it” however, both grammars are capable of accepting the utterance. The context manager **50** therefore has to make a choice by referring to the context list **62** of applications in order of recency access. The context manager **50** tests the utterance against these grammars (indicated by the contexts **70** in the context list **62**) in

priority order, and passes the commands on to the first application 26 that has a grammar that will accept the phrase.

Ex. 1005, 8:37-40, 9:7-13.

184. Thus, it is my opinion that a POSITA would have understood that Ross discloses *selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context.*

7. Limitation 1[f]: “formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;”

185. In my opinion, Ross renders obvious this limitation.

186. After the system selects the target external application (i.e., domain agent), Ross translates the received utterance to be processed by the external application. Ex. 1005, 5:55-57 (“The utterance is then translated into the appropriate script engine 38 calls and dispatched to the target application 26), 6:15-18 (“When a successful match is found, the corresponding translation 74 is dispatched to the script engine 38 for execution[.]”), 6:48-51 (“The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26); 7:26-29 (“In step 108, the context manager 50

direct the speech representation to be translated according to the selected context 27[.]”); 9:9-13, Abstract; Fig. 5 (step 108).

187. This translation is then associated with the “grammar defined for each application.” Ex. 1005, 6:29-31 (“Toward that end, a grammar is defined for each application 26, along with the associated translations for a script engine.”).

188. A person of ordinary skill in the art would therefore have understood Ross’s “translation” of the received utterance *formulates the at least one request* (i.e., the command) *contained in the utterance in accordance with a grammar used by the selected domain to process requests associated with the determined context.* E.g., Ex. 1005, 6:48-51 (“The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26).

8. Limitation 1[g]: “invoking the selected domain agent to process the formulated request; and”

189. In my opinion, Ross renders obvious this limitation.

190. As I discussed previously, Ross discloses that after the received utterance is translated (i.e., formulated), the “appropriate script engine 38 calls and [the translated script] is dispatched to the target application.” Ex. 1005, 5:55-57; Fig. 2. The external application (i.e., the domain agent) then processes the command by executing the call. Ex. 1005, 6:15-17 (“When a successful match is found, the corresponding translation 74 is dispatched to the script engine 38 for execution.”),

10:19-23 (“the command is applied to the selected speech enabled application including launching the selected speech enabled application as needed and performing the function of the command.”).

191. A person of ordinary skill in the art would have understood that when the command is dispatched to the target application (i.e., the domain agent) for processing, it *invokes the selected domain agent to process the formulated request*.

192. Thus, it is my opinion that a POSITA would have understood that Ross discloses *formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context*.

9. Limitation 1[h]: “presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.”

193. In my opinion, Ross renders obvious this limitation.

194. As shown below, Ross provides several instances of *presenting results of the processed request* that were *generated as a result of the invoked domain agent processing the formulated request*. Ross describes presenting the results of the processed request as follows:

In one example, a user is working with a spreadsheet application, but then asks about the user’s next appointment and obtains the information from a calendar program. In another example the user asks what is the

current stock price for a particular company and obtains the stock price from a stock monitoring service. Furthermore the user asks whether the user has any new messages and receives an answer supplied by a message or mail application.

Ex. 1005, 2:22-30; *id.* at 2:31-36 (launching a word processing application in response to a request to take a letter), 8:47-9:6 (describing print requests made to electronic mail and calendar applications resulting in the printing of messages and appointments, respectively).

195. A person of ordinary skill in the art would have considered the performance of the function of the command by the external application (the *invoked domain agent*) that *presents results* in the form of an action responding to the user. Thus, it is my opinion that a person of ordinary skill in the art would have understood that Ross discloses *presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.*

D. Dependent Claim 4: “The method according to claim 1, wherein determining a context for the at least one request contained in the utterance includes:”

196. In my opinion, Ross in view of Ittycheriah renders obvious this limitation for the same reasons as claim 1, in addition to the below limitations 4[a]-4[c].

1. Limitation 4[a]: “matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords, the dynamically updated set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities”

197. It is my opinion that Ross in view of Ittycheriah renders obvious this limitation.

198. As I discussed above with respect to limitation 1[d], the output of Ittycheriah is the recognition message input into Ross’s conversation manager (the *parser*). See Section VII.C.5. Further, as I discussed above with respect to limitation 1[c], Ross’s “ordered list of applications (e.g., a context list 62),” is a *dynamic set of prior probabilities* that “may prioritize the contexts based on the access characteristic.” See Section VII.C.4; Ex. 1005, 3:1-2; *id.* 2:47-49.

199. Ross further discloses *matching the keywords contained in the recognized words and phrases* to those keywords in the context list, for example describing how a request to “print an appointment” would be matched to keywords associated a calendar application. Ex. 1005, 8:1-24 (discussing keywords for mail and calendar applications).

200. For example, Ross discloses sample grammars associated with different applications such as “appointment” for a calendar application:

This grammar allows phrases to be spoken by a user of the electronic mail application 26, such as:

“Open the first message.”

“Create a message.”

“Send this.”

“Print it.”

The following is a sample grammar for a calendar application 26:

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

This grammar allows phrases to be spoken by a user of the calendar application 26, such as:

“Open the first appointment.”

“Create an appointment.”

“Print the fourth appointment.”

“Print it.”

Ex. 1005, 8:1-24 (discussing keywords for mail and calendar applications).

201. In fact, the Ross would inform a person of ordinary skill in the art that the applications in the context list each have *associated keywords* merely as a design choice. Ex. 1005, 9:16-20 (“In other embodiments, context 70 may include another type of grammar, lists of keywords or phrases, hashed lists of words, or other suitable approaches to allow identification of a phrase as acceptable to a speech-enabled application.”). A person of ordinary skill in the art would thus understand that

matching keywords contained in the recognized words to keywords in Ross's context stack is both disclosed, but also merely a design choice that a person of ordinary skill in the art could adopt.

202. Thus, it is my opinion, that a person of ordinary skill in the art would have understood that Ross in view of Ittycheriah discloses *matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords, the dynamically updated set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities.*

2. Limitation 4[b]: “scoring one or more possible contexts for the matched keywords using one or more of a profile of the user, data content associated with the domain agents, or a context stack of one or more recent contexts; and”

203. In my opinion, Ross renders obvious this limitation.

204. As I discussed with respect to limitation 1[b] previously, Ross uses “recency of relevant access to the context” (i.e., a relevance score) as an “access characteristic” such that Ross “is biased to send commands to the most recently accessed application.” *See* Section X.C.3; Ex. 1005, 8:46-53. A person of ordinary skill in the art would understand Ross's description to be a definition of a “stack” (or content stack), as it is a data structure that employs a “last in, first out” (LIFO), which means you first turn to the most recent context that you just placed on the stack.

205. As I discussed previously, Ross uses the example of the spoken command “print it,” which can be matched to either an electronic mail application context or a calendar application context. See Section X.D.1; Ex. 1005, 8:1-24.

206. Ross scores the most recent, relevant context as the highest, forming a context list (*a context stack of one or more recent context*). Ross explains that:

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

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

Ex. 1005, 8:38-53, 8:54-9:13.

207. A person of ordinary skill in the art would have therefore understood that Ross scores the one or more possible contexts based on the context stack of one or more recent contexts, by “test[ing] the utterance against” multiple grammars in

the context list and scoring the possible contexts as not matching or matching. Ex. 1005, 8:37-46, 8:54-9:1.

208. It is therefore my opinion that a person of ordinary skill in the art would have therefore understood that Ross discloses *scoring one or more possible contexts for the matched keywords using one or more of a profile of the user, data content associated with the domain agents, or a context stack of one or more recent contexts.*

3. Limitation 4[c]: “determining the context by selecting one or more of the possible contexts having a score satisfying a predetermined confidence level.”

209. In my opinion, Ross renders obvious this limitation.

210. As I discussed with respect to limitation 1[e] previously, Ross discloses scoring possible contexts based on whether the application is capable of accepting the received utterance. *See* Section X.C.6; Ex. 1005, 8:37-40, 9:7-13.

211. From these disclosures in Ross, a person of ordinary skill in the art would have understood that by finding a context that accepts the utterance, that context must have a *score satisfying a predetermined confidence level.*

212. A person of ordinary skill would have however, would understand that a context that does not match the utterance well enough would not meet the predetermined confidence level. In contrast, a POSITA would understand that a context that does not match the utterance well enough would not meet the predetermined confidence level.

213. It is therefore my opinion that Ross discloses *determining the context by selecting one or more of the possible contexts having a score satisfying a predetermined confidence level*.

XI. OPINIONS ON GROUND 2:

A. A POSITA Would Have Been Motivated to Combine Ross, Ittycheriah, and Ross018

214. In my opinion, a person of ordinary skill in the art would have been motivated to use Ross018's more sophisticated parser in conjunction with Ross to "provide the speech center system with more knowledge about a domain," and to 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." Ex. 1009, 2:3-8.

215. Ross and Ross018 are related patent applications that share a common provisional, No. 60/261,372. Ex. 1016. A person of ordinary skill in the art would have been motivated to consult patents related to Ross, including Ross018, in developing a natural language processing system, to determine what additional features or qualities to consider developing in such a system.

216. The shared provisional of Ross and Ross018 explain that the two patents serve to complement each other, with the technology recited in Ross018 being "basically an expansion" of Ross's conversation manager. Ex. 1016.

217. A person of ordinary skill in the art would thus be motivated to add the parsing features of Ross018 to Ross to better respond to user's utterances it receives

by incorporating additional information about the speech-enabled applications it controls and using additional reasoning to generate a response.

218. Moreover, by increasing the sophistication and complexity of the parser, allows for a more natural interaction between the user and Ross's conversation manager.

B. A POSITA Would Have Had a Reasonable Expectation of Success

219. In my opinion, a person of ordinary skill in the art would have had a reasonable expectation of success in combining the teachings of Ross, Ittycheriah, and Ross018.

220. Ross teaches that its speech engine 22, as shown below in Figure 2, could be any commercially-available speech recognition engine. Accordingly, a person of ordinary skill in the art would have used the software algorithms of Ittycheriah's speech recognition process to implement the details of speech engine 22:

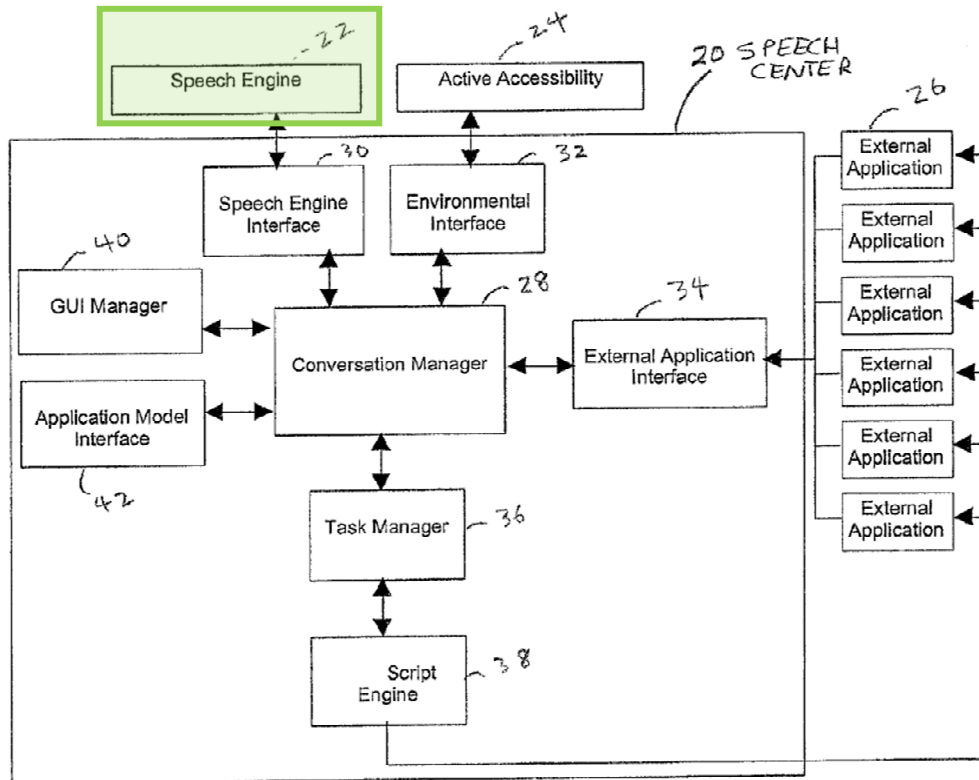


Fig. 2

Ex. 1005, Fig. 2 (annotated)

This is particularly true since both Ross and Ittycheriah were developed by IBM.

221. A POSITA would have also understood that Ittycheriah's speech recognition system would be compatible with Ross, as Ittycheriah discloses that the system's decoded keywords and scripts could be used to direct speech requests to external speech-enabled applications to perform their respective automatic functions, just like in Ross. Ex. 1007, 4:63-5:3 ("It is to be appreciated that the output of the comparator 22 (decoded keyword) and the output of the engine 16

(decoded script) may be connected to the actual phone system or navigation system [speech recognition applications] such that the phone or navigation systems are responsive to the decoded keyword or script and thus perform the respective automatic functions (i.e., dialing or navigating).”).

222. A person of ordinary skill in the art 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 (e.g., known systems for speech recognition and processing natural language requests) according to known methods (e.g., known methods of 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 adding knowledge of a user’s domain, plans, tasks, and processes in parsing the user’s natural language utterance.

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

223. It is my opinion, that Ross and Ittycheriah in view of Ross-018 discloses claim 6 for the same reasons as claim 1, in addition to the below limitations 6[a]-6[d].

1. Limitation 6[a]: “determining one or more required values and one or more optional values for a request in the grammar used by the selected domain agent”

224. In my opinion, Ross in view of Ross018 discloses this limitation.

225. First, Ross discloses a *domain agent associated with the determined context*. The '209 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.” Ex. 1001, 2:50-59. A person of ordinary skill in the art would understand a “domain agent” as claimed would include an autonomous executable software module specific to each application area.

226. Ross discloses “external applications 26” that “communicate with the speech center 20, a speech engine 22, and active accessibility module 24.” Ex. 1005, 4:16-21. Each of these external applications have a defined grammar, “along with associated translations for a script engine 38.” Ex. 1005, 6:29-31. And 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. Ex. 1001, Abstract, 2:22-36. As Ross explains:

The speech center can allow cross-application speech. In one example, a user is working with a spreadsheet application, but then asks about the user’s next appointment and obtains the information from a calendar program. In another example, the user asks what is the current stock

price for a particular company and obtains the stock price from a stock monitoring service. Furthermore, the user asks whether the user has any new messages and receives an answer supplied by a message or mail application.

Applications do not even have to be running (i.e., currently performing on a computer system). For example, the user makes a request to take a letter, and a word processing application starts up (i.e., is launched by the speech center) if it is not already running.

Ex. 1005, 2:22-36; *see id.* 2:41-42 (“Any application that supports a robust automation user interface can participate.”).

227. Similarly, Ross discloses a task manager that “provides the capability to proceed with multiple execution requests simultaneously, to queue up additional script commands for busy applications 26, and to track the progress of execution.”

Ex. 1005, 5:1-5.

228. Based on these disclosures in Ex. 1005, a person of ordinary skill in the art would thus understand that the external applications in Ross are autonomous executables triggered by the script commands issued by Ross’s task manager.

229. Accordingly, each of Ross’s external, speech-enabled applications is a *domain agent* as claimed in the ’209 patent.

230. 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.” Ex. 1009, 11:61-65.

231. According to Ross018, each speech-enabled external application (i.e., *domain agent*) to “load custom grammars or define task specific vocabulary” to Ross’s speech center. Ex. 1009, 5:21-30. Using those custom grammars or defined task-specific vocabulary, Ross018 discloses that once a natural language 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 15.” Ex. 1009, 5:63-56.

232. Ross018 also discloses that in translating (i.e., *formulating*) the request contained in the utterance *in the grammar used by the domain agent*, Ross018 *determines one or more required values and one or more optional values*. For example, Ross018 states: “The utterance 15 is then translated into the appropriate script engine 38 calls and dispatched to the target application 26.” Ex. 1009, 5:67-6:2.

233. Specifically, Ross discloses that its conversation manager “converts recognized utterances 15 into sequences of script calls to frame-building functions.” Ex. 1009, 9:51-59. These frame-building functions are executed to create “an initial

semantic representation of the utterance 15.” Ex. 1009, 9:59-62. Ross018 then uses these initial semantic representations to create a series of “attribute-object-value triples, which are termed ‘propositions.’” Ex. 1009, 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,” as shown in the example below:

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

Ex. 1009, 10:5-15; *see id.* at 7:35-40

234. These values represent “the kinds of objects, attributes, and operations that the application 26 makes available” to Ross018’s speech center. Ex. 1009, 7:41-51.

235. These attribute and object values in Ross018 align with the way the term “values” is used in the ’209 patent as criteria that describe different attributes. For example, the ’209 describes personal profile values such as “gender” and “occupation.” *See* ’209 patent, 14:30-48, 18:10-14.

236. Ross018 also discusses the use of grammars with respect to its dialogue manager: “The dialog manager 56 serves as a traffic cop for information flowing back and forth between the reasoning facility 52 and the user. Questions generated by the reasoning facility 52 as well as answers derived to user questions and unsolicited announcements by the speech center system 20 are all processed by the dialog manager 56. The dialog manager 56 also is responsible for managing question-answering grammars, and converting incomplete answers generated by the user into a form digestible by the reasoning facility 52.” Ex. 1009, 11:18-27.

237. Although Ross and Ross018 discuss BNF grammars, Ross also recognizes that any other grammars could be used as a matter of design. Ex. 1005, 9:14-20 (“In a preferred embodiment, the grammar used for each speech-enabled application 26 may be a Backus Naur Form (BNF) grammar as shown above. In other embodiments, context 70 may include another type of grammar, lists of keywords or phrases, hashed lists of words, or other suitable approaches to allow identification of a phrase as acceptable to a speech-enabled application 26.”).

238. By 2002, it was well known to use grammars that would have optional elements. For example, the JSpeech Grammar Format was developed as part of the World Wide Web consortium and is based on the Java programming language. It uses square brackets around tokens (like [speed value]) to indicate optional values. Likewise, variants of BNF grammar commonly used indicators for optional values.

For example, even looking at Ross, it shows a sample grammar in which the word “the” is optional for the command to be accepted, as denoted by the question mark (?):

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

Ex. 1005, 8:8-17.

239. The '209 patent itself does not distinguish between what it considers to be a “required value” and an “optional value.” Ex. 1001, 18:10-12, 18:49-52, 24:58-61, 25:33-36. Nevertheless, because the triples in Ross018 would be inclusive of all values contained in the utterance regardless of whether they are required to formulate the request, a POSITA would understand such triples would include both optional and required values. Ex. 1005, 9:49-10:35. These attribute and object values in Ross018 align with the way the term “values” is used in the '209 patent as criteria that describe different attributes, such as “gender” and “occupation” as personal profile values. Ex. 1001, 14:28-46, 18:8-16. Accordingly, a person of ordinary skill in the art would understand that Ross018’s determination of attribute and object values determine *one or more required values and one or more optional values associated with formulating request in the grammar used by the domain agent.*

2. Limitation 6[b]: “extracting one or more criteria and one or more parameters from the keywords contained in the recognized words and phrases using procedures sensitive to the determined context;”

240. In my opinion, Ross in view of Ross018 discloses this limitation.

241. '209 patent uses the terms “criteria,” “parameters” and “values,” as claimed in the '209 patent, somewhat interchangeably. For example, the '209 patent appears to consider “specific parameters and values” to also be criteria. Ex. 1001, 11:59-64 (“The system agent 150 manages the criteria handlers 152, which handle specific parameters or values (criteria) used to determine context for questions and commands.”), 14:10-16 (describing parameters as including “criteria”). However, the '209 patent lists examples of “criteria” for weather as the “location, date and time,” and for commands as “yes/no, on/off, pause/stop.” Ex. 1001, 18:26-28. It also describes “parameters” in a stock report as the stock’s “name and the company name.” Ex. 1001, 14:60-62. Accordingly, a person of ordinary skill in the art would understand that “criteria” refers to the attributes of a request (e.g., location) and a parameter acts as the value of that attribute (e.g., Alexandria or Virginia).

242. As discussed above with respect to values in Section XI.C.1 (Ground 2, 6[a]), Ross018 describes *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*. In generating its “propositions,” which are the set of attribute-object-value triples that are an “initial semantic representation of the utterance,”

Ross018 converts the recognized utterance into frame representations that serve as input to the reasoning facility. Ex. 1009, 9:59-62.

243. A person of ordinary skill in the art would understand that the one or more “criteria” of the ’209 patent corresponds to the one or more “attributes” of Ross018’s attribute-object-value triples. For example, Ross018 shows 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:

```
5  ┌───────────────────────────────────────────────────────────────────────────────────┐
    │ Begin("command")                                                                │
    │   AssociateValue("action")                                                       │
    │   Begin("action")                                                                │
    │     AssociateClass("make")                                                       │
    │     AssociateValue("patient")                                                    │
    │     Begin("thing")                                                              │
    │       AssociateClass("column")                                                    │
    │       End("thing")                                                              │
    │       AssociateValue("destination")                                              │
    │       AssociateParameter("green")                                               │
    │     End("action")                                                                │
    │   End("command")                                                                │
15 └───────────────────────────────────────────────────────────────────────────────────┘
```

Ex. 1009, 9:59-10:35

244. Ross018 also develops its attribute-object-value triples in *using procedures sensitive to the determined context*. Namely, 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. Ex. 1009, 6:40-47, 7:10-21.

245. By accounting for the domain model in developing its attribute-object-value triples, Ross018 thus uses procedures sensitive to the determined context to extract its one or more criteria and one or more parameters.

3. Limitation 6[c]: “inferring one or more criteria and one or more parameters using the dynamic set of prior probabilities or fuzzy possibilities; and”

246. In my opinion, Ross in view of Ross018 discloses this limitation.

247. Ross018 discloses a “reasoning facility” that “performs the reasoning process for the conversation manager.” Ex. 1009, 10:36-37.

248. As Ross018 explains, again 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.” Ex. 1009, 10:31-35. A person of ordinary skill in the art would thus understand that Ross018’s reasoning facility is thus required to identify *one or more further criteria and one or more further parameters associated with the request*.

249. Ross018’s reasoning facility accomplishes this with an *inference engine* that *uses a dynamic set of prior probabilities or fuzzy possibilities*. Ross018 describes the reasoning facility as “a goal-directed rule based system composed of

an inference engine, memory, rule base and agenda.” 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.” Ex. 1009, 10:36-51. In other words, the reasoning facility creates attribute-object-value triples in which one of the elements is a “variable” that must be inferred.

250. The reasoning facility *uses fuzzy possibilities* by scanning through already registered rules to see what, if any, rules include “actions unifying with the goal” searching for a match. For example, if a match is found, “[t]hese become new goals for the inference engine of the reasoning facility 52 to achieve, based on the content of the memory and the conversational record.” Ex. 1009, 10:57-58.

251. A person of ordinary skill in the art would thus understand that Ross018’s reasoning facility *infers one or more further criteria and one or more further parameters associated with the request*.

252. To the extent it is not able to infer the further criteria or parameters, the reasoning facility will generate a question to the user to facilitate “achiev[ing] the goals derived from the user’s questions and commands.” Ex. 1009, 10:59-64. As an 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.” Ex. 1009, 11:8-10. The speech center would be required to infer the “appointment object whose time attribute is changing” as an additional criteria and parameter. Ex.

1009, 11:10-14. Using the dynamic list of prior probabilities or 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. Ex. 1009, 11:14-16.

253. Moreover, the prior probabilities or fuzzy possibilities are *dynamic*. Ross018 discloses that "[e]ach utterance is indexed in the conversational record 60, along with the results of the semantic analysis. This information is eventually purged from the conversational record when it is no longer relevant to active goals and after some predefined period of time has elapsed." Ex. 1009, 11:2-7. Because the conversational record and any corresponding prior semantic analysis are constantly evolving, a person of ordinary skill in the art would understand that these prior probabilities or fuzzy possibilities, as used by the inference engine, are dynamic in nature.

4. Limitation 6[d]: "transforming the extracted and the inferred criteria and parameters into a format compatible with the grammar used by the selected domain agent, the transformed criteria and parameters including all of the required values and one or more of the optional values."

254. In my opinion, Ross in view of Ross018 discloses this limitation.

255. 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).” Ex. 1009, 12:15-20.

256. This is because Ross018’s reasoning facility, in generating its condition and action propositions (the attribute-object-value triples with variables) “can contain embedded script function calls” for external applications (i.e., *domain agents*). A person of ordinary skill in the art 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*.

257. 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 person of ordinary skill in the art to be a collection of tokens, where each token is an individuated or non-reducible text element of data. The value assigned to a token during a parse, in the context of a lexical grammar, is a specific instance of a word or phrase that appears in a spoken utterance. A grammar, which can be expressed in BNF for example, defines syntactic rules on tokens.

258. Moreover, a person of ordinary skill in the art would have found it obvious based on Ross018 that the script call or command (i.e., the collection of *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.

259. 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.” Ex. 1009, 10:59-61. Accordingly, a person of ordinary skill in the art 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. And if the required values are absent, Ross018 would prompt the user for more information, including for additional values.

XII. OPINIONS ON GROUND 3:

A. A POSITA Would Have Been Motivated to Combine Ross, Ittycheriah and Aridor

260. A person of ordinary skill in the art would have been motivated to add Aridor's “knowledge agents” that perform meta-searching over the Internet to Ross's external applications (*domain agents*) to collect information from a large body of documents (like the Internet) and to filter and rank the results based on relevance.

261. Ross discloses that its external applications “perform the function of the command” in the natural language utterance. Ex. 1005, 9:44-50, 12:6-13. Ross, however, does not disclose how its external applications obtain, and process information sourced from the Internet. However, by 2002, a person of ordinary skill

in the art would understand that certain information and responses to queries could be found by searching the Internet.

262. Accordingly, a person of ordinary skill in the art would have been motivated to look for and rely on Internet searching mechanism to aid in retrieving and processing responses to a user's queries, such as those disclosed in Aridor, also developed by IBM. A person of ordinary skill in the art would also understand that Aridor's metasearch functionality would be compatible with Ross, as Aridor operates by using separate knowledge agents based on domain, such that each of Ross's external applications could have its own knowledge agent specific to their domain.

263. 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 limitation 1[e]-1[g] (Section X.C.6-8). Ex. 1010, 3:53-59. A person of ordinary skill in the art would further be motivated to add to Ross the "ranking and filtering" features of Aridor that process the results of processing the user's query (natural language question or command) to find the most relevant information.

264. In addition, a person of ordinary skill in the art 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. Ex. 1010, 3:65-4:2, 14:56-15:19.

B. Reasonable Expectation of Success

265. Aridor and Ross are both products from IBM, and Aridor simply expands on Ross’s ability to process natural language commands and questions by expanding the source of information to the Internet.

266. A person of ordinary skill in the art 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 (e.g., known systems for speech recognition and Internet searching based on text queries) according to known methods (e.g., known methods of 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.

267. A person of ordinary skill in the art 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 (e.g., known systems for speech recognition and Internet searching based on text queries) according to known methods (e.g., known methods of 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.

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

268. In my opinion, Ross combined with Ittycheriah discloses this limitation for the same reasons as claim 1. Section X.C. It is also my opinion that Ross in combination with Aridor discloses claim 7 for the reasons set forth in this paragraph and further below.

269. It is my opinion that a POSITA would 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). Ex. 1005, 2:25-30.

1. Limitation 7[a]: “directing the at least one command to local or remote processing devices; and”

270. In my opinion, Ross in view of Aridor discloses this limitation.

271. Ross describes requests that include *at least one question and at least one command*. For example, Ross describes the user asking, “what is the current

stock price for particular company,” which would require “obtain[ing] the stock price from a stock monitoring service.” Ex. 1005, 2:22-30.

272. A person of ordinary skill in the art would understand this request to include both a question of “what is the current stock price,” and a command to retrieve the current stock price from the stock monitoring service.

273. Taking Ross’s example of asking ‘what is the current stock price for a particular company,” Ross would direct that question to the “stock monitoring service,” which is one of its described external applications (*domain agents*). Ex. 1005, 2:22-30. But Ross is not explicit on how its domain agents (the stock monitoring services) process and return results to the user’s query. For this reason, a person of ordinary skill in the art would add Aridor’s knowledge agent to Ross. For example, Aridor discloses that “[a] user 22 uses a computing device 24 to connect to a network 28, typically the Internet, and to conduct a search among information repositories, such as Web sites 26,” which can be “any suitable general purpose computer processor having a connection to network 28.” Ex. 1010, 8:1-12.

2. Limitation 7[b]: “querying local or network information sources to resolve information requested in the at least one question.”

274. In my opinion, Ross in view of Aridor discloses this limitation. As discussed above with respect to limitation 7[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 XII.C.1.

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

275. In my opinion, Ross combined with Ittycheriah and Aridor discloses this limitation for the same reasons as claims 1 and claim 7. Section X.C; Section XII.C. It is also my opinion that Ross in combination with Aridor discloses claim 8 for the reasons set forth in this paragraph and further below.

1. Limitation 8[a]: “submitting a plurality of asynchronous queries to the information sources, the plurality of queries including a plurality of duplicate queries submitted to distinct information sources;”

276. In my opinion, Ross in view of Aridor discloses this limitation.

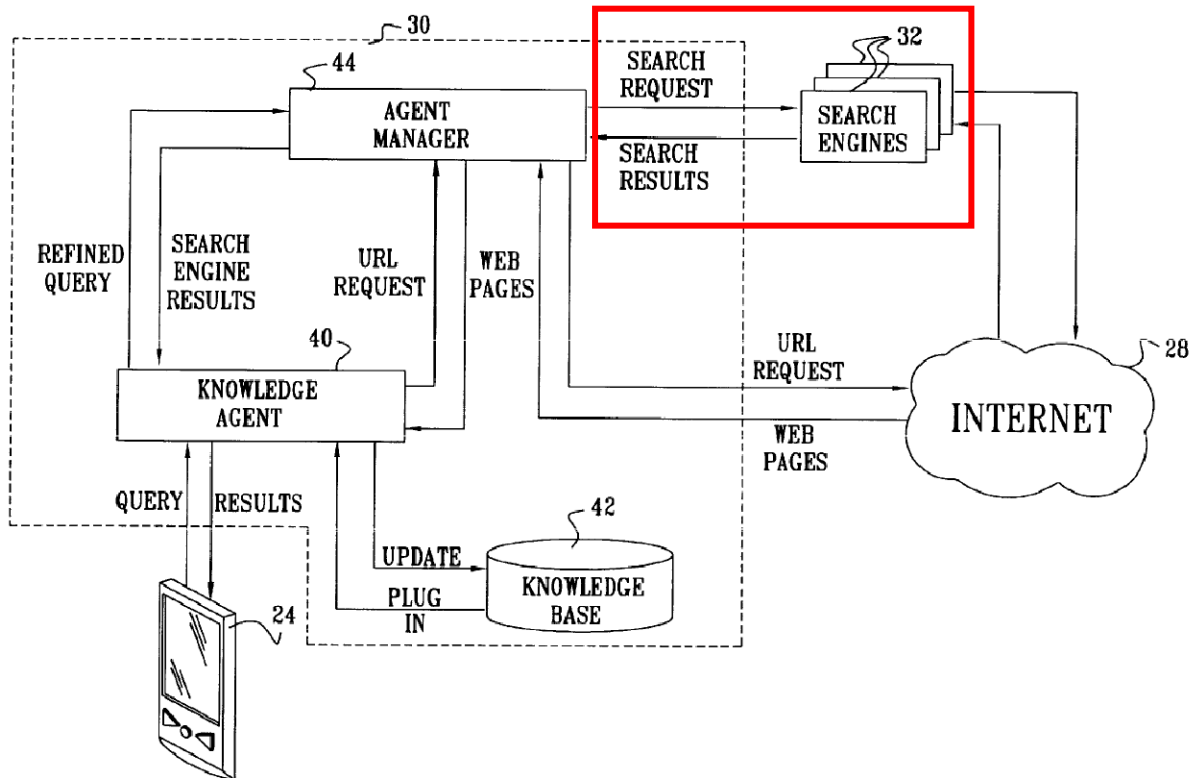
277. As discussed above (Motivation to Combine), a person of ordinary skill in the art would be motivated to add to Ross’s external application agents (*domain agents*) the features and functionalities of Aridor’s “agent manager” and “knowledge agents” (also referred to as the “knowledge agent program”) to facilitate searching a corpus of documents either locally or on the Internet. Sec. XI.A.

278. For example, 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.” Ex. 1010, 3:33-35. In performing the search, Aridor’s agent manager and knowledge agent refines a query and *submits a*

plurality of asynchronous queries to the information sources. Ex. 1010, 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 specified by user 22, such as AltaVista, Google or Hotbot[.]”), Fig. 2.

279. Aridor further discloses that agent manager “passes the refined query to a number of search engines.” Ex. 1010, 4:10-16, 8:22-27 (“A knowledge agent program...is responsible for carrying out the search for user 22. 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.”). This is also shown in Figure 2 of Aridor, in which the same search request is submitted to multiple search engines:

FIG. 2



Ex. 1010, Fig. 2 (annotated)

A person of ordinary skill in the art would understand that a metasearch that sends the same search request multiple times to different search queries are *duplicate queries submitted to distinct information sources*.

2. Limitation 8[b]: “a synchronously receiving a plurality of results of the queries from the information sources; and”¹

280. In my opinion, Ross in view of Aridor discloses this limitation.

¹ For the purposes of this Petition, I understand that Petitioner assumes this claim recites “asynchronously.”

281. 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." Ex. 1010, 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).

282. A person of ordinary skill in the art would understand that each user-selected search engine sends results *asynchronously*. 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. Ex. 1010, 8:52-59.

3. Limitation 8[c]: "scoring a relevance of each of the asynchronously received results using the dynamic set of prior probabilities or fuzzy possibilities to determine one or more best responses to the at least one question"

283. In my opinion, Ross in view of Aridor discloses this limitation.

284. 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. Aridor describes that once the results of the metasearch are received, they are returned to the knowledge agent for "ranking and filtering." Ex. 1010, 8:56-57.

Specifically, 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."

Ex. 1010, 11:19-34.

285. 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." Ex. 1010, 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." Ex. 1010, 11:32-34.

286. Aridor *uses the dynamic set of prior probabilities or fuzzy possibilities* to score the relevance of the results. 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." Ex. 1010, 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 in returning results to the user. Ex. 1010, 14:35-43. A person of ordinary skill in the art 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.

287. 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." Ex. 1010, 14:35-43, 14:56-15:19.

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

288. In my opinion, Ross combined with Ittycheriah and Aridor discloses this limitation for the same reasons as claims 1, 7 and 8. Section X.C, Section XII.C-D. It is also my opinion that Ross in combination with Aridor discloses claim 9 for the reasons set forth in this paragraph and further below.

1. Limitation 9[a]: "determining one or more tokens required to formulate a response to the question;"

289. In my opinion, Ross in view of Aridor discloses this limitation.

290. 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."). Ex. 1010, 9:42-49. Specifically, Aridor discloses that it 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.” Ex. 1010, 11:55-59.

291. A person of ordinary skill in the art 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). This aligns with how a person of ordinary skill in the art would understand the word “token,” i.e., as an individuated or non-reducible text element of data. A token, in the context of a lexical grammar, is a specific instance of a word or phrase that appears in the formulated response.

2. Limitation 9[b]: “extracting one or more values for the determined tokens from the asynchronously received results, the one or more values extracted by applying one or more scraping criteria to the asynchronously received results;”

292. In my opinion, Ross in view of Aridor discloses this limitation.

293. Aridor discloses how once it submits the query to “one or more general purpose search engines,” the results are collected from those search engines, which Aridor calls the “root set 54 of sites 26.” Ex. 1010, 10:18-23. Next, Aridor 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.
Ex. 1010, 10:40-11:18.

294. Aridor then *applies one or more scraping criteria* to the results in the “site list” to *extract one or more values for the determined tokens (the asynchronously received results)*. Specifically, Aridor scrapes the results for both textual similarity and for topological aspects, assigning scores for both features. Ex. 1010, 11:19-34, 11:46-51, 12:13-62, Fig. 3.

295. For textual similarity, Aridor scrapes the words in each page and divides the words into three grounds: “Strong text includes the words that appear in the title or in large font headers; medium text includes words that are either highlighted (bold, italics, etc.) or in small font headers; and all the rest is regular text.” Ex. 1010, 11:46-51. Aridor then applies weights to each type of word (strong, medium, and regular) and computes a “query textual score,” and in a similar way, computes a “domain score” based on those documents in the knowledge base. Ex. 1010, 12:13-60. The “query textual score” and “domain score” are then “normalized and combined to create the overall textual similarity score 68 for each site[.]” Ex. 1010, 12:60-62.

296. 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 to surfers such 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 “is presumably answers the specific query.” Ex. 1010, 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*). Ex. 1010, 14:11-23.

3. Limitation 9[c]: “evaluating the extracted values using the dynamic set of prior probabilities or fuzzy possibilities, wherein evaluating the extracted values includes resolving ambiguous, incomplete, or conflicting information; and”

297. In my opinion, Ross in view of Aridor discloses this limitation.

298. As discussed above with respect to limitation 9[b], Aridor extracts values via a scraping of text and a scraping of links in the site list of potential results. Section XI.E.2. Further, as described above with respect to limitation 8[c], Aridor’s “textual score” and “link topology score” both use *the dynamic set of prior probabilities or fuzzy possibilities* in how it computes its combined score, which uses “persistent, acquired knowledge in a domain,” to improve the quality of the received results. Section XII.D.3; Ex. 1010, 3:53-4:11, 14:56-15:11.

299. 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,” used in deriving

the textual and topological information. Ex. 1010, 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. By taking into account textual affinity to the particular query and to the domain of interest, as well as topological information, Aridor is thus able to find the most “authoritative” pages.” A person of ordinary skill in the art would understand that this is certainly a good way of “resolving ambiguous, incomplete, or conflicting information.

300. A person of ordinary skill in the art would understand that by using sites saved in a domain’s knowledge base to increase the accuracy of results would *include resolving ambiguous, incomplete, or conflicting information.*

4. Limitation 9[d]: “selecting one or more best values for the determined tokens from the evaluated values, wherein the one or more best responses include the one or more selected best values.”

301. In my opinion, Ross in view of Aridor discloses this limitation.

302. 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.” Ex. 1010, 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*.

303. By ranking and then downloading the “top few entries,” a person of ordinary skill in the art 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*.

XIII. OPINIONS ON GROUND 4:

A. A POSITA Would Have Been Motivated to Combine Hartono and Baker

304. In my opinion, a person of ordinary skill in the art would have been motivated to use Baker’s speech recognition method as Hartono’s “automatic speech recognition system,” to implement a way to quickly complete the speech recognition process based on both acoustic probabilities and context probabilities.

305. Hartono discloses that speech is input into an “automatic speech recognition” module, shown below labeled as item 102a in Figure 2:

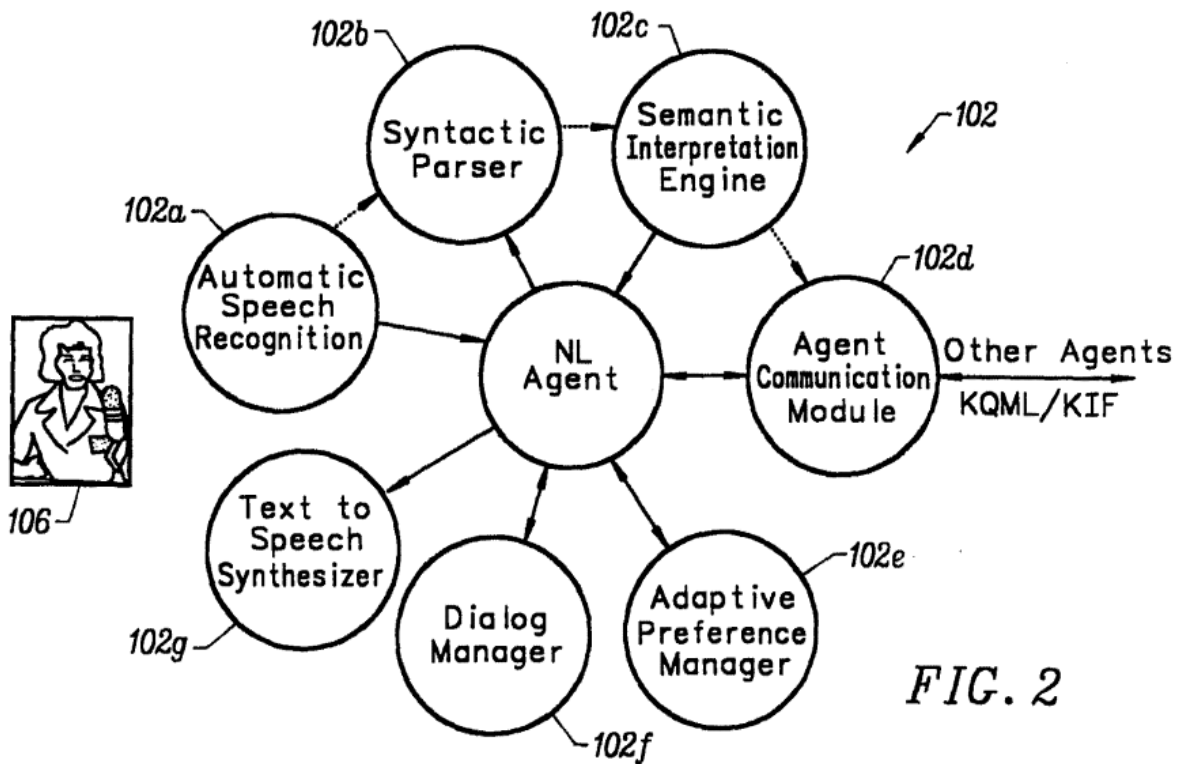


FIG. 2

306. Hartono, however, recognizes that by 1999, “automatic speech recognition systems for speech input are readily and commercially available off the shelf.” Ex. 1006, 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 in the natural language interface system 100 of the present invention.” Hartono 11:4-12.

307. In my opinion, a POSITA reading Hartono would thus be motivated to look for commercially available speech recognition systems to perform the method steps described in Hartono.

308. Baker is such a speech recognition system developed by Dragon Systems, Inc., a well-known developer of speech recognition software in the mid-to-late 1990s. Ex. 1008, Cover.

309. In my opinion, 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." Ex. 1008, 1:5-15.

310. Hartono also employs the Government and Binding approach to its syntactic parsing, which is based on the "implied syntactic structure in English language sentences." Ex. 1006, 12:1-10. Baker's speech recognition system that recognizes words based on their context within sentences, is thus compatible with Hartono's syntactic parser, which aims to accurately identify noun phrases, verb phrases, and adjective phrases. Ex. 1006, 12:15-13:10.

311. Thus, it is my opinion that a POSITA would be motivated to use Baker's speech recognition system with Hartono's because Baker recognizes words "on the fly" as data that is needed based on both acoustic matching and context matching probabilities based on existing and on-line reference sources, instead of by building data-intensive statistical language models or data-training sets. Ex. 1008, 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.

B. A POSITA Would have Had a Reasonable Expectation of Success

312. In my opinion, a person of ordinary skill in the art would have had a reasonable expectation of success in combining the teachings of Hartono and Baker.

313. Specifically, a POSITA would have used the software algorithms of Baker's speech recognition process as Hartono's automatic speech recognition system, which can be "[a]ny suitable off the shelf speech recognition system." Ex. 1006, 11:4-12.

314. A POSITA would have also 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 (e.g., known systems for speech recognition and processing natural language requests) according to known methods (e.g., known methods of 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.

C. Independent Claim 1

315. In my opinion Hartono and Baker disclose the independent claim.

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

316. In my opinion, Hartono discloses the preamble to the extent limiting.

317. Hartono discloses a “method for receiving commands and/or adaptively outputting results using a natural language speech interface[.]” Ex. 1006, 8:8-10; *see id.* at 1:19-23 (“provides a method for receiving commands, executing received commands and adaptively interacting with the user using a natural language interface.”), 4:20-25 (“In one embodiment, the 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 from the natural language agent and to output an executable instruction to an application program.”), 5:16-24, 10:13-20.

318. Thus, it is my opinion that a POSITA would have understood that Hartono discloses *a method responsive to a user generated natural language speech utterance.*

2. Limitation 1[a]: “receiving the user generated natural language speech utterance, the received user utterance containing at least one request;”

319. In my opinion, Hartono discloses the limitation.

320. Hartono teaches “receiving commands and/or adaptively outputting results and responses using a natural language interface, such as a natural language speech interface.” Ex. 1006, 4:9-11, 8:8-10; *see* Ex. 1006, 1:20-23 (“[T]he present invention provides a method for receiving commands, executing received

commands, and adaptively interacting with the user using a natural language interface, such as a natural language speech interface.”), 4:20-25 (“In one embodiment, the natural language interface for a computer system includes a natural language agent adapted to receive and interpret the natural language user input[.]”), 5:11-12 (“The natural language interface instructions generally comprises receiving natural language user input...”), 5:17-23 (similar).

321. 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.” Ex. 1006, 9:3-10.

322. A POSITA would understand a “command” to be *at least one request* as claimed in the ’209 patent.

323. Hartono also provides several examples of *receiving a user generated natural language speech utterance* that include *at least one request*, such as “‘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?’” Ex. 1006, 29:2-6; *id.* at 25:20-23 (“For example, requests such as ‘How do I,’ ‘Can you show me,’ ‘What are the possible values for’ will result in a response from the application class agent 102 with instructions and explanation on how to perform the task.”).

324. Thus, it is my opinion that a POSITA would have understood that Hartono in view of Baker discloses *receiving the user generated natural language speech utterance, the received user utterance containing at least one request.*

3. Limitation 1[b]: “maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance;”

325. In my opinion, Hartono and Baker disclose the limitation.

326. As noted above in Section X.3, the '209 patent does not define what is “each stage of processing the received user utterance.” Accordingly, the Petition addresses this limitation as it would apply to recognizing step 1[c] and parsing step 1[d]. In my opinion, a person of ordinary skill in the art would not have considered any of the other claimed steps related to a “stage of processing the received user utterance,” but steps associated with selecting and invoking a speech-enabled application. As such, I will address this limitation as it relates to limitation 1[c] and 1[d] here. However, to the extent prior probabilities or fuzzy possibilities are required for other limitations, I will address those separately in the section for the respective limitations. It is my opinion that a person of ordinary skill in the art would have understood that those prior probabilities or fuzzy possibilities would likewise be maintained as part of the proposed combinations.

327. To the extent “maintaining” step 1[b] applies to the “recognizing” step 1[c], while Hartono alone does not disclose this limitation, it is my opinion that

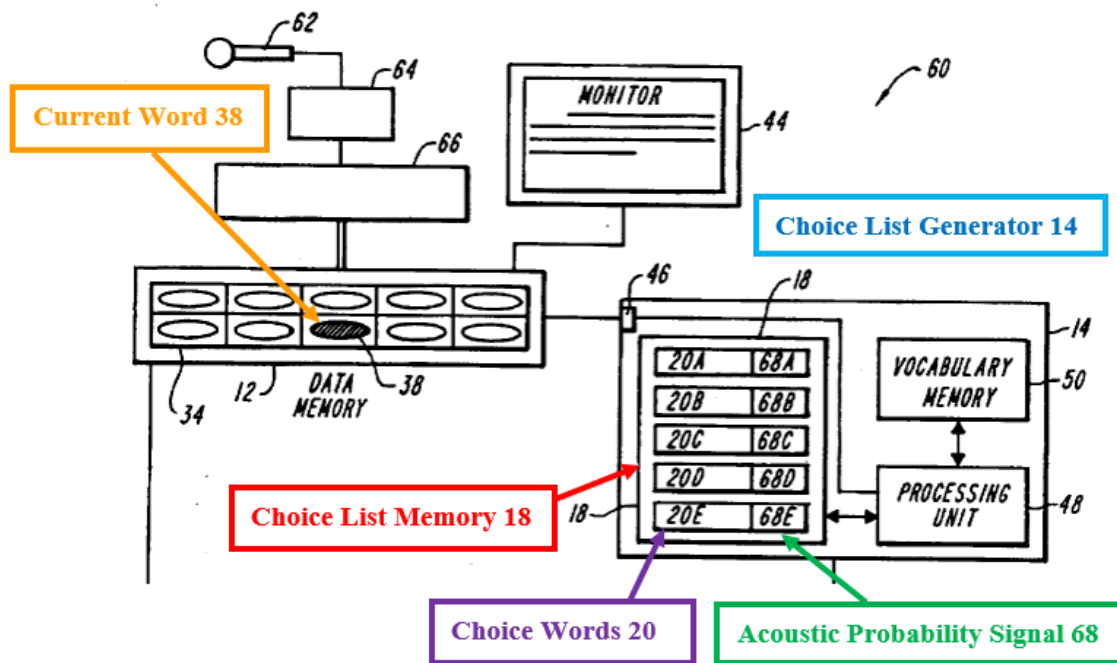
Baker does. Baker discloses methods for “speech recognition, optical character recognition, machine translation, and other recognition systems.” Ex. 1008, 6:29-32. Figs. 2, 3 (speech recognition system).

328. Baker describes systems and methods for “detect[ing] words within portions of speech signals and compar[ing] these detected words to acoustic models of known words, or sequence of acoustic models of known words.” Ex. 1008, 18:13-16.² “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.” Ex. 1008, 18:16-19. Words with an acoustic probability signal indicating a “high degree of certainty” are deemed recognized (referred to as “context words”). Ex. 1008, 18:16-19. However, words with a “low degree of certainty” indicate that the word has not been confidently identified (referred to as the “current word”). Ex. 1008, 18:22-24.

329. For those “unrecognized words,” Baker creates a “choice list **18**” that “store[s] the words **20** associated with the acoustic models from the vocabulary

² Notably, Baker defines the term “word” to mean “a word or a phrase and may be spoken, written, or represented as electronic data signals.” Ex. 1008, 6:1-3. Accordingly, Baker’s use of the term “word” also encompasses “phrases.”

memory 50 which are acoustically similar to the current ambiguous word 38 being identified, as illustrated below in the following excerpt of Figure 2:



Ex. 1008, Fig. 2 (excerpted and annotated)

330. Baker explains that each “choice word 20A-E” has a corresponding “acoustic probability signals 68A-E” that represent “the degree of certainty that the word 20 associated with the acoustic model is substantially representative of the current word 38 being recognized.” Ex. 1008, 18:30-35.

331. Baker further teaches that probability signals 68A-E “can represent the acoustic model probability adjusted by an n-gram probability, or any suitable language model.” Ex. 1008, 18:35-37, 20:30-46 (“The probability signals 68A-68E

can represent the likelihood that the associated word choice 20A-20E is substantially representative of the current word 38 being recognized.”), 22:55-64.

332. In my opinion, a POSITA would understand that Baker’s “choice list” stored in memory *maintains a set of prior probabilities or fuzzy possibilities* that is used to recognize words or phrases in the received utterance.”

333. Baker’s “choice list 20” is also *dynamic* as it is created by the “choice list generator 14” in response to the input speech by “speech analysis unit 66” that is used for “analyzing the speech signals and for recording with a certain degree of probability, the words that make up the speech signals.” Ex. 1008, 18:1-4, 19:53-20:29 (explaining how the “list of choice words 20” is generated by the “continuous speech utterance”).

334. Baker discloses this limitation in another way. Specifically, once a “choice word 20” has been identified, Baker teaches generating a “word list” of words that “have a semantic association with the choice word 20.” Ex. 1008, 14:57-69.

335. The “semantic association” for each choice word (which Baker also refers to as “passage words” (labeled as 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.” Ex. 1008, 20:55-58 (“The word list

generator 16 can access via a transmission path the reference database 40 and can identify information in the reference database 40 that relate to the topic, theme, or subject of a choice word 20.”).

336. Baker discloses using a data processor to “perform[] 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”). Ex. 1008, 21:19-47.

337. By looking at the “relative frequency of matches between the “context words” and the “passage words,” the data processor calculates a “probability signal 28 representative of the likelihood that the choice word or phrase 20 substantially represents the utterance currently being recognized.” Ex. 1008, 21:19-47, 21:49-22:6, Figs, 2-4.

338. Baker’s data processor then generates a “rank signal 30” that “represents the likelihood...that a choice word 20 is substantially similar to” the word being recognized” as a function of “context matching and acoustic model matching” probabilities. Ex. 1008, 23:22-27.

339. A table correlating each “choice word 20” (words selected based on acoustic-matching) with context-matching “probability signal 28” and “rank signal 30” is shown below in an annotated excerpt of Figure 2:

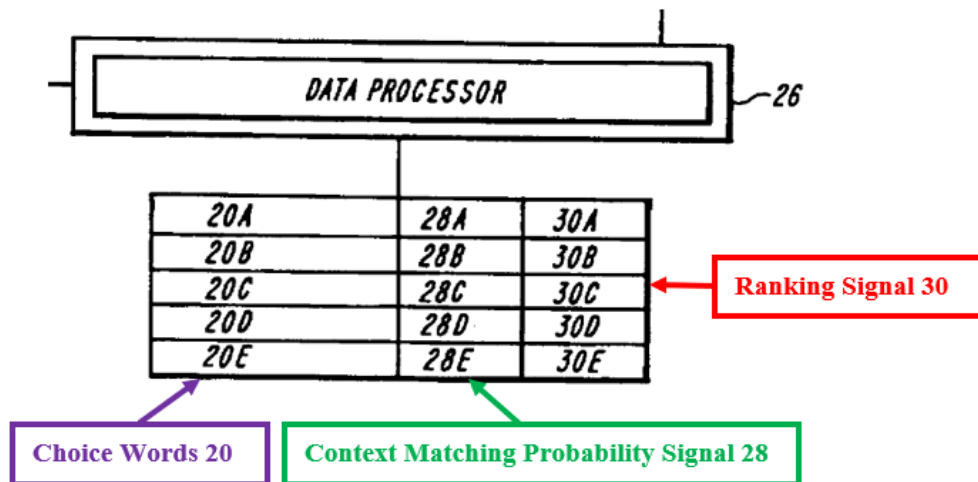


Fig. 2 (excerpted and annotated),

340. Baker's data processor 26 thus *maintains a set of prior probabilities* that include probability signals 28 and ranking signal 30 in memory. Ex. 1008, 21:62-65.

341. A POSITA would also understand that both the set of context-matching probability signals 28 and the set of ranking signals 30 associated with each choice word 20 are *dynamic* because they are generated "on the fly" after receiving the speech input. Ex. 1008, 4:64-5:8.

342. In my opinion, a POSITA would understand that Baker *maintains* at least three *dynamic sets prior probabilities* that it uses in "recognizing" step 1[c]: (1) a prior probability 68 based on an "acoustic model," (2) a prior probability based on context 28, (3) and a ranking signal 30, that is a probability signal as a function of signals 68 and 28.

343. ***Second***, to the extent this limitation applies to the “parsing” step 1[d], Hartono in view of Baker discloses this limitation.

344. As explained above, in the proposed combination, Baker’s output is the recognized words or series of words of a spoken utterance “just as if the data were being typed.” Ex. 1008, 1:4-7, 1:32-36. Those recognized words or series of words are then inputted into Hartono’s natural language “syntactic parser” and “semantic interpreter.” Ex. 1006, 11:14-16:11, Fig. 2. Accordingly, Hartono in view of Baker maintains the same set prior probabilities for step 1[c] as it does for step 1[d].

345. Moreover, Hartono alone discloses this limitation as applied to “parsing” step 1[d]. As explained further below in Section XIII.C.5, Hartono’s “natural language syntactic parser 102b” and “natural language semantic interpreter 102c” perform the “parsing” step 1[d].

346. Hartono discloses that after the natural language utterance has been syntactically parsed, its “semantic interpretation engine” applies “context sensitive methodologies” to invoke the appropriate application. Ex. 1006, 13:13-21. To do so, Hartono’s teaches using a “reverse grammar generation mechanism” that maintains “a list or vector for each word and corresponding probabilities for each word in the list.” Ex. 1006, 14:10-20.

347. For example, Hartono describes how the “associated vector or list” that includes the words “‘I,’ ‘eye,’ or ‘aye,’” have “corresponding probabilities of 80%, 15%, and 5%.” Ex. 1006, 14:10-20.

348. A POSITA would understand Hartono’s probability list to be *maintaining a set of prior probabilities or fuzzy possibilities* as it applies to “parsing” step 1[d]. 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. Ex. 1006, 14:17-20.

349. Thus, it is my opinion that a POSITA would have understood that Hartono in view of Baker discloses *maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance*.

4. Limitation 1[c]: “recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;”

350. In my opinion, Hartono and Baker disclose the limitation.

351. Hartono alone does not disclose this limitation. Hartono, however, recognizes that “[a]utomatic speech recognition systems for speech input are readily and commercially available off the shelf. Any suitable off the shelf speech recognition systems may be used as the automatic speech recognition system 102a in the natural language interface system 100 of the present invention.” Ex. 1006,

11:4-12. Hartono combined with the speech recognition system of Baker does disclose this limitation.

352. As discussed above in Sections VIII.E (Overview of Baker) XIII.3 (limitation 1[b]), Baker teaches generating a “word list” of words that “have a semantic association with the choice word 20.” Ex. 1008, 14:57-69. The “semantic association” for each choice word (which Baker also refers to as “passage words” (labeled as item 24 in Fig. 2) in the “word list” is derived from a “reference database 40” which “can be electronic memory that stores passages of text in an addressable format which typically includes a relevance indicator, such as a dictionary head word, that provides entry point information for the word list generator 16.” Ex. 1008, 9:18-24.

353. Baker provides several examples of “reference database 40,” including “dictionary, thesaurus, antonym dictionary, medical dictionary, legal dictionary, scientific and engineering dictionary, or other type of reference source.” Ex. 1008, 20:55-58 (“The word list generator 16 can access via a transmission path the reference database 40 and can identify information in the reference database 40 that relate to the topic, theme, or subject of a choice word 20.”), 15:50-8, 15:16-31, 21:6-12.

354. 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, with each passage indexed by head word.” Ex. 1008, 15:16-20, 15:32-37 (using passages from “Grolier’s Encyclopedia” as an on-line reference database), 16:8-12 (Encyclopedia Britannica).

355. A POSITA would understand that “discrete passages of text” that are “indexed by a head word” is a phrase table as claimed in the ’209 patent.

356. As discussed in Section VIII.E, Baker *uses information in one or more dictionary and phrase tables to recognize words and phrases contained in the received utterance*. 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 relevant passages “to determine the likelihood that the choice phrase represents the utterance being analyzed.” Ex. 1008, 21:19-47, 21:49-22:6.

357. Thus, it is my opinion that a POSITA would have understood that Hartono in view of Baker discloses *recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables*.

5. **Limitation 1[d]: “parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases;”**

358. In my opinion, Hartono discloses the limitation.

359. Hartono's "natural language syntactic parser 102b" and "natural language semantic interpreter 102c" perform the "parsing" step 1[d]. Hartono discloses first syntactically parsing the natural language input into a structure using a Government and Binding approach (a computation method based on linguistic theory). Ex. 1006, 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. Ex. 1006, 13:8-10.

360. Hartono's semantic interpreter then applies "context sensitive methodologies" to the syntactic parse to *determine a content for the at least one request contained in the utterance*. Ex. 1006, 13:13-21. Specifically, Hartono "takes the syntactic parse of the spoken language request and maps it to a generic concept frame," such as that shown below 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

Ex. 1006, 13:18-21, Table I, Table III

361. The semantic interpreter uses *keywords contained in the recognized words or phrases* to determine a context, by using verbs in the recognized words or phrases as "action-concepts" and nouns that "describ[e] the objects on which the

actions are performed” as “topic-concepts.” Ex. 1006, 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.”).

362. The determined context is shown in Table I as the “Application-Context.” Ex. 1006, Table I, 6:2-5 (“The natural language interface parses the user’s input and semantically maps it into a knowledge concept structure. The system then determines which application context should be responsible for interpreting and executing that command concept.”).

363. Thus, it is my opinion that a POSITA would have understood that Hartono discloses *parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases.*

6. Limitation 1[e]: “selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context;”

364. In my opinion, Hartono discloses the limitation.

365. The ’209 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.” Ex. 1001, 2:50-59. A person of ordinary skill in the art would thus understand a “domain agent” to be a software module specific to an external application.

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

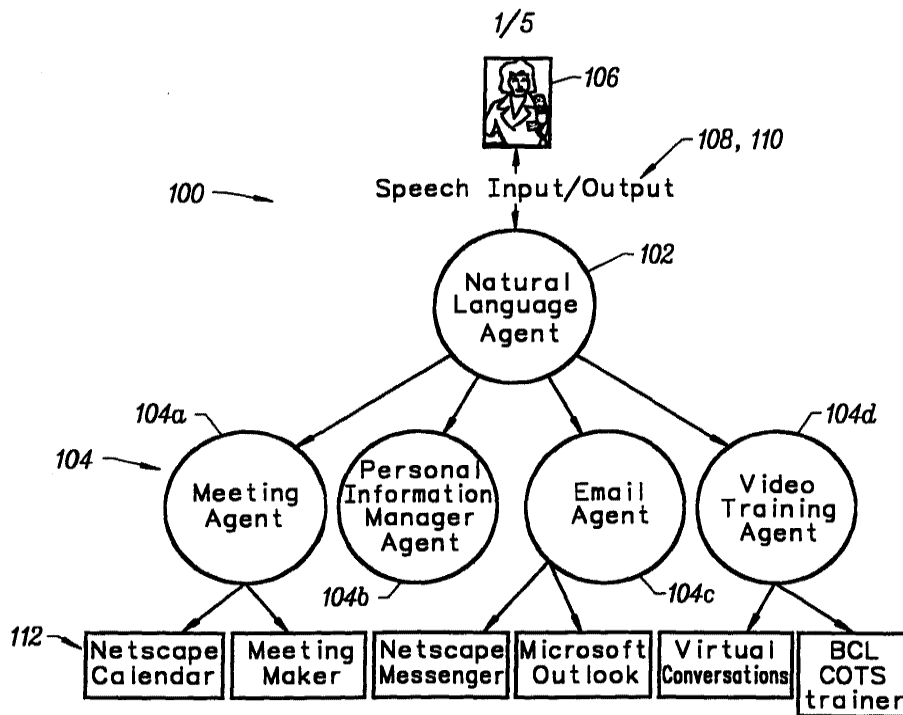


FIG. 1

Ex. 1006, Fig 1

367. A POSITA would understand that Hartono’s “task agents” (also referred in Hartono as “application class agents”) correspond to the claimed *domain agent* in the '209 patent because they are software modules specific to an application

or application area. Ex. 1006, 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”).

368. Hartono’s task agents are *autonomous executables* as each agent “includes a process-specific execution module for each application.” Ex. 1006, 30:15-20 (“For example, the word processing task agent may include an execution module for MICROSOFT WORD”), 36:7-9 (“application agent...[is] adapted to output an executable instruction to an application program”).

369. Hartono further teaches using a “task interpretation engine” to *select at least one domain agent based on the determined meaning*, determined by the semantic interpreter. Ex. 1006, 24:21-25:3 (“The task interpretation engine 104B receives the semantic frame representation as input. Based on the frame’s head verb (action request) and noun phrases (parameters), the task interpretation engine 104B invokes a routine that sends a set of requests to the task application wrapper 104A.”); *see* Hartono, 15:13-15 (“A combination of the action-concept and the topic-concepts are used to determine which task agent should handle the request. If the request is for a specialized task agent, the request is routed to that specialized agent.”).

370. Thus, it is my opinion that a POSITA would have understood that Hartono discloses *selecting at least one domain agent based on the determined*

meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context.

7. Limitation 1[f]: “formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;”

371. In my opinion, Hartono discloses the limitation.

372. Hartono discloses that each task agent includes a “wrapper [that] is preferably written in one of the platform specific macro languages,” and is “configured to communicate with a corresponding application program.” Hartono 24:14-16, Table II (showing platform specific macro languages); *see* Ex. 1006, 5:4-8.

373. The application wrapper “translates the conceptual action to a set of application specific operations.” Ex. 1006, 24:4-13, 6:6-7 (“The system utilizes task application wrappers to map the complex application tasks to vendor-specific executable tasks.”), 24:21-25:3 (“The task interpretation engine [that serves as the knowledge base for each task agent] receives the semantic frame representation as input. Based on the frame’s head verb (action request) and noun phrases (parameters), the task interpretation engine 104B invokes a routine that sends a set of requests to the task application wrapper.”).

374. A POSITA would understand that Hartono's task agent that translates the parsed utterance into a set of application specific operations to be executed by the application therefore teaches *formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context.*

375. Thus, it is my opinion that a POSITA would have understood that Hartono discloses *formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context.*

8. Limitation 1[g]: “invoking the selected domain agent to process the formulated request; and.”

376. In my opinion, Hartono discloses the limitation.

377. Hartono discloses that its task agents are *invoked* to output an executable instruction to an application program (*process the formulated request*). Ex. 1006, Abstract (“[A]t least one application agent [is] adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program.”), 4:20-25, 5:4-15, 5:21-25 (“[O]utputting the output command to an application class agent, semantically interpreting the output command and generating an executable instruction from the output command, and outputting the executable instruction to an application program for execution by the application.”), 36:7-9 (Hartono's “application agent”

is “adapted to receive and further interpret the output command from the natural language agent and adapted to output an executable instruction to an application program.”).

378. Thus, it is my opinion that a POSITA would have understood that Hartono discloses *invoking the selected domain agent to process the formulated request*.

9. Limitation 1[h]: “presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.”

379. In my opinion, Hartono discloses the limitation.

380. Hartono teaches that its adaptive natural language interface “outputs results and responses” to natural language commands that “allow[] users to control multiple desktop applications by abstract commands.” Ex. 1006, 6:1-9.

381. Hartono teaches several different applications that are used to execute the natural language input *generated as a result of the invoked domain agent*, 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*.” Ex. 1006, 44:1-13.

382. 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. Ex. 1006, 5:23-24 (recognizing that an

application program executes an executable instruction from an “application class agent”).

383. Thus, it is my opinion that a POSITA would have understood that Hartono discloses *presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.*

D. Dependent Claims

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

384. In my opinion, Hartono in view of Baker renders obvious this limitation for the same reasons as claim 1, in addition to the below limitations 4[a]-4[c].

a. Limitation 4[a]: “matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords, the dynamically updated set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities”

385. In my opinion, Ross in view of Ittycheriah renders obvious this limitation for the same reasons as claim 1, in addition to the below limitations 4[a]-4[c].

386. As discussed above with respect to limitations 1[b] and 1[d], Hartono *maintains a dynamic set of prior probabilities or fuzzy possibilities* that is usable to *determine the context* for the at least one request from the recognized words and

phrases. Sections XIII.C.3, XIII.C.5. Hartono discloses how its reverse grammar generation mechanism (that is part of its *parser*) “includes a list or vector for each word and corresponding probabilities for each word in the list.” Ex. 1006, 14:13-15. For example, Hartono describes how the “associated vector or list” that includes the words “‘I,’ ‘eye,’ or ‘aye,’” have “corresponding probabilities of 80%, 15%, and 5%.” Ex. 1006, 14:10-20.

387. A person of ordinary skill in the art would also understand that Hartono’s word list is *dynamic*, as the user can select what words are to be used. Ex. 1006, 14:17-20 (“These probabilities may be predetermined and may adjusted depending upon each user’s selection of words used or depending upon a subset or all of the users’ selection of words used.”);

388. Hartono further explains that “[u]pon receiving the syntactic parse of the spoken language request, the semantic interpretation engine 102c determines the permutations of the syntactic parse using the list for each word.” Ex. 1006, 14:21-15:1. By determining the different permutations, it is my opinion that Hartono *matches the keywords contained in the recognized words or phrases to the dynamically updated set of keywords.*

b. Limitation 4[b]: “scoring one or more possible contexts for the matched keywords using one or more of a profile of the user, data content associated with the domain

**agents, or a context stack of one or more recent contexts;
and”**

389. In my opinion, Hartono in view of Baker renders obvious this limitation.

390. As I discussed for limitation 1[b] and 1[d], Hartono’s “natural language syntactic parser 102b” and “natural language semantic interpreter 102c” (*parser*) receive the recognized words (*received keywords*) and probability and rank signals (*associated prior probabilities or fuzzy possibilities*) from speech recognition engine (*system agent*) of Baker. See Section XIII.C.3 (Ground 3, limitation 1[b]); Section XIII.C.5 (Ground 3, limitation 1[d]).

391. Upon receiving the recognized words, Hartono then *determin[es]...a score for each of at least two possible contexts* for the *natural language speech utterance* by parsing through the recognized words to determine the best fit for the syntactic parse.

392. Hartono discloses syntactically parsing the recognized word into a structure using a Government and Binding approach (a computation method based on linguistic theory). Ex. 1006, 12:1-20.

393. 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.” Ex. 1006, 13:8-10.

394. Hartono's semantic interpreter then applies "context sensitive methodologies" to the syntactic parse to *determine a score for each of at least two possible contexts*. Ex. 1006, 13:13-21.

395. Specifically, Hartono's semantic interpreter uses a "reverse grammar generation mechanism" that maintains "a list or vector for each word and corresponding probabilities for each word in the list." Ex. 1006, 14:10-20. This "associated vector or list," for example, includes the words "'I,' 'eye,' or 'aye,'" and has "corresponding probabilities of 80%, 15%, and 5%", resulting in different permutations of the natural language utterance, scored from highest to lowest probability. Ex. 1006, 14:10-20. Consequently, the *possible contexts* are "I", "eye, or "aye."

396. Hartono uses these different probability-based permutations to "determine[] which words best fit the grammar of the syntactic parse," which are *determined scores for possible contexts*. Ex. 1006, 15:1-12.

397. A person of ordinary skill in the art would have understood that to determine "best fit," the system would need to assign a value or score to the evaluated grammar of the syntactic parse to determine whether or not the word is "suitable given the context." To dig further into the case of the example presented by Hartono, we know that Hartono proceeds through the permutations of the syntactic parse from the word with the highest probability to the word with the

lowest probability (*associated prior probabilities or fuzzy possibilities*), checking to see which alternatives best fit the grammar of the syntactic parse. It does not simply take the permutation with the highest probability – it must also fit the grammar.

398. In the context of an agent for dictating email messages using speech, the phrase “I [0.80] want to go home” may very well be represented in the grammar. However, the phrases “Eye [0.15] want to go home” and “Aye [0.05] want to go home” will not be in the grammar. Here we have also indicated the probability of the word in question within each syntactic parse.

399. Hartono’s system starts by testing “I [0.80] want to go home” because it has the highest probability and determines that in fact fits the grammar. In other words, the word “I” is suitable given the context, which is an email dictation agent. Neither of the other two permutations needs to be considered because neither of them makes much sense when dictating an email.

400. However, in a more complex case where there are multiple plausible permutations of the syntactic parse, we see more instructive behavior. For example, consider an interaction between a ticket selling agent and a user when the prompt from the selling agent has been “Would you like to receive a free concert ticket?”:

- (A) “I would like two [0.60]” or
- (B) “I would like to [0.40]”

In this case, there are two possible interpretations for the last word in the user's response, and the syntactic parser has determined that "two" has a higher probability.

So the permutation "I would like two [0.60]" is tested first.

401. But we can envision that in the case of this particular ticket selling agent, a user is not allowed to request more free tickets than they have been offered; they are only allowed to accept or decline the offer of a single free ticket. In this context, "two" is not suitable even though it has the highest probability – it is not represented in the grammar, so the overall score is low even though the probability of the word is high. So the semantic interpretation engine then turns to the second alternative, "I would like to [0.40]" and finds that it is suitable given the context – the score is suitably high enough that the system takes this as the interpretation of the user's utterance.

402. Also consider a situation where there can be two contexts, and the user makes the following utterance: "I need 8 ounces of honey." Perhaps the user has access to two agents who can take instructions: one for ordering groceries to be delivered, and one for online banking. Say that the syntactic parser returns the following two options for the final word:

- (A) "I need 8 ounces of money [0.60]" or
- (B) "I need 8 ounces of honey [0.40]"

Because “I need 8 ounces of money [0.60]” has the highest probability, the semantic interpretation engine tries it first, checking with the grammar for the online grocery agent, where the permutation is not considered suitable because the grocery agent does not arrange to deliver money. When checking with the online banking agent, the permutation is also not suitable because the agent’s grammar does not include the concept of “ounces” of money.

403. With both the grocery agent and the banking agent returning low scores, the semantic interpretation engine turns to the permutation with the lower probability: “I need 8 ounces of honey [0.40]” When checking with the banking agent, its grammar returns a low score because “honey” does not correspond to the currency or other monetary instruments it delivers to customers. Finally, when checking with the grocery agent, its grammar returns a suitable score because it is programmed to interpret “8 ounces of honey” as part of a plausible grocery order.

404. The suitability given the context is based on the recognized word and probability and rank signals (*wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities*). As Hartono teaches, “[i]f the word is not suitable given the context, the next word having each next highest probability is then evaluated and determined if that word is suitable given the context, until a suitable word is determined.” Ex. 1006, 15:8-

12. Consequently, Hartono will iterate through the recognized words received from Baker depending on the probabilities and ranks provided.

405. In addition, recognizing that “[r]elevance feedback is widely used for preference optimization,” Hartono discloses using an “adaptive preference manager 102e...associated with the natural language each agent 102 and with each user 106” that “learn[s] what default conditions are preferred by the user 106 by either monitoring the users[sic] action implicitly (i.e. observing in the background) and/or by being instructed explicitly by the user 106 on positive and/or negative preferences.” Ex. 1006, 17:23-18:5, 6:19-20 (“The system and method can be adapted to user preferences with feedback using active and passive relevance feedback techniques”).

406. It is my opinion that because Hartono is constantly monitoring a user’s actions, a POSITA would find it obvious to include information about the user collected by the adaptive preference manger (i.e., *one or more of a profile of the user*) to aid in determining whether a word is suitable.

407. In my opinion, Hartono in view of Baker renders obvious *determining, for the natural language speech utterance, the suitability (a score) for the grammar of the syntactic parse for the different permutations of the words (for each of at least two possible contexts)* based on the recognized words and probability and rank signals received from Baker (wherein the scores are determined based on the

received keyword and associated prior probabilities or fuzzy possibilities), and in which a person of ordinary skill in the art would understand Hartono would be motivated to use and to try *scoring using one or more of a profile of the user, data content associated with the domain agents, or a context stack of one or more recent contexts* based on Hartono's adaptive preference manager.

c. Limitation 4[c]: “determining the context by selecting one or more of the possible contexts having a score satisfying a predetermined confidence level.”

408. In my opinion, Hartono discloses this limitation.

409. By evaluating whether a word is sufficiently suitable for a given context, a person of ordinary skill in the art would understand that Hartono finds that the word satisfies a predetermined confidence level and thereby determines (and selects) the context. *See* Ex. 1006, 15:8-12 (“If the word is not suitable given the context, the next word having each next highest probability is then evaluated and determined if that word is suitable given the context, until a suitable word is determined.”).

410. More specifically, a person of ordinary skill in the art would understand that once the semantic interpreter has scored the words in the recognized natural language utterance, it then *determines the context by selecting one or more of the possible contexts* that have been determined to be suitable (*having a score satisfying a predetermined confidence level*) by “take[ing] the syntactic parse of the spoken

language request and map[ing] it to a generic concept frame,” such as that shown below 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

Ex. 1006, 13:18-21, Table I, Table III.

411. Namely, the semantic interpreter uses verbs in the recognized words or phrases as “action-concepts” and nouns that “describ[e] the objects on which the actions are performed” as “topic-concepts.” Ex. 1006, 14:1-7. A person of ordinary skill in the art would understand that Hartono “uses the combination of the action-concept and the topic-concepts” to “determine which task agent should handle the request.” Ex. 1006, 15:13-19; *id.*, 6:2-5 (“The natural language interface parses the user’s input and semantically maps it into a knowledge concept structure. The system then determines which application context should be responsible for interpreting and executing that command concept.”).

2. **Claim 6: “The method according to claim 1, wherein formulating the at least one request in accordance with the grammar used by the selected domain agents includes:”**

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

- a. Limitation 6[a]: “determining one or more required values and one or more optional values for a request in the grammar used by the selected domain agent”**

413. In my opinion, Hartono discloses this limitation.

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

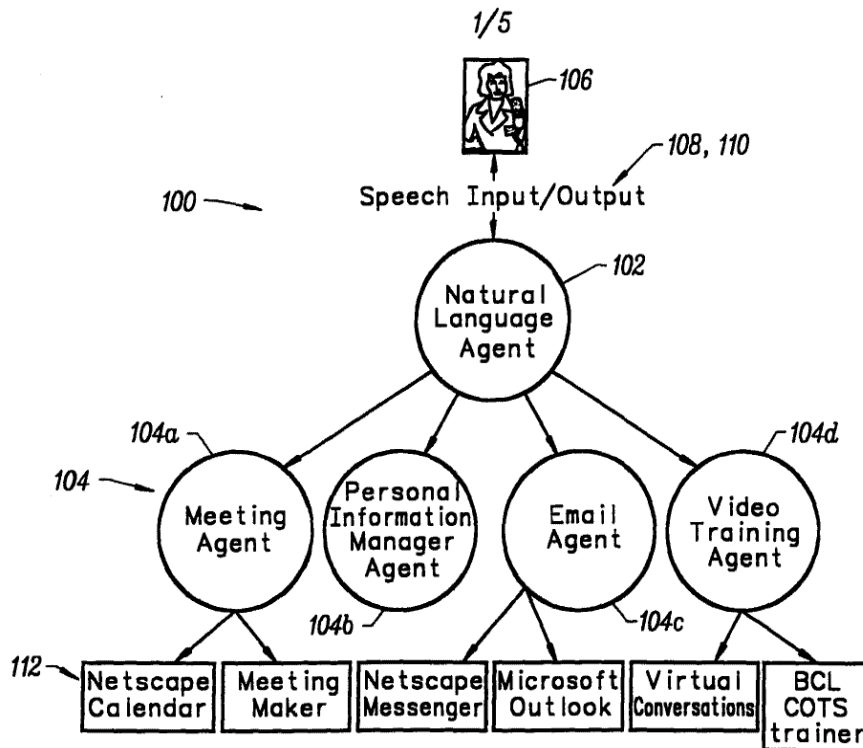


FIG. 1

Ex. 1006, Fig 1

Ex. 1006, 8:10-13, 9:9-14.

415. A person of ordinary skill in the art would understand that Hartono’s “task agents” (also referred in Hartono as “application class agents”) correspond to the claimed *domain agent* in the ’209 patent because they are software modules specific to an application or application area. Ex. 1006, 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”).

416. Hartono’s task agents are autonomous executables, as each agent “includes a process-specific execution module for each application.” Ex. 1006,

30:15-20 (“For example, the word processing task agent may include an execution module for MICROSOFT WORD”), 36:7-9 (“application agent...is adapted to output an executable instruction to an application program”).

417. Hartono further discloses a “natural language syntactic parser” and “natural language semantic interpreter” that are used to parse natural language commands into a computer action with noun phrases (NP) as the object and the adjective phrase (AP) as properties of the object, and then map the spoken language request “into a generic concept frame used to invoke the appropriate application method.” Ex. 1006, 12:11-10, 13:12-20.

418. 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”). Ex. 1006, 12:21-13:10. Hartono also 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. Ex. 1006, 30:21-31:5. Hartono’s also emphasizes the use of such optional elements. Optional elements can convey additional information to the dialog manager that could be useful in formulating a response to the user: for example, conveying a degree of exasperation on the part of the user. Hartono uses optional elements to

great effect in making “personality assessments” of the user to aid in responding.

Ex. 1006, 32:13-33:13.

419. A person of ordinary skill in the art would 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.

b. Limitation 6[b]: “extracting one or more criteria and one or more parameters from the keywords contained in the recognized words and phrases using procedures sensitive to the determined context;”

420. In my opinion, Hartono in view of Baker discloses this limitation.

421. '209 patent uses the terms “criteria,” “parameters” and “values,” as claimed in the '209 patent, somewhat interchangeably. For example, the '209 patent appears to consider “specific parameters and values” to also be criteria. Ex. 1001, 11:59-64 (“The system agent 150 manages the criteria handlers 152, which handle specific parameters or values (criteria) used to determine context for questions and commands.”), 14:10-16 (describing parameters as including “criteria”). However, the '209 patent lists examples of “criteria” for weather as the “location, date and time,” and for commands as “yes/no, on/off, pause/stop.” Ex. 1001, 18:26-28. It also describes “parameters” in a stock report as the stock’s “name and the company name.” Ex. 1001, 14:60-62.

422. Accordingly, a person of ordinary skill in the art would understand that “criteria” refers to the attributes of a request (e.g., location) and a parameter acts as the value of that attribute (e.g., Alexandria or Virginia).

423. Under any reasonable construction, however, Hartono’s semantic interpreter then 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*. Ex. 1006, 13:13-21.

424. Specifically, Hartono “takes the syntactic parse of the spoken language request and maps it to a generic concept frame,” thus extracting the various criteria and parameters of the request, such as that shown below 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

Ex. 1006, 13:18-21, Table I, Table III

425. 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.” Ex. 1006, 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.” Ex. 1006, Table I, 6:2-5 (“The natural language interface parses the user’s input and semantically maps it into a knowledge concept structure. The system then determines which application context should be responsible for interpreting and executing that command concept.”).

c. Limitation 6[c]: “inferring one or more criteria and one or more parameters using the dynamic set of prior probabilities or fuzzy possibilities; and”

426. In my opinion, Hartono discloses this limitation.

427. A person of ordinary skill in the art would understand Hartono’s probability list to be a *set of prior probabilities or fuzzy possibilities*. Namely, 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.” Ex. 1006, 14:10-20. For example, Hartono describes how the “associated vector or list” that includes the words “‘I,’ ‘eye,’ or ‘aye,’” have “corresponding probabilities of 80%, 15%, and 5%.” Ex. 1006, 14:10-20.

428. 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. Ex. 1006, 14:17-20.

429. 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*. As Hartono explains, its reverse generation grammar mechanism uses various permutations of the syntactic parse to “determine[] which words best fit the grammar of the syntactic parse.” Ex. 1006, 15:4-4.

430. 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.” Ex. 1006, 15:5-12. By using the prior probabilities to determine the best fitting word, a person of ordinary skill in the art 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.

d. Limitation 6[d]: “transforming the extracted and the inferred criteria and parameters into a format compatible with the grammar used by the selected domain agent, the transformed criteria and parameters including all of the required values and one or more of the optional values.”

431. In my opinion, Hartono discloses this limitation.

432. A person of ordinary skill in the art would understand that Hartono’s task agent that, using the task application wrapper, translates the parsed utterance into a set of application specific operations to be executed by the application

therefore 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.*

433. 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.” Ex. 1006, 24:14-16, Table II (showing platform specific macro languages); *see* Ex. 1006, 5:4-8. The application wrapper “translates the conceptual action to a set of application specific operations.” Ex. 1006, 24:4-13, 6:6-7 (“The system utilizes task application wrappers to map the complex application tasks to vendor-specific executable tasks.”), 24:21-25:3 (“The task interpretation engine [that serves as the knowledge base for each task agent] receives the semantic frame representation as input. Based on the frame’s head verb (action request) and noun phrases (parameters), the task interpretation engine 104B invokes a routine that sends a set of requests to the task application wrapper.”).

434. Hartono’s “application specific operations” to be to be processed by an external application with the criteria and parameters of the request would be understood by a person of ordinary skill in the art to be a collection of tokens, where each token is an individuated or non-reducible text element of data.

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

435. In my opinion, Hartono combined with Baker disclose this limitation for the same reasons as claim 1. *See* Section XIII.C. 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). Ex. 1006, 29:2-6.

a. Limitation 7[a]: “directing the at least one command to local or remote processing devices; and”

436. In my opinion, Hartono discloses this limitation.

437. Hartono provides an example of a request contained *in the utterance* that includes *at least one command*. Hartono describes the request “can you create a letter for my friend, John Smith?,” which includes both a question (can you create?) and a command (create a letter to John Smith). Ex. 1006, 29:2-6.

438. As set forth below in limitations 7[a] and 7[b], Hartono discloses that its invoked application class agent (*domain agent*) processes the formulated request. Hartono further discloses that its semantic mapper “maps the user the input to a specific action within the small set of semantic actions 142,” by “draf[ing] a letter to John Smith and the same task is performed.” Ex. 1006, 29:2-30:6, 30:12-14. A

person of ordinary skill in the art would understand that for the application class agent (*domain agent*) in Hartono to perform the command on a computer, it must be *directed to a local or remote processing device*. Ex. 1006, 33:21-35:3 (describing Hartono's implementation on a computer that includes a multi-processor system).

b. Limitation 7[b]: “querying local or network information sources to resolve information requested in the at least one question.”

439. In my opinion, Hartono discloses this limitation.

440. Hartono discloses that its request application class agents, or task agents, depending on the requested question *will query local or network information sources to resolve information requested*. Ex. 1006, 29 (Table IV), 39:1-13 (describing an application agent for each of the classes of application programs that include local information sources (such as addresses) and network information sources (such as internet retail sales or internet-based meeting scheduler). For example, Table IV shows how certain task agents (*domain agents*) will locate information requested in a question over the Internet or local information sources, such as an address book:

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 7,398,209

TABLE IV	
TASK AGENT	SAMPLE TASKS
Meeting	Arranges and schedules meetings
Scheduler	Schedules tasks on the computer
Telephone	Dials out and receives calls; integrates with address book
Voice Mail	Sends, receives, plays, manages voicemail
Word Processor	Writes, prints, manipulates, formats documents
Spreadsheet	Writes, prints, manipulates, formats numerical data
Drawing	Draws, manipulates, formats diagrams, merges predrawn images
Web	Connects, navigates, searches the internet/world-wide web
Network	Connects networks, manages connections
Mathematical	Manages mathematical and scientific manipulation of numeric and formulaic data
Directory Assistance	Locates telephone numbers and addresses over the internet
Internet Retail Sales	Describes objects and sells to customers over the internet
Common Household Utility Agent (e.g., VCR, Toaster, HVAC)	Controls household devices
K-12 Education on, e.g. Physics, Chemistry, Math	Teaches, runs games, quizzes, etc. for mathematics and science subjects
General Education on History, Economics, Philosophy	Teaches specific liberal arts and humanities courses
Hands-on training for job-based tasks	Trains users to operate equipment
Internet Event lookup	Locates events, such as conferences, meetings, concerts and festivals, through the internet
Internet Product-Information lookup	Finds products and prices through the internet
Internet-based Meeting scheduler	Schedules meetings over the internet
Hardware manager	Manager Computer Hardware (screen, disk, etc.)

Ex. 1006, 29 (Table IV).

XIV. OPINIONS ON GROUND 5:

A. A POSITA Would Have Been Motivated to Combine Hartono, Ittycheriah, and Aridor

441. A person of ordinary skill in the art would have been motivated to add Aridor's "knowledge agents" that perform meta-searching over the Internet to Hartono's application class agents (*domain agents*) to collect information from a large body of documents (like the Internet) and to filter and rank the results based on relevance. Although Hartono describes that its domain agents are able to find information on the Internet, Hartono does not detail how it performs such searches. Ex. 1006, 29 (Table IV), 39:1-13.

442. It is therefore my opinion that a person of ordinary skill in the art would have been motivated to look for and rely on Internet searching mechanism to aid in retrieving and processing responses to a user's queries (such as those disclosed in Aridor).

443. A person of ordinary skill in the art would also understand that Aridor's metasearch functionality would be compatible with Hartono, as Aridor operates by using separate knowledge agents based on domain, such that each of Hartono's application class agents could have its own knowledge agent specific to their domain. 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 select the domain

agent as set forth in limitations 1[e], 1[f], and 1[g] (Section XII.C.6-8). Ex. 1010, 3:53-59.

444. A person of ordinary skill in the art would further be motivated to add to Hartono the “ranking and filtering” features of Aridor that process the results of the user’s query (natural language question or command) to find the most relevant information.

445. In addition, a person of ordinary skill in the art 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. Ex. 1010, 3:65-4:2, 14:56-15:19.

B. A POSITA Would Have Had a Reasonable Expectation of Success

446. In my opinion, a person of ordinary skill in the art would have had a reasonable expectation of success.

447. Aridor expands on Hartono’s ability to process natural language commands and questions by detailing how a metasearch over the Internet could be used to find the most relevant results. Indeed, adding Aridor to Hartono would be as simple as adding another task agent to its system—a feature Hartono is designed to do and accept.

448. A person of ordinary skill in the art 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 (e.g., known systems for speech recognition and Internet searching based on text queries) according to known methods (e.g., known methods of 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.

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

449. In my opinion, Hartono and Ittycheriah in view of Aridor renders this claim obvious.

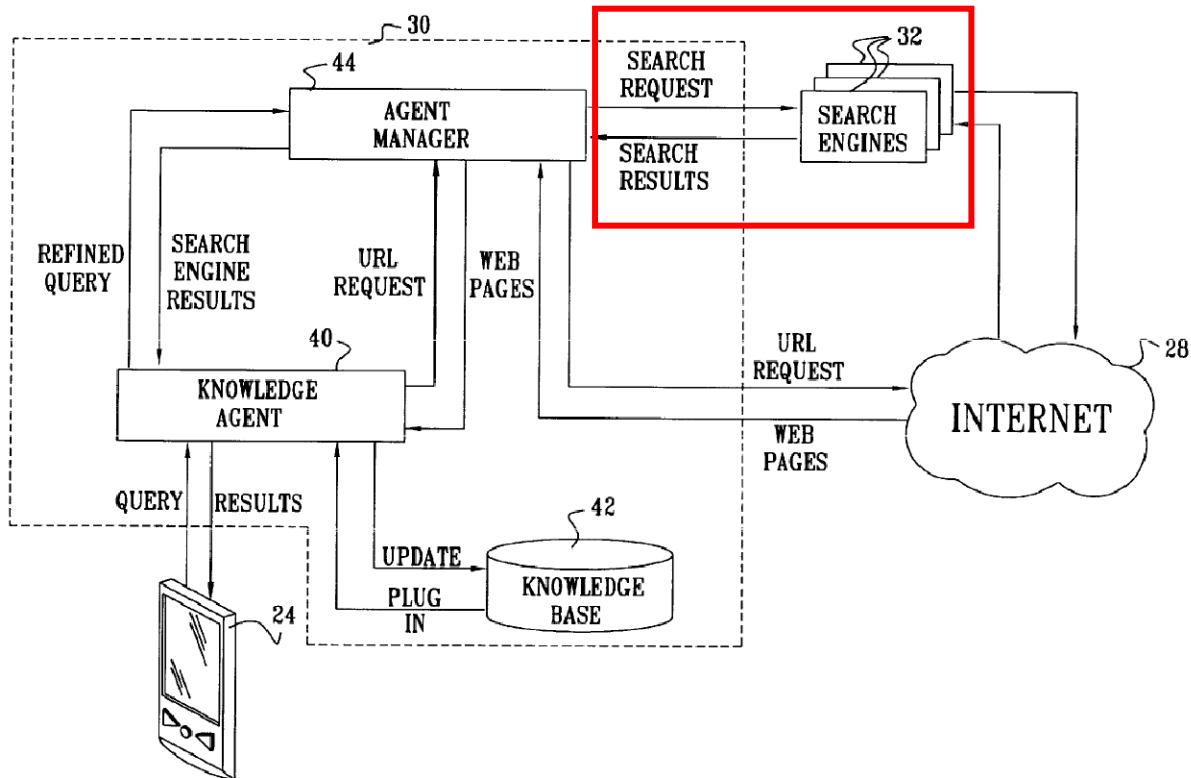
1. Limitation 8[a]: “submitting a plurality of asynchronous queries to the information sources, the plurality of queries including a plurality of duplicate queries submitted to distinct information sources;”

450. In my opinion, Hartono in combination with Aridor discloses this limitation. As discussed above with respect to Motivation to Combine, Aridor’s metasearch engine that utilizes knowledge agents is one of Hartono’s application task agents (*domain agents*).

451. Aridor discloses that its knowledge agents and agent manager “receive the user’s query and carries out the search by simulating the steps involved in the conventional interactive search process.” Ex. 1010, 3:33-35. In performing the search, Aridor’s agent manager and knowledge agent refines a query and *submits a plurality of asynchronous queries to the information sources*. Ex. 1010, 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 specified by user 22, such as AltaVista, Google or Hotbot[.]”), Fig. 2.

452. Aridor further discloses that knowledge agent and agent manager “passes the refined query to a number of search engines.” Ex. 1010, 4:10-16, 8:22-27 (“A knowledge agent program...is responsible for carrying out the search for user 22. 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.”). This is also shown in Figure 2 of Aridor, in which the same search request is submitted to multiple search engines:

FIG. 2



Ex. 1010, Fig. 2 (annotated).

453. A person of ordinary skill in the art would understand that a metasearch that sends the same search request multiple times to different search queries are *duplicate queries submitted to distinct information sources*.

2. Limitation 8[b]: “a synchronously receiving a plurality of results of the queries from the information sources; and”

454. In my opinion, Hartono in combination with Aridor discloses this limitation.³

455. 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.” Ex. 1010, 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).

456. A person of ordinary skill in the art would understand that each user-selected search engine sends results *asynchronously*, as the results are sent independent of the operation of other search engines. 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. Ex. 1010, 8:52-59.

³ For the purposes of this Petition, I understand that Petitioner assumes this claim recites “asynchronously.”

3. Limitation 8[c]: “scoring a relevance of each of the asynchronously received results using the dynamic set of prior probabilities or fuzzy possibilities to determine one or more best responses to the at least one question”

457. In my opinion, Hartono in combination with Aridor discloses this limitation.

458. Ardior describes that once the results of the metasearch are received, they are returned to the knowledge agent for “ranking and filtering.” Ex. 1010, 8:56-57. Specifically, 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.” Ex. 1010, 11:19-34.

459. 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.” Ex. 1010, 11:19-12:67 (textual scores), 13:1-15:11 (topology scores), Fig. 3.

460. 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.” Ex. 1010, 11:32-34.

461. Aridor uses the *dynamic set of prior probabilities or fuzzy possibilities* to score the relevance of the results.

462. 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.” Ex. 1010, 11:60-12:13, 14:24-34.

463. “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 in returning results to the user. Ex. 1010, 14:44-55.

464. A person of ordinary skill in the art 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.

465. 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.” Ex. 1010, 14:35-43, 14:56-15:19.

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

1. Limitation 9[a]: “determining one or more tokens required to formulate a response to the question;”

466. In my opinion, Hartono discloses this limitation.

467. The '209 patent describes the “tokens” to formulate a response to be distinct pieces of information that are generally required to produce a response. Ex. 1001, 19:57-66. For example, “for a question on weather,” the tokens include “date, day of the week, predicted high temperature, predicted low temperature, chance of precipitation, expected cloud cover, expected type of precipitation,” among others. Ex. 1001, 19:59-67.

468. 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.”). Ex. 1010, 9:42-49. Specifically, Aridor discloses that it 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.” Ex. 1010, 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).

469. Such tokens are typical. For example, Hartono likewise discloses that a response to a question requires determining values for specific pieces of information (*tokens*). Hartono describes ordering a pizza, which requires tokens for “size” and “topping,” without such information causes the question to be incomplete. Ex. 1006, 21:23-22:3. For example, the request “I want a cheese pizza,” is incomplete because it lacks size information. Ex. 1006, 22:4-10.

2. Limitation 9[b]: “extracting one or more values for the determined tokens from the asynchronously received results, the one or more values extracted by applying one or more scraping criteria to the asynchronously received results;”

470. In my opinion, Hartono in view of Aridor discloses the limitation.

471. Aridor discloses how once it submits the query to “one or more general purpose search engines,” the results are collected from those search engines, which Aridor calls the “root set 54 of sites 26.” Ex. 1010, 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. Ex. 1010, 10:40-11:18.

472. With the results of the site list, Aridor *applies one or more scraping criteria* to the results to *extract one or more values for the determined tokens (the asynchronously received results)*. Aridor scrapes the “site list” results for both

textual similarity and for topological aspects, assigning scores for both features. Ex. 1010, 11:19-34, 11:46-51, 12:13-62, Fig. 3.

473. For textual similarity, Aridor scrapes the words in each page and divides the words into three grounds: “Strong text includes the words that appear in the title or in large font headers; medium text includes words that are either highlighted (bold, italics, etc.) or in small font headers; and all the rest is regular texts.” Ex. 1010, 11:46-51. Aridor then applies weights to each type of word (strong, medium, and regular) and computes a “query textual score,” and in a similar way, computes a “domain score” based on those documents in the knowledge base. Ex. 1010, 12:13-60. The “query textual score” and “domain score” are then “normalized and combined to create the overall textual similarity score 68 for each site[.]” Ex. 1010, 12:60-62.

474. For the topological aspects, Aridor 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 to surfers such 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 “is presumably answers the specific query.” Ex. 1010, 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*). Ex. 1010, 14:11-23.

3. Limitation 9[c]: “evaluating the extracted values using the dynamic set of prior probabilities or fuzzy possibilities, wherein evaluating the extracted values includes resolving ambiguous, incomplete, or conflicting information; and”

475. In my opinion, Hartono in view of Aridor discloses this limitation.

476. As discussed above, Aridor extracts values via a scraping of text and a scraping of links in the site list of potential results. Section XIV.D.2.b (Ground 4, limitation 6[b]). Further, as described above with respect to limitation 9[b], Aridor’s “textual score” and “link topology score” both use *the dynamic set of prior probabilities or fuzzy possibilities* in how computes its score (*evaluate[s] the extracted values*) using “persistent, acquired knowledge in a domain,” to improve the quality of the received results. *See* Section XIV.D.2.b; Ex. 1010, 3:53-4:11, 14:56-15:11.

477. 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.” Ex. 1010, 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.

478. A person of ordinary skill in the art 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*. In fact, a person of ordinary skill in the art would find this to be a very good way to resolve these issues and achieve high quality results.

4. Limitation 9[d]: “selecting one or more best values for the determined tokens from the evaluated values, wherein the one or more best responses include the one or more selected best values.”

479. In my opinion, Hartono in view of Aridor discloses this limitation.

480. As discussed above for limitation 6[d.ii.C.3], Aridor discloses that it 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.” Ex. 1010, 11:19-34.

481. By ranking and downloading (or otherwise sending to the user) 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.

XV. CONCLUSION

482. For all the reasons stated above, it is my opinion that the challenged claims of the '209 patent are unpatentable.

483. This declaration and my opinions are made to the best of my knowledge and understanding, and based on the material available to me at the time of signing this declaration. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true. Further, I am aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001. I declare under penalty of perjury that the foregoing is true and correct.

Executed this 18 day of August 2025, in Madrid, Spain.

A handwritten signature in black ink, appearing to read 'Dan Lopresti', with a horizontal line underneath it.

Daniel P. Lopresti, Ph.D

Appendix A

Exhibit No.	Description
1001	U.S. Patent No. 7,398,209 to Kennewick, et al. (“the ’209 patent”)
1002	File History of U.S. Patent No. 7,398,209
1003	Declaration of Daniel Lopresti (“Lopresti Decl.”)
1004	Curriculum Vitae of Daniel Lopresti
1005	U.S. Patent No. 7,085,723 to Ross et al. (“Ross”)
1006	WO 00/11571 to Hartono et al. (“Hartono”)
1007	U.S. Patent No. 5,937,383 to Ittycheriah et al. (“Ittycheriah”)
1008	U.S. Patent No. 5,680,511 to Baker et al. (“Baker”)
1009	U.S. Patent No. 7,249,018 to Ross et al. (“Ross018”)
1010	U.S. Patent No. 6,636,848 to Aridor et al. (“Aridor”)
1011	U.S. Patent Application Publication No. 2002/1065860 to Glover et al. (“Glover”)
1012	U.S. Patent No. 7,127,402 to Ross et al. (“Ross402”)
1013	Appendix A – Infringement Contentions for U.S. Patent No. 7,398,209, <i>Dialect LLC v. Meta Platforms, Inc.</i> , No. 7:25-CV-00060-DC-DTG (E.D. Tex. July 16, 2025)
1014	Reserved
1015	Reserved
1016	U.S. Provisional Patent Application No. 60/261372 to Ross et al.

Appendix A

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