

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

DIALECT, LLC,
Patent Owner.

IPR2025-01351
Patent: 7,398,209
Issued: July 8, 2008
Application No. 10/452,147

Filed: June 3, 2003

Title: SYSTEMS AND METHODS FOR RESPONDING
TO NATURAL LANGUAGE SPEECH UTTERANCE

DECLARATION OF PAUL JACOBS

TABLE OF CONTENTS

	Page
I. INTRODUCTION AND ENGAGEMENT	6
II. BACKGROUND AND QUALIFICATIONS	6
III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '209 PATENT.....	11
IV. UNDERSTANDING OF PATENT LAW	13
V. GROUNDS	17
VI. LEVEL OF SKILL IN THE ART AND STATE OF THE ART.....	18
A. Person Of Ordinary Skill In The Art.....	18
B. State Of The Art	18
VII. THE '209 PATENT.....	24
a. The '209 Patent's Specification	24
A. The Prosecution History.....	25
VIII. CLAIM CONSTRUCTION	27
A. Claim Construction Order For Related '006 Patent.....	28
IX. GROUND 1: CLAIMS 1, 2, 4-9, AND 12-14 ARE OBVIOUS OVER KHAN.....	28
A. Khan (EX1004)	29
1. Claim 1	32
a) Element [1.1]	32
a) Element [1.2]	33
b) Element [1.3]	34
c) Element [1.4]	38

d)	Element [1.5]	39
e)	Element [1.6]	42
f)	Element [1.7]	44
g)	Element [1.8]	47
h)	Element [1.9]	48
2.	Claim 2	52
a)	Element [2.1]	52
b)	Element [2.2]	52
c)	Element [2.3]	54
d)	Element [2.4]	55
3.	Claim 4	58
a)	Element [4.1]	58
b)	Element [4.2]	58
c)	Element [4.3]	60
d)	Element [4.4]	61
4.	Claim 5	63
a)	Element [5.1]	63
b)	Element [5.2]	63
c)	Element [5.3]	65
5.	Claim 6	65
a)	Element [6.1]	65
b)	Element [6.2]	66
c)	Element [6.3]	66

	d)	Element [6.4]	68
	e)	Element [6.5]	68
6.		Claim 7	71
	a)	Element [7.1]	71
	a)	Element [7.2]	72
	b)	Element [7.3]	73
	c)	Element [7.4]	75
7.		Claim 8	75
	a)	Element [8.1]	75
	b)	Element [8.2]	76
	c)	Element [8.3]	77
	d)	Element [8.4]	78
8.		Claim 9	81
	a)	Element [9.1]	81
	b)	Element [9.2]	81
	c)	Element [9.3]	82
	d)	Element [9.4]	85
	e)	Element [9.5]	87
9.		Claim 12	87
	a)	Element [12.1]	87
	b)	Element [12.2]	87
	c)	Element [12.3]	89
10.		Claim 13	90

a)	Element [13.1]	90
b)	Element [13.2]	90
c)	Element [13.3]	90
d)	Element [13.4]	93
e)	Element [13.5]	95
11.	Claim 14.....	97
a)	Element [14.1]	97
b)	Element [14.2]	97
X.	GROUND 2: CLAIMS 1,2, 4-9, AND 12-14 ARE OBVIOUS OVER KHAN IN VIEW OF RICCARDI.....	98
A.	Riccardi (EX1005)	99
B.	The Khan-Riccardi Combination	100
XI.	GROUND 3: CLAIMS 1,2, 4-9, AND 12-14 ARE OBVIOUS OVER KHAN IN VIEW OF RICCARDI AND IIZUKA	102
XII.	GROUND 4: CLAIMS 1,2, 4-9, AND 12-14 ARE OBVIOUS OVER KHAN IN VIEW OF RICCARDI, IIZUKA, AND HARTONO....	103

I, Paul Jacobs, do hereby declare as follows:

I. INTRODUCTION AND ENGAGEMENT

1. I have been retained as an independent expert on behalf of Microsoft Corporation in connection with the above-captioned Petition for *Inter Partes* Review (“IPR”) to provide my analyses and opinions on certain technical issues related to U.S. Patent No. 7,398,209 (hereinafter “the ’209 patent”).

2. Specifically, I have been asked to provide my opinions regarding whether claims 1, 2, 4-9, and 12-14 (each a “Challenged Claim” and collectively the “Challenged Claims”) of the ’209 patent would have been obvious to a person having ordinary skill in the art (“POSITA”) as of June 3, 2002. It is my opinion that each Challenged Claim would have been obvious to a person having ordinary skill in the art after reviewing the prior art discussed herein.

3. I am being compensated at my usual and customary rate for the time I spent in connection with this IPR. My compensation is not affected by the outcome of this IPR.

II. BACKGROUND AND QUALIFICATIONS

4. My name is Paul Jacobs, and I am over 21 years of age and otherwise competent to make this Declaration. I make this Declaration based on facts and matters within my own knowledge and on information provided to me by others,

and, if called as a witness, I could and would competently testify to the matters set forth herein.

5. I have summarized in this section my educational background, career history, and other qualifications relevant to this matter. A more complete recitation of my professional experience including a list of my journal publications, patents, conference proceedings, book authorship, and committee memberships may be found in my curriculum vitae, which is attached hereto as Appendix A.

6. I am an expert in natural language processing and have a broad background in computer science that I have applied for decades as a scientist in the field of natural language processing, as a technology leader and executive, as a consultant, a professor, and as an advisor regarding software patents.

7. I received a Bachelor of Science in Applied Mathematics from Harvard University in 1981, a Master of Science in Applied Mathematics from Harvard in 1981, and a Ph.D. in Computer Science from the University of California at Berkeley in 1985.

8. I have authored or co-authored over 50 scientific and technical publications, primarily in the fields of artificial intelligence, natural language processing, and information retrieval. I am listed as an inventor on two U.S. patents directed to computational lexicons, and I have over 40 years of experience in the computer and information industry.

9. I have served in numerous professional and scientific capacities, including one year as a visiting professor of computer science at the University of Pennsylvania and several years as a member of the executive committee of the Association for Computational Linguistics. Currently, I serve on the Technology Policy Committee of the Association for Computing Machinery (USTPC). As an adjunct lecturer, I taught classes in the College of Information Studies (The "iSchool") at the University of Maryland in College Park from 2007. While my specialty has always been in artificial intelligence and natural language processing, teaching at the graduate level kept me current in a broad base of technologies pertaining to speech recognition and natural language processing systems.

10. Between 1985 and 1994, after completing my doctorate, I was employed as a computer scientist with General Electric ("GE") Corporate Research and Development, where I worked on a broad range of speech recognition and natural language processing research and development, including text processing and written and spoken language interfaces. I also consulted for Infonautics, an early Internet information services and advanced search company. I was the editor of a book, entitled "Text-Based Intelligent Systems." The book was a collection of papers based on a symposium I chaired in 1990, which brought together leaders of the field of Information Retrieval to address issues related to large-scale advanced text

processing. My work foreshadowed subsequent advances in AI that captured the power of large quantities of real data to simulate human-like capabilities.

11. During my years at GE, I was principal investigator for the GE team in the Tipster program, sponsored by the Advanced Research Projects Agency (ARPA) of the United States Department of Defense and other government agencies. The technology developed in Tipster formed the foundation of the first web search engines, and my work as an ARPA principal investigator in natural language kept me in close contact with research in related fields, particularly spoken language recognition. For example, I served on the program committee for ARPA's Speech and Natural Language Conference.

12. I joined SRA International ("SRA") in the latter part of 1994 and became director of media information technologies. My responsibilities included new ventures and technology related to the Internet and the World Wide Web. From 1994 until 2002, as the Web came of age, I held a series of technology and business management jobs in organizations focused on networked information management applications. For example, from late 1999 through the end of 2000, I was president and then CTO of AnswerLogic, one of the first companies directed toward answering naturally-expressed user questions on the Web. During this time we also experimentally developed spoken-language versions of our technology.

13. From my initial academic training in artificial intelligence, which was supervised by Marvin Minsky at MIT's Artificial Intelligence Laboratory while I was completing my Master's work at Harvard, through my career in commercial and academic R&D and as a graduate instructor, I was closely involved in the advances in speech and language technology. I visited Verbex, the predecessor of Dragon Systems, while at Harvard, made multiple invited trips to Bell Laboratories and while at Berkeley regularly participated in seminars and collaboration with SRI, where I witnessed the birth of the company that later became Nuance, to which the '209 patent was assigned (the commercial heritage of Nuance is confusing, because the name belonged to another company called Scansoft that was in a different business, but Nuance's speech technology that was later used in Siri came from the SRI spinoff that was purchased by Nuance). As a DARPA principal investigator, I participated regularly in some of the first Speech and Language Workshops (sometimes known as Human Language Technology workshops) where we collaborated and tested against shared "benchmarks". As mentioned above, I later collaborated with Nuance and others in integrating question answering technology with speaker-independent spoken language recognition.

14. Based on my experiences described above, and as indicated in my Curriculum Vitae, I am qualified to provide the following opinions with respect to the patents in this case. While my extensive background in these areas is well beyond

the capabilities of one of ordinary skill—and was so in June 2002—I am qualified to opine on what one of ordinary skill would have known and understood as of the priority date of the '209 patent.

III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '209 PATENT

15. In preparing this declaration, I have reviewed the following materials bearing Exhibit Nos. that I understand are being referenced in the IPR to which my declaration accompanies:

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 7,398,209 (“the '209 patent”)
1002	File History of U.S. Patent No. 7,398,209
1004	International Patent App. No. WO 2001/080096A1 (“ Khan ”)
1005	U.S. Patent No. 8,392,188 (“ Riccardi ”)
1006	U.S. Patent No. 6,424,980 (“ Iizuka ”)
1007	International Patent App. Pub. No. WO 2000/011571A1 (“ Hartono ”)
1008	U.S. Patent No. 7,043,420 (“ Ratnaparkhi ”)
1009	U.S. Patent No. 7,761,296 (“ Bakis ”)
1010	Jurafsky, D., & Martin, J., <i>Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition</i> , Prentice-Hall, Inc. Publ. (2000) (excerpts) (“ Jurafsky ”)

No.	Description
1011	Huang, X., et al., <i>Spoken Language Processing – A Guide to Theory, Algorithm, and System Development</i> , Prentice-Hall, Inc. Publ. (2001) (excerpts) (“ Huang ”)
1012	<i>Nuance Speech Recognition System</i> , Version 7.0, Nuance Grammar Developer’s Guide (“ Nuance ”)
1013	U.S. Patent No. 3,466,394 (“ French ”)
1014	U.S. Patent No. 5,946,654 (“ Newman ”)
1015	U.S. Patent No. 7,092,928 (“ Elad ”)
1016	U.S. Patent No. 7,136,818 (“ Cosatto ”)
1017	U.S. Patent No. 8,402,068 (“ Clendinning ”)
1018	U.S. Patent No. 7,216,073 (“ Lavi ”)
1019	U.S. Patent No. 5,860,064 (“ Henton ”)
1020	U.S. Patent No. 5,970,453 (“ Sharman ”)
1021	U.S. Patent No. 5,689,697 (“ Edwards ”)
1022	De Mori, R., et al., <i>A Probabilistic Approach to Person-Robot Dialogue</i> , NATO ASI Series Vol. F75 (1992) (“ De Mori ”)
1023	U.S. Patent App. No. 2002/0032564 (“ Ehsani ”)
1024	U.S. Patent No. 5,027,406 (“ Roberts ”)
1025	U.S. Patent No. 6,262,730 (“ Horvitz ”)
1026	U.S. Patent App. No. 2002/0069194 (“ Robbins ”)
1027	U.S. Patent No. 6,144,938 (“ Surace ”)
1029	List of Challenged Claims – USP 7,398,209

No.	Description
1030	Jacobs, P., <i>Concretion: Assumption-Based Understanding</i> , Proceedings of the 12th International Conference on Computational Linguistics (1988) (“ Jacobs-Concretion ”)
1031	Jacobs, P., <i>Joining statistics with NLP for text categorization</i> , Proceedings of the Third Conference on Applied Natural Language Processing (1992) (“ Jacobs-Joining ”)
1032	U.S. Patent No. 5,652,897 (“ Linebarger ”)
1033	Jacobs, P., <i>TRUMP: A Transportable Language Understanding Program</i> , International Journal of Intelligent Systems (1992) (“ Jacobs-TRUMP ”)

IV. UNDERSTANDING OF PATENT LAW

16. I am not an attorney. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to my opinions. My understanding of the law was provided to me by the Petitioner’s attorneys.

17. I understand that when considering the scope of the claims of a patent that the patent claim terms should generally be given the ordinary meaning that the terms would have to a person of ordinary skill in the art in question after reading the patent as of the earliest claimed priority date. As discussed below, it is my opinion that the challenged claims are not entitled to the earliest priority date asserted by the ’209 patent. However, in my opinion, the meaning of the claim terms would not have materially differed to a person of ordinary skill in the art (“POSITA”) whether the

claims were evaluated at the time of the earliest claimed priority date June 3, 2002, or at the time the application that led to the '209 patent was filed June 3, 2003.

18. I understand that the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the term appears, but in the context of the entire patent, including the other claims and the specification of the application as filed. I further understand that the principal considerations regarding the scope and meaning of the claims are the plain language of the claim (including the surrounding claim language and context), the patent specification, and the prosecution history. I understand that while a claim is to be read in light of the specification, one must generally avoid importing limitations into the claim from the specification. I am also informed that the prosecution history can often inform the meaning of the claim by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be. I applied these understandings when considering the scope and meaning of the claims of the '209 patent.

19. I understand that a prior art reference anticipates an asserted claim, and thus renders the claim unpatentable, if all elements of the claim are disclosed in that prior art reference, either explicitly or inherently (i.e., necessarily present or implied).

20. I further understand that a claim is unpatentable if it would have been obvious. Obviousness of a claim requires that the claim would have been obvious from the perspective of one of ordinary skill at the time the alleged invention was made. I understand that a claim could have been obvious from a single prior art reference or from a combination of two or more prior art references. I understand that an obviousness analysis requires an understanding of the scope and content of the prior art, any differences between the alleged invention and the prior art, and the level of ordinary skill in evaluating the pertinent art.

21. I further understand that a claim would have been obvious if it unites old elements with no change to their respective functions, or merely substitutes one element for another known in the field, and that combination yields predictable results. While it may be helpful to identify a reason for this combination, I understand that there is no strict requirement of finding an express teaching, suggestion, or motivation to combine within the references. When a product is available, design incentives and other market forces can prompt variations of it, either in the same field or different one. If one of ordinary skill can implement a predictable variation, obviousness likely bars its patentability. For the same reason, if a technique has been used to improve one device and one of ordinary skill would recognize that it would improve similar devices in the same way, using the technique would have been obvious. I understand that a claim would have been obvious if

common sense directs one of ordinary skill, without hindsight bias caused by knowledge of the challenged claims, to combine multiple prior art references or add missing features to reproduce the alleged invention recited in the claims.

22. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill or expertise in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus—a connection—between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

23. I am not aware of any allegations by the named inventors of the '209 patent or any assignee of the '209 patent that any secondary considerations tend to rebut the obviousness of any Challenged Claim of the '209 patent.

24. I understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the descriptions found in the patent being considered.

25. I understand that Petitioner has the burden of proving unpatentability by a preponderance of evidence, which means that the claims are more likely than not unpatentable.

V. GROUNDS

26. I understand that Petitioner is requesting *inter partes* review of claims 1, 2, 4-9, and 12-14 (the “Challenged Claims”), on the following statutory grounds:

	Reference(s)	Basis	Claims
Ground 1	Khan (EX1004) ¹	§ 103 – obviousness	1, 2, 4-9, 12-14
Ground 2	Khan and Riccardi (EX1005) ²	§ 103 – obviousness	1, 2, 4-9, 12-14
Ground 3	Khan, Riccardi, and Iizuka (EX1006) ³	§ 103 – obviousness	1, 2, 4-9, 12-14
Ground 4	Khan, Riccardi, Iizuka, and Hartono (EX1007) ⁴	§ 103 – obviousness	1, 2, 4-9, 12-14

¹ International Patent App. No. WO 2001/080096A1.

² U.S. Patent No. 8,392,188.

³ U.S. Patent No. 6,424,980.

⁴ International Patent App. Pub. No. WO 2000/011571A1.

VI. LEVEL OF SKILL IN THE ART AND STATE OF THE ART

A. Person Of Ordinary Skill In The Art

27. The person of ordinary skill in the art at the earliest alleged priority date in June 2003 (“POSITA”) would have had a bachelor’s degree in electrical engineering, computer science, computer engineering, or equivalent field, and two years of experience working with speech recognition and natural language processing systems. Additional work experience could make up for less education and vice versa. This definition would not differ meaningfully were the date in question June 2002. In this Declaration, I use the terms “one of ordinary skill” and “skilled artisan” as synonymous with the “person of ordinary skill in the art.” The state of the art discussed below reflects some of the knowledge and skill of one of ordinary skill.

B. State Of The Art

28. Certain concepts and processes reflect the state of the art for speech recognition and natural language processing systems during the relevant 2002-2003 time period, as described in textbooks such as Jurafsky (EX1010, published 2000) and Huang (EX1011, published 2001).

29. Systems that processed continuous spoken natural language and produced helpful, application-specific responses were widely available prior to 2002. In 1999, for example, as I led a company developing natural-language

question-answering systems, I would ride my bike to work wearing a headset to test interfaces that would give me stock quotes and other information through my phone, even with the wind blowing against the mike. As the '209 patent explains, these natural language systems were getting better. EX1001, 1:38-43.

30. Most of the major improvements alluded to in the '209 patent's introduction were driven by probabilistic methods and included increasingly large language models. These methods used probabilities at every stage in processing, searching a large space of interpretations in order to choose the most likely meaning (i.e., a semantic interpretation of the words in an acoustic input). Invariably, these models, or at least some part of the model, was domain-specific (i.e., based on context), because one of the main technical issues was to narrow down the number of possible phonetic or word choices for a system at every stage (known as "perplexity"). *E.g.*, EX1010, 204; EX1011, 122, 560; EX1023, [0020]. Therefore, nearly every spoken language system, and there were many, relied upon a probabilistic domain model. *E.g.*, EX1010, 14 (explaining how "probabilistic and data-driven models had become quite standard throughout natural language processing" in the 1990s); EX1011, 73-131 (describing probability theory for natural language processing), 554-560 (describing probabilistic language models).

31. The '209 specification does not pretend to advance the state of the art in spoken language processing. Rather, its claimed advance is to provide a "complete

environment.” EX1001, 1:15-17, 1:43-46. Therefore, even in the most generous reading of the patent and claims in their entirety, the claimed innovation is a combination of already-known methods. The specification does not really inform on what part of this “complete environment” might lead to better or more helpful responses. The “complete environment” includes components such as parsing an input, determining its meaning, determining a context given the meaning, invoking a “domain agent” for the context, and using the domain agent to generate a response.

32. Virtually every natural language system I have worked with includes all of these elements together. For example, while at GE Corporate R&D in the 1980s my team developed a transportable natural language interface called TRUMP, where the whole idea was to provide a knowledge framework for language analysis that would work in multiple domains. *E.g.*, EX1030, 270; EX1033, 245-246, 248-251. We connected it to the company database of researchers and answered simple questions about their contact info and organization. EX1033, 247-248. We also connected it to a speech recognizer, which was, of course, probabilistic.

33. Like many other teams, we also developed technology to combine statistical and natural language methods (e.g., syntactic and semantic parsing) to categorize language. EX1031, 178-184. We did *not*, to my recollection, combine in one “environment” the well-known techniques of speech recognition, domain agents (i.e., the back-end, for example, to query the database and generate the responses),

and categorization (i.e., the probabilistic determination of context), although we experimented with and made advances in each individual area.

34. However, my team was not the only one working in these areas, and, as I have noted, by 2002/2003 all of these technologies had emerged from the laboratory into widespread commercial use. There were also numerous other projects that I am aware of that produced helpful responses to spoken queries in various domains. And some of those, as will be described below, included a component to identify the domain (of course, by using the meaning of the interpreted language) and then pass the interpreted query to a domain-specific agent for processing and response. *E.g.*, EX1004, EX1007.

35. To recap, technology enabling human-machine dialogue in spoken natural language existed and was commercially available in 2002/2003. EX1010, 1. Skilled artisans at the time would have known that a spoken language system for human-machine interaction “needs to have both **speech recognition** and **speech synthesis** capabilities.” EX1011, 2 (emphasis added). For such system to be useful, “[a]n **understanding and dialog component** is required to manage interactions with the user; and **domain knowledge** must be provided to guide the system’s interpretation of speech and allow it to determine the appropriate action.” EX1011, 2 (emphasis added). Those of ordinary skill were guided by the goals of “robustness, flexibility, ease of integration, and engineering efficiency.” EX1011, 2.

36. As I mentioned earlier, skilled artisans knew that natural language systems relied heavily on probabilistic models. Computers recognize the words in speech, converting spoken language into text representations by using prior probabilities, learned from the statistical analysis of large amounts of speech and text data, to determine which words are most likely given the sounds of the spoken language. EX1011, 4-5. Speech recognition systems are based on Bayes' rule for estimating the likelihood of the observed acoustic signal given a word sequence (using acoustic models) and combining it with the prior probability of that word sequence (using language models like n-grams). *Id.*, 4-5, 133-135 (discussing Bayesian decision theory for pattern recognition), 453 (describing confidence measures for speech recognition), 554-560 (describing stochastic language models and n-gram language models); EX1010, 142 ("the essential probabilistic architecture that we will use to solve both spelling and pronunciation problems: the Bayes Rule and the noisy channel model"), 191-204 (describing n-grams), 220-224 (describing n-grams for correcting spelling and modelling pronunciation), 238-241 ("the prior probability, is called the language model").

37. Skilled artisans understood that adapting language models based on user observation helps "to get a language as natural as possible." EX1022, 364. Similarly, "modelling the dynamics of dialogue, as well as user characteristics,"

enables “[f]luent interaction” between a human user and a computer system. EX1022, 364.

38. One of ordinary skill also knew that “when confronted with syntactically ambiguous input that allows for more than one possible syntactic representation[,] linguistic preferences captured by probabilistic models have been found to resolve a significant portion of syntactic ambiguity.” EX1023, [0018]. Further, “[a]s is understood by those skilled in the art, the techniques utilized for speech recognition and speech synthesis are inherently related.” EX1009, 1:42-44.

39. The '209 patent includes broad and generic disclosure of “the use of probabilistic and fuzzy reasoning” throughout its specification. EX1001, 2:5-8; *id.*, 4:14-17 (“probabilistic or fuzzy set decision and matching methods can be applied”), 4:24-29 (“use probabilistic and fuzzy set decision and matching methods”), 5:30-35 (“using probabilistic or fuzzy set decision and matching methods”), 5:41-43 (“using for example, probabilistic or fuzzy set matching methods”), 6:58-61 (“using a probabilistic or fuzzy set decision method”), 11:66-12:3 (“[t]he library may include utilities for ... fuzzy and probabilistic evaluation”), 13:56-57 (“probabilistic or fuzzy weights”), 17:13-21 (“the fuzzy set possibilities or prior probabilities”), 17:45-47 (“applying prior probabilities or fuzzy possibilities”), 17:51-56 (“these probabilities or possibilities”), 18:18-26 (“applying probabilistic or fuzzy reasoning”), 19:11-17 (“apply probabilistic or fuzzy set analysis”), 19:47-51 (“using probabilistic or fuzzy

scoring techniques”), 20:66-21:3 (“using probabilities or fuzzy possibilities”), 23:30-33 (“probabilistic or fuzzy set decision and matching methods may be applied”), 25:3-11 (“applying probabilistic or fuzzy reasoning”). This broad and general disclosure reflects the fact that one of ordinary skill understood how to use probabilistic models for natural language methods.

VII. THE '209 PATENT

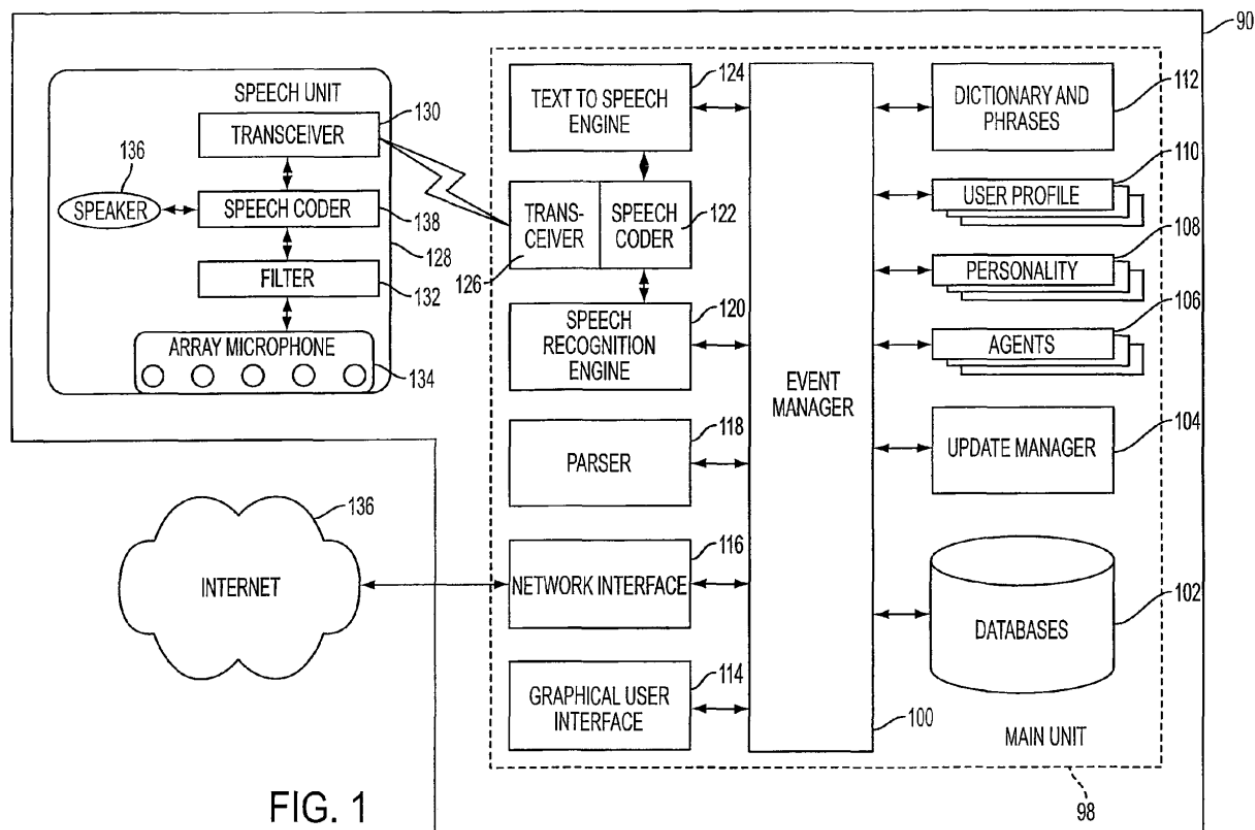
40. The '209 patent issued July 8, 2008, from an application filed June 3, 2003, and alleges priority to a provisional application filed June 3, 2002. EX1001, Cover.

a. The '209 Patent's Specification

41. The '209 patent describes “a complete speech-based information query, retrieval, presentation and command environment.” EX1001, 1:61-64. Within this environment, speech recognition is first applied to a user utterance (*e.g.*, EX1001, 16:53-56), as was done in “a wide range of [then-existing] applications” (EX1001, 1:41). The recognized speech is then parsed, with keyword matching used to determine a context / domain of the speech. *E.g.*, EX1001, 17:41-18:65, 21:56-59. “For example, a question with the keywords [sic] ‘temperature’ implies a context value of weather for the question.” EX1001, 17:49-51. A domain agent associated with the context / domain then processes a request in the speech. *E.g.*, EX1001, 2:50-57, 17:47-49, 18:8-10, 18:62-65. Specifically, the domain agent(s) retrieves the data

needed to respond to the request and processes the data. *E.g.*, EX1001, 19:18-20, 19:46-51. Using a text to speech engine, the results are then “relayed to the user and/or third parties as an audio message.” EX1001, 20:45-47, 22:31-33, 11:2-10.

Figure 1 depicts the various software modules that perform this method:



EX1001, FIG. 1; *id.*, 9:66-11:10.

A. The Prosecution History

42. I have reviewed the file history of the '209 patent. At this time, I do not believe the file history informs the meaning of the claims or prior art.

43. During prosecution, the Examiner rejected the claims as unpatentable over Halverson (U.S. Patent No. 6,523,061) alone or in combination with either

Reisman (U.S. Patent No. 7,062,488) or Handel (U.S. Patent No. 6,195,651). EX1002, 189-195. In response, Applicant amended the independent claims to recite, *inter alia*, “formulating the at least one request ... in accordance with a grammar used by the selected domain agent,” and argued that Halverson does not teach this concept. EX1002, 230-233, 243-245. The Examiner disagreed and rejected the amended claims as obvious over Halverson in view of Kuhn (U.S. Patent No. 6,901,366) alone or in further view of either Junqua (U.S. Patent No. 6,415,257) or Handel. EX1002, 307-314.

44. The Examiner found, however, that newly added independent claim 44, which recited “phrase tables” not recited by the other claims, was allowable because “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 and selecting a domain agent.” EX1002, 315. Applicant accepted the allowed claims (i.e., cancelled the rejected claims). EX1002, 334. In the subsequent notice of allowability, the Examiner emphasized the claims’ phrase tables: “[w]hile it is known to recognize words based on a vocabulary defined by a dictionary for speech recognition, the prior art of record does not disclose or reasonably suggest additionally utilizing phrase tables for speech recognition.” EX1002, 342.

45. The Examiner made these statements despite the well-established use of phrases in speech recognition, as even described in prior art references submitted by the Applicant. *E.g.*, EX1002, 94, 96 (Weber WO reference submitted in IDS). One prior art reference in the file history describes matching phrases to entries in a context-specific grammar, which enables faster speech recognition while “allow[ing] the user to speak in a more conversational style, using sentence fragments.” EX1002, 108-109. One of ordinary skill understood phrase tables as augmenting the lexicon (dictionary) with known phrases, such as proper nouns (e.g., “New York”), because this is how speech recognition systems built their lexicons. Even the first natural language processing system I worked on used a lexicon that included words and phrases, and was called PHRED (PHRasal English Diction). *See* Appendix A. Further, the specification of the ’209 patent does not explain what a phrase table is or how it differs from a dictionary, so the ’209 specification itself suggests that one of ordinary skill would have understood that a dictionary would have been augmented with a phrase table. *See* EX1001, 16:52-56 (“recognizes words and phrases, using information in the dictionary and phrase tables”). The Examiner made an error by allowing the claims on this basis.

VIII. CLAIM CONSTRUCTION

46. Unless otherwise discussed below, I have been instructed to apply the plain and ordinary meaning of all claim terms. I understand that Petition reserves the

right to argue in any district court case or other proceeding that terms in the '209 patent are indefinite or otherwise, and to raise additional issues of claim construction.

A. Claim Construction Order For Related '006 Patent

47. I understand that certain claim terms appearing in the '209 patent are similar to or the same as terms appearing in claims of related patent U.S. 8,015,006, and have been construed by a court as indicated below. As explained below, the claims are obvious at least under these constructions.

Term	Construction for '006 patent in E.D. Va.
context	“the subject matter area to which a particular user input is directed and which is used to determine the meaning of the user input” (EX1028, 1)
context stack	“a data structure that stores or references contexts in a manner that can be ordered” (EX1028, 1)
grammar	“a set of rules governing how inputs and requests to a domain agent should be structured” (EX1028, 2)
domain agent	“executables associated with a specific domain that receive, process and respond to user questions, queries and commands” (EX1028, 2)
procedures sensitive to the determined context	“procedures that differ based on which context has been determined” (EX1028, 2)

IX. GROUND 1: CLAIMS 1, 2, 4-9, AND 12-14 ARE OBVIOUS OVER KHAN

48. As explained below, claims 1, 2, 4-9, and 12-14 are obvious over Khan.

A. Khan (EX1004)

49. Khan is an international patent application filed April 18, 2001, published October 25, 2001, and claiming priority to a U.S. application filed April 18, 2000. EX1004, Cover. It is my understanding that Khan thus qualifies as prior art under at least Sections 102(a) and (e). Khan is analogous art to the '209 patent partly because they are both in the same field of endeavor of fulfilling user requests submitted through a speech interface. EX1001, 1:13-22; EX1004, 1:9-11.

50. Khan describes “a service engine capable of fulfilling users['] requests submitted in natural language form from a variety of input devices with a high degree of accuracy using a variety of data sources particularly adapted to fulfilling users['] needs in particular domains.” EX1004, 1:23-26. To fulfill a user’s spoken request, the system performs several stages of processing, which Khan illustrates using an airfare example:

- Speech Recognition: Khan “translat[es] spoken words ... in natural language voice format into text format.” EX1004, 3:21-24, 4:29-5:7.
 - Airfare Example: The user says “I want to fly from New York to Seattle next weekend. Find me the cheapest fare available.” EX1004, 9:7-11. The recognition engine produces the outputs:

- “i want to fly from new york to seattle next weekend find me the cheapest fare available” with 95% probability of correctness;
 - “i want fly from newark to seattle next week and find me the cheapest fare available” with 5% probability of correctness. EX1004, 9:11-21.
- Natural Language Understanding: Khan parses the recognized text to “determine its meaning” and “select the domain of the request.” EX1004, 6:29-31, 16:9-11; *id.*, 11:30-16:26 (describing parsing algorithms).
 - Airfare Example: “Travel is identified as the domain as the probability of the domain is identified as one hundred percent,” and relevant “information contained in the request is identified as ‘fly’, ‘new york’, [‘]seattle’, ‘next weekend’, ‘cheapest’, and ‘fare’.” EX1004, 16:20-26; 15:11-15 (showing extracted keyword phrases “from new york” and “to seattle”).
- Agent Processing: Khan formats the request and sends the formatted request to an “intelligent service agent” specific to the understood domain of the request. The agent automatically performs tasks to fulfill user requests, “such as searching a database for records corresponding

to a query or electronically reserving or purchasing tickets, such as airline tickets, or electronically purchasing a specified item.” EX1004, 2:27-31; *id.*, 7:6-14.

- Airfare Example: The intelligent service agent for the travel domain maps the request to different variables of a travel-domain template, asks the user questions to clarify or obtain essential information, and performs the search as well as “any other actions necessary to fulfill the user’s request.” EX1004, 16:27-18:33.

- For example, “if the above request did not specify the date or dates of travel to be ‘next weekend’, the intelligent agent might prompt the user with the question ‘What date or dates would you like to travel?’ and process the user’s response.” EX1004, 17:34-18:7.
- Aside from searching, “if the user had requested that the service engine reserve the least expensive seat on the applicable flight, instead of merely identifying it, the intelligent agent might generate an order automatically,” which is then “sent automatically to the applicable airline

or other vendor and a confirmation might be included with the results of the search.” EX1004, 18:25-30.

- Voice Response: Khan selects the best result(s) that fulfill the user request, and assembles a response “to be played to the user.” EX1004, 19:5-18.

51. For the different stages of processing the request, Khan provides a set of “probabilities” that are dynamically updated over time to learn and improve from user requests. EX1004, 8:32-9:6 (probabilities for speech recognition), 12:9-12 (describing “probability factor” for “a priori-determined grammar and dictionary databases”), 14:32-34 (statistical methodologies “uses each request to update its learning of specific requests”), 15:27-30 (“suggests one or several domains with assigned probability factors”), 16:8-14 (parsing “based on probabilistic methods”).

52. Khan discloses the same integrated approach to a natural language speech interface as the ’209 patent. As explained in detail below, one of ordinary skill implementing Khan would have at least found each Challenged Claim to be obvious.

1. Claim 1

a) Element [1.1]

[1.1] A method responsive to a user generated natural language speech utterance, comprising:

53. To the extent that the preamble is limiting, Khan teaches it. Khan discloses a “method for fulfilling a user’s request using a service engine, wherein the user is prompted to submit a request through an interface in natural language format.” EX1004, 2:6-11; *id.*, 8:33-9:5 (describing speech recognition for “each computed *utterance*”) (emphasis added). Khan discloses various interfaces including a “[t]elephone interface” with “speech recognition software” allowing users to interact via speech. EX1004, 4:29-5:7; *see also id.*, 4:11-28 (describing interfaces enabling interaction via “any communication device convenient to users” such as telephones and smart telephones), 5:29-32 (describing an “[a]utomobile communication system interface” using “standard voice and data communications protocols”). Khan thus provides a method responsive to a user generated natural language speech utterance.

a) Element [1.2]

[1.2] receiving the user generated natural language speech utterance, the received user utterance containing at least one request;
--

54. Khan teaches this element. Khan discloses an example natural language utterance spoken by a user (i.e., *user generated*). EX1004, 9:7-11; *see also id.*, 4:29-5:7 (describing a “[t]elephone interface” and “speech recognition software”). Khan discloses a natural language utterance as a “request [that] is received by the service engine” (i.e., *containing at least one request*). EX1004, 8:30; *id.*, 8:30-9:6 (the

processing of a “request” provided via voice interface “constructs words of the *requests* and associates probabilities to each computed *utterance*”) (emphasis added). Khan defines a “Request” as “[a]n instruction to a service engine, such as an instruction to search for information relating to a query, an instruction to place an order, an instruction to make a reservation, or an instruction to perform several tasks, especially several related tasks.” EX1004, 3:8-10. Khan thus receives a user-generated natural-language speech utterance containing at least one request.

b) Element [1.3]

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

55. Khan teaches this element. Khan discloses maintaining a dynamic set of prior probabilities usable at each stage of processing the received user utterance, because Khan teaches “associat[ing] probabilities to each computed utterance” during initial speech recognition that are then sent “for further processing,” including semantic parsing where a “probability factor” is provided for “the closest possible outcomes.” EX1004, 8:30-9:6, 12:9-12; *id.*, 9:7-21 (providing example outputs and probabilities for a given speech utterance), 13:10-16 (describing “estimates on the context of the phrase”). Further, a “statistical parsing algorithm applies statistical methodologies to arrive at the meaning and context of requests,” and “uses a knowledge base of past learned data and uses each request to update its learning of

specific requests” (*dynamic*). EX1004, 14:32-34; EX1001, 25:6-9 (“Prior probabilities or fuzzy possibilities and associated values are received from a number of sources including the history of the dialog, the user profile 110, and the agent.”). Khan also discloses a “domain identity parsing algorithm [that] performs the task of identifying the service domain that is required to activate to answer the user queries,” where the algorithm “suggests one or several domains with assigned probability factors.” EX1004, 15:27-30.

56. One of ordinary skill would have understood that the “probabilities” and “probability factors” taught by Khan are a set of prior probabilities. Specifically, as Khan discloses accessing “several associated databases” when processing the utterance and “associates probabilities to each computed utterance” (EX1004, 8:32-9:6, 9:11-20), one of ordinary skill would have understood that a set of probabilities is stored and thus *maintained* in at least one of these databases. The skilled artisan further would have understood that the probabilities are *prior probabilities* because they are probability estimates based on previous, historical information, namely “past learned data.” EX1004, 14:33-34; *see also* EX1004, 6:7-12 (describing “databases relating to past requests of users”), 9:10-20 (describing example where probabilities are associated with computed utterances based on “requests previously entered by the user or by different users”); EX1015, 8:54-65 (describing “ascrib[ing] different *a priori probabilities* to various possible interpretations of input that

correspond to sub-domain tasks”) (emphasis added). These prior probabilities are *dynamic* because the service engine “uses each request to update its learning of specific requests.” EX1004, 14:32-34; EX1001, 25:6-9.

57. Further, skilled artisans knew that a dynamic set of prior probabilities would “enhance speech recognition by *dynamically* changing the likelihoods over words, and thus, phonemes uttered.” EX1025, 32:7-10 (emphasis added). As Horvitz explained in 1998, “[t]hose skilled in the art of speech recognition will appreciate the power that could be gained if the speech language models used to characterize phonemes by probability were dynamically influenced by information computed about the goals, intentions, or needs of computer users.” EX1025, 31:3-7; *id.*, 31:12-33:8. Riccardi teaches dynamically adapting a set of prior probabilities for use in speech recognition “in the user’s next utterance” by “transform[ing] a *prior probability* distribution from one domain to another without speech transcriptions (on-line adaptation).” EX1005, 6:9-18 (emphasis added). This allows an “adaptive spoken dialog system” to “react to changes in the task ... and adapt to unseen speech and language events” (e.g., “out-of-domain speech data”) with “on-line learning of language. EX1005, 6:18-20, 2:22-49, 4:39-44. These teachings of dynamic probabilities are consistent with how the ’209 patent describes maintaining a dynamic set of prior probabilities, specifically: “the fuzzy set possibilities or prior probabilities for the words in the dictionary and phrase tables may be dynamically

updated to maximize the probability of correct recognition at each stage of the dialog.” EX1001, 17:13-16.

58. The skilled artisan therefore would have been motivated to implement Khan with a *dynamic* set of prior probabilities because Khan teaches “us[ing] each request to update its learning of specific requests,” and the skilled artisan would have been further motivated, with a reasonable expectation of success, to maintain a *dynamic set of prior probabilities* to ensure that the processing at each stage (e.g., during speech recognition, during parsing, etc.) adapts and improves over time, while allowing flexibility for new domains to be added. EX1004, 14:33-34; *see also id.*, 26:14-15, 26:28-29, 27:8-9, 27:19-20 (claims for “updating the database with the determined meaning of the request”); EX1005, 2:18-65 (describing advantages of “fast adaptation of language model probabilities to a new domain” and “on-line learning of language” over traditional speech recognition techniques); EX1024, Abstract (“The system gets more and more word models and keeps improving its performance the more it is used.”). The skilled artisan would have reasonably expected success because it was well known how to “constantly update[]” statistical / probabilistic models used for natural language processing. EX1024, Abstract; *id.*, 13:44-14:9 (updating “probabilistic language model[s]” which are “well-known to those skilled in the art of natural language speech recognition”).

c) **Element [1.4]**

[1.4] recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;

59. Khan teaches this element. Khan teaches recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables. Specifically, Khan teaches a “recognition engine” that “has several associated databases, which allow it to detect phonemes in the request.” EX1004, 8:30-33; *id.*, 3:21-24 (defining a “Voice Recognition Interface”). To recognize words and phrases in the utterance, the recognition engine of Khan “access[es] specialized databases containing specific grammar, dialect, and other speech recognition related information.” EX1004, 9:10-14. Specialized databases taught by Khan include domain-specific “grammars and dictionaries” as well as “domain specific keyword and phrase databases” (*information in one or more dictionary and phrase tables*). EX1004, 11:33-34, 13:14. Using this information, the recognition engine of Khan “constructs words of the requests and associates probabilities to each computed utterance.” EX1004, 8:33-9:5; *id.*, 9:15-20 (describing example outputs of recognized words and phrases for a given utterance).

60. Khan thus teaches recognizing words and phrases contained in the received utterance using information in one or more dictionary or phrase tables.

d) **Element [1.5]**

[1.5] 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;

61. Khan teaches this element. Khan discloses a “natural language processing module 152, which *parses* the formatted information *to determine its meaning*, preferably utilizing a plurality of parsing algorithms” (*parsing the recognized words and phrases to determine a meaning of the utterance*). EX1004, 6:29-31 (emphasis added); *id.*, 11:30-16:7 (describing and illustrating “four parallel parsing algorithms” including “case/tree parsing, keyword/phrase parsing, statistical parsing, and domain identity parsing”). The “formatted information” comprises *recognized words and phrases* of the utterance. See EX1004, 4:29-5:7 (describing telephone interface with “speech recognition software”), 6:19-28 (describing how “[e]ach request received from a user is received by the appropriate interface module, which then extracts the information and ... converts the extracted information into a common format”).

62. To determine the meaning of the utterance, each of the parsing algorithms of Khan 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. For example, one of the parsing algorithms is a “keyword/phrase parsing

algorithm” that “identifies the keywords embedded in each request,” and “use[s] *phrases and keywords from the request to understand the context* of the request.” EX1004, 13:10-14 (emphasis added). Another parsing algorithm is a “domain identity parsing algorithm [that] performs the task of identifying the service domain that is required to activate to answer the user queries.” EX1004, 15:27-28. The domain identity parsing algorithm “processes the request by *identifying the keywords and types of phrases* in the request and *suggests one or several domains* with assigned probability factors.” EX1004, 15:28-30 (emphasis added). Similarly, a “statistical parsing algorithm applies statistical methodologies to arrive at *the meaning and context of requests*.” EX1004, 14:32-34 (emphasis added). Further, the “case/tree parsing algorithm uses semantic based parsing methods to process the query,” where “domain based, a priori-determined grammar and dictionary databases” are used “to *match the query with the closest possible outcomes* and provide the probability factor to it.” EX1004, 12:9-12 (emphasis added). One of ordinary skill would have understood that this “match” using grammar and dictionary databases is a determination of the context based on one or more keywords contained in the recognized words and phrases.

63. Finally, the output of the parsing algorithms is evaluated “to select the domain of the request” (*determining a context for the at least one request contained in the utterance*). EX1004, 16:8-11; *id.*, 16:11-16 (discussing probabilistic

approaches to combining the output). Khan uses both “context” and “domain” to refer to a particular subject matter area, such as “travel,” that informs the meaning of the utterance. *E.g.*, EX1004, 13:10-16 (the service engine “accesses domain specific keyword and phrase databases” and “use[s] phrases and keywords from the request to understand the context of the request”), 16:21-22, 16:27-32 (describing using a “template relating to the domain of the request (and possibly to certain keywords in the request),” indicating more specific contexts within a domain), 20:16-18 (claimed service engine “identifies the context of the request” and “the body of information searched in response to the request is chosen by the processor based on the context of the request”). Thus, to the extent the construction of “context” as “the subject matter area to which a particular user input is directed and which is used to determine the meaning of the user input” applies, Khan discloses this element.

64. Khan provides an illustrative example where the utterance is “I want to fly from New York to Seattle next weekend. Find me the cheapest fare available.” EX1004, 9:10-11. The speech recognition engine produces outputs of recognized words and phrases, with associated probabilities that they are correct:

i want to fly from new york to seattle next weekend find
me the cheapest fare available (95%)
i want fly from newark to seattle next week and find me
the cheapest fare available (5%)

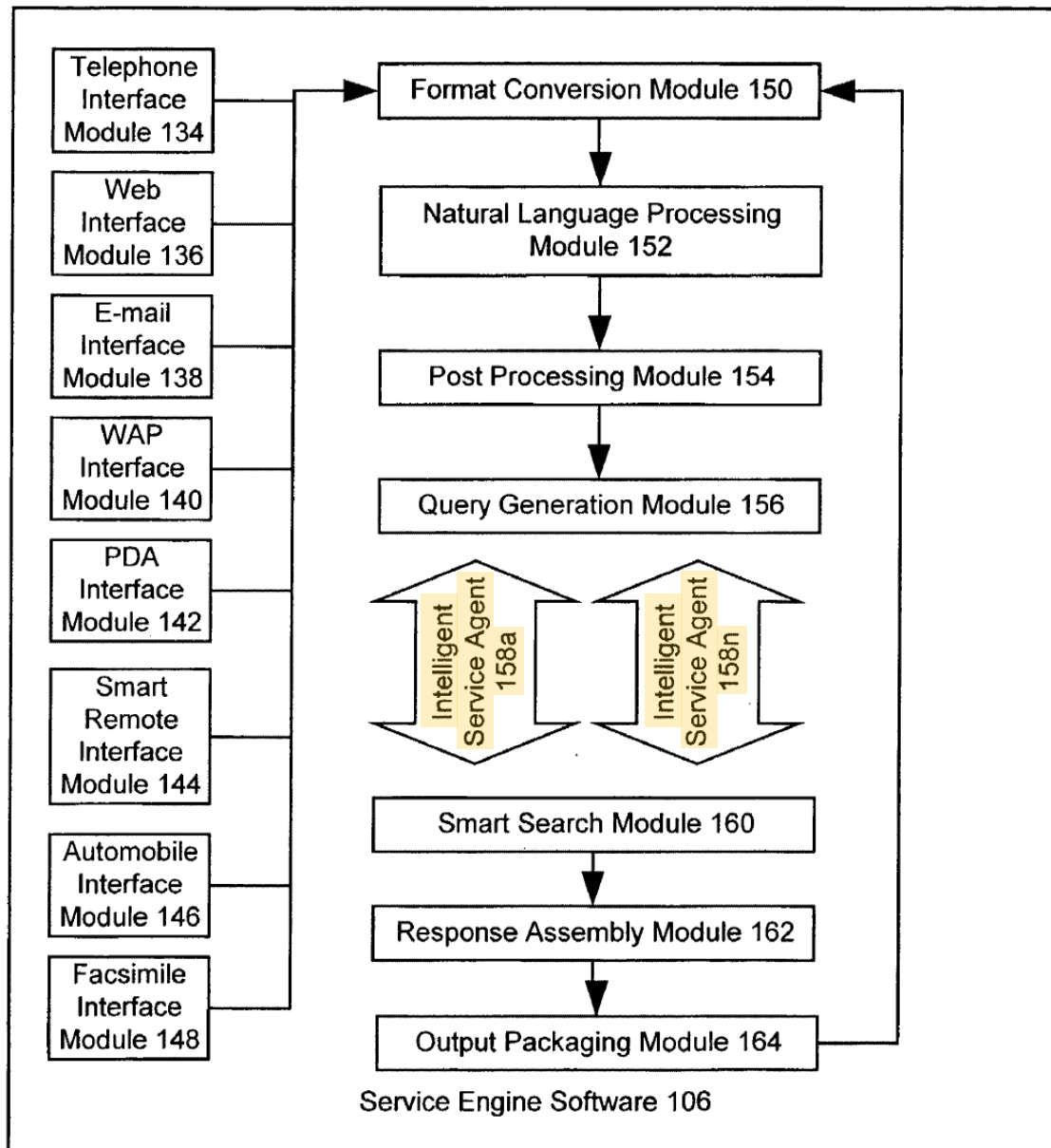
EX1004, 9:17-20. To illustrate how the domain identity parsing algorithm determines the context “by identifying the keywords and types of phrases in the request,” Khan discloses that, for this example, “the domain identity parsing algorithm detects the words [‘]fly[’], [‘]from’, ‘to’ and ‘fare’,” and that “[t]hese keywords match the travel domain and hence the output of the module is to identify the travel domain with a 100% certainty.” EX1004, 15:27-34; *id.*, 16:20-26. Khan thus discloses this element.

e) **Element [1.6]**

[1.6] 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;

65. Khan teaches this element. Khan discloses that an “intelligent service agent is selected based on the domain to which the complex request relates” (*selecting at least one domain agent based on the determined meaning*). EX1004, 28:15-16; *id.*, 7:6-9 (“forwarding to an appropriate intelligent service agent, which is preferably a domain specific agent”). An “Intelligent Service Agent” (*domain agent*) is a “program that performs a task automatically” (*an autonomous executable*) and “understands specific domains queried by a user and assists in identifying and collecting all information relevant to fulfilling the user’s request” (*receives, processes, and responds to requests associated with the determined*

context). EX1004, 2:27-31. The service engine of Khan includes a plurality of intelligent service agents, as depicted in FIG. 1B:



EX1004, FIG. 1B (highlighting added). Each intelligent service agent “is preferably a domain specific agent” (*domain agent*). EX1004, 7:6-9. To process and respond to requests associated with a determined context, “[i]ntelligent service agents 158a

through 158n preferably contain knowledge of variables (especially domain specific variables) and knowledge bases related to those domains and are able to generate follow-up questions for the user when additional information is necessary.” EX1004, 7:9-12. The intelligent service agents are *autonomous* because they automatically perform tasks independent of further user control, such as “electronically reserving or purchasing tickets, or electronically purchasing a specified item.” EX1004, 2:27-31; *id.*, 18:24-33. For all these reasons, Khan also satisfies the construction of “domain agent” as “executables associated with a specific domain that receive, process and respond to user questions, queries and commands.” *Supra* Section VIII.

f) Element [1.7]

[1.7] 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;

66. Khan teaches this element. Khan discloses “generat[ing] an appropriate query in an internally used format for forwarding to an appropriate intelligent service agent, which is preferably a domain specific agent.” EX1004, 7:6-9. Khan also discloses “grammars” that are “domain specific” (*a grammar used by the selected domain agent to process requests associated with the determined context*). EX1004, 6:9-12; *id.*, 9:11-14, 11:33-34. After determining the context/domain of the request, “the request is mapped by an intelligent service agent to a template relating to the domain of the request” (*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). EX1004, 16:27-28; see also id., FIG. 2A (step 210: “Map combined output to a template”).

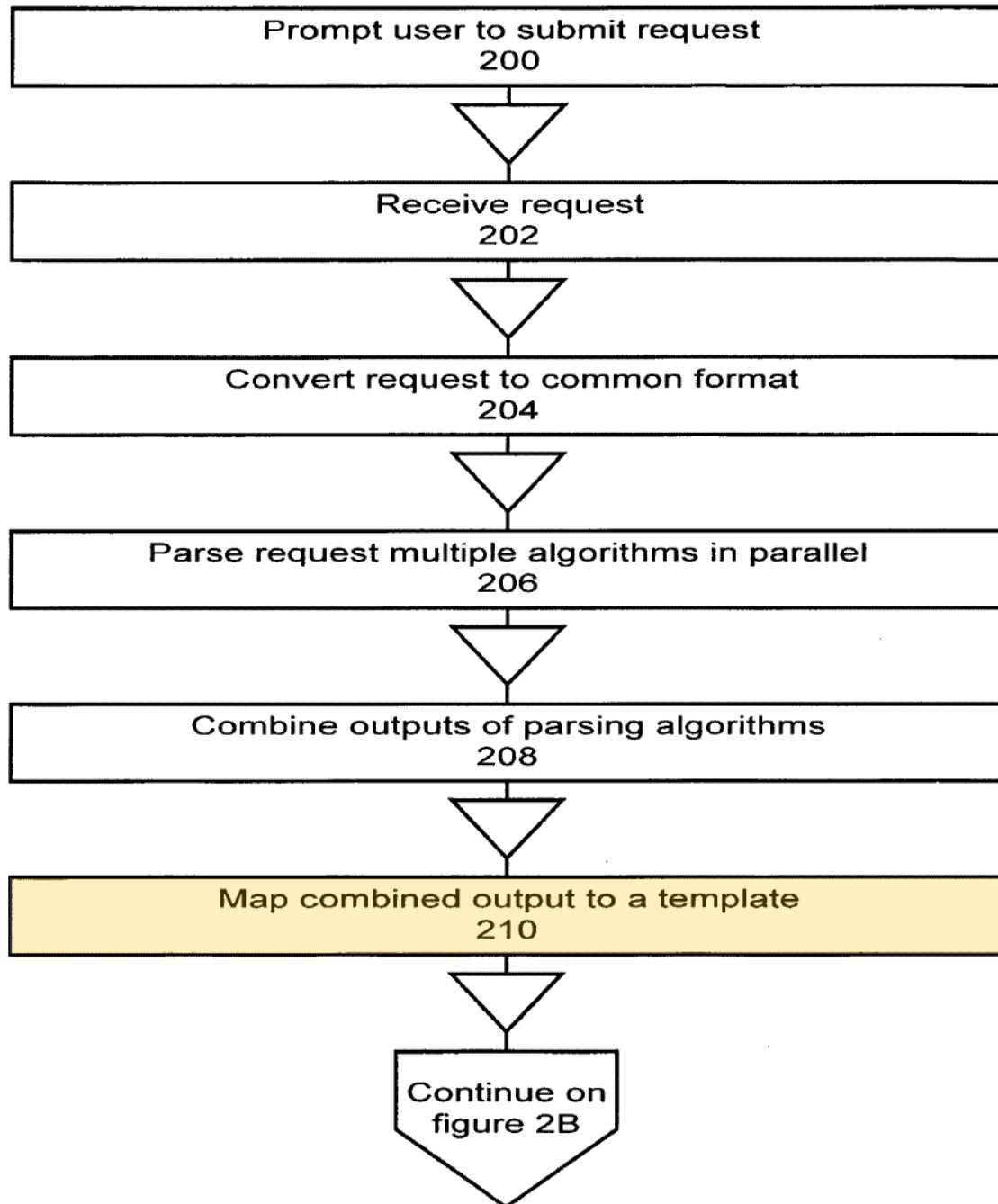


Figure 2A

EX1004, FIG. 2A (highlighting added).

67. Khan maps the request to the domain-specific template in accordance with the domain-specific grammar used by the intelligent service agent (*domain agent*) and thus formulates the request in a way that allows the intelligent service agent to process the request. EX1004, 6:9-12 (“grammars that may be ... domain specific”), 9:11-14 (specific grammar ... specific to the travel domain”), 11:33-34 (“grammars ... specific to the domain of the request”), 11:30-16:8 (parsing the utterance into an “XML request”), 16:27-17:30 (mapping the XML request into the domain-specific template).

68. Khan also satisfies the “grammar” construed as “a set of rules governing how inputs and requests to a domain agent should be structured.” *Supra* Section VIII. Khan teaches using “an XML template relating to the domain of the request (and possibly to certain keywords in the request),” where the XML template for a given domain or context encodes the expected structure of valid inputs and reflects known parameters and constraints of the domain. EX1004, 16:31-17:8. An example XML template for the travel domain includes semantic tags such as “<MODE>” for mode of transportation, “<ORIG>” for origin, “<DEST>” for destination, and so on. EX1004, 17:11-30. This XML schema, which includes semantic tags for “required variables, secondary variables and search variables,” governs how inputs and requests to the intelligent service agent (*domain agent*) are

structured. EX1004, 16:31-17:10 (describing XML template and variables of the request). By mapping the request to the template, Khan *formulates the request in accordance with a grammar used by the selected domain agent*.

g) Element [1.8]

[1.8] invoking the selected domain agent to process the formulated request; and

69. Khan teaches this element. Khan discloses the intelligent service agent (*selected domain agent*) “executing the complex request” (*process the formulated request*). EX1004, 27:32; *id.*, 28:30-31, 29:11, 29:23. Khan teaches “forwarding [an appropriate query in an internally used format] to an appropriate intelligent service agent, which is preferably a domain specific agent” (*invoking the selected domain agent*). EX1004, 7:6-9; *id.*, FIG. 1B (intelligent service agents 158a through 158n). In response, the “appropriate intelligent service agent” (*the selected domain agent*) taught by Khan “performs a task automatically, such as searching a database for records corresponding to a query or electronically reserving or purchasing tickets, such as airline tickets, or electronically purchasing a specified item” (*process the formulated request*). EX1004, 2:27-31; *id.*, 18:24-28 (describing how, “if the user had requested that the service engine reserve the least expensive seat on the applicable flight, instead of merely identifying it, the intelligent agent might generate an order automatically using previously stored user information, such as a credit card number of the user and the user’s seating preferences”). Further, as part of processing

the formulated request, “the intelligent agent prompts the user to supply any essential information that is not understood or any essential information that has not been supplied by the user.” EX1004, 17:31-33; *id.*, 16:32-17:10, 17:33-18:13, 2:6-11, 2:29-31, 7:9-12.

h) Element [1.9]

[1.9] 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.

70. Khan teaches this element. In Khan, “the results generated by the service engine are assembled into a response” (*the presented results generated as a result of the invoked domain agent processing the formulated request*), and “the response is sent to the user using the designated interface” (*presenting results of the processed request to the user*). EX1004, 19:5-6, 19:18; *id.*, 7:15-19 (describing modules for preparing results into “context specific response” for “transmission to the user”), 19:6-17 (describing the “content and format of this response” that “will be returned to the user”). Specifically, the intelligent service agent (*the invoked domain agent*) performs “any search necessary to fulfill the request” and “any other actions necessary to fulfill the user’s request” (*processing the formulated request*). EX1004, 18:14-33.

71. Khan provides an illustrative example where results of the processed request are presented to the user. Where the request involves finding the cheapest

airfare available for a flight from New York to Seattle next weekend (EX1004, 9:7-11), Khan discloses that “instead of merely identifying [the least expensive seat on the applicable flight], the intelligent agent might generate an order automatically using previously stored user information, such as a credit card number of the user and the user’s seating preferences” (*the invoked domain agent processing the formulated request*). EX1004, 18:25-28. Khan teaches that “a confirmation [of this order] might be included with the results of the search” (*results of the processed request*). EX1004, 18:28-30. These are the results that Khan teaches “assembl[ing] into a response” and “sen[ding] to the user using the designated interface,” such as via the voice interface. EX1004, 19:5-7, 14-21. Figure 1B of Khan depicts the response assembly module, output packaging module, and format conversion module taught by Khan for presenting results of the processed request to the user:

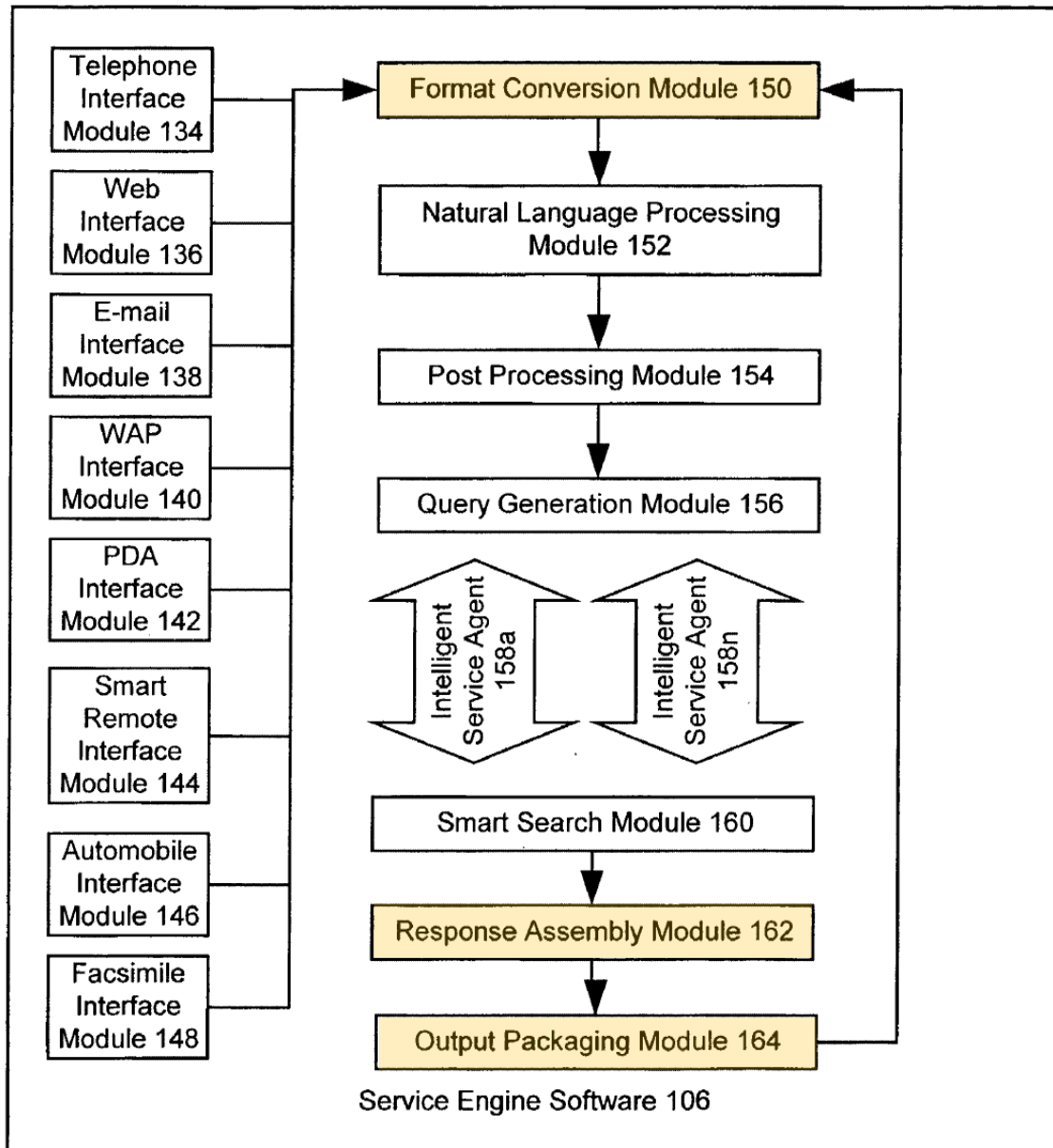


Figure 1B

EX1004, FIG. 1B (highlighting added). Figure 2B of Khan depicts the steps of assembling the response, packaging the response, and sending the response:

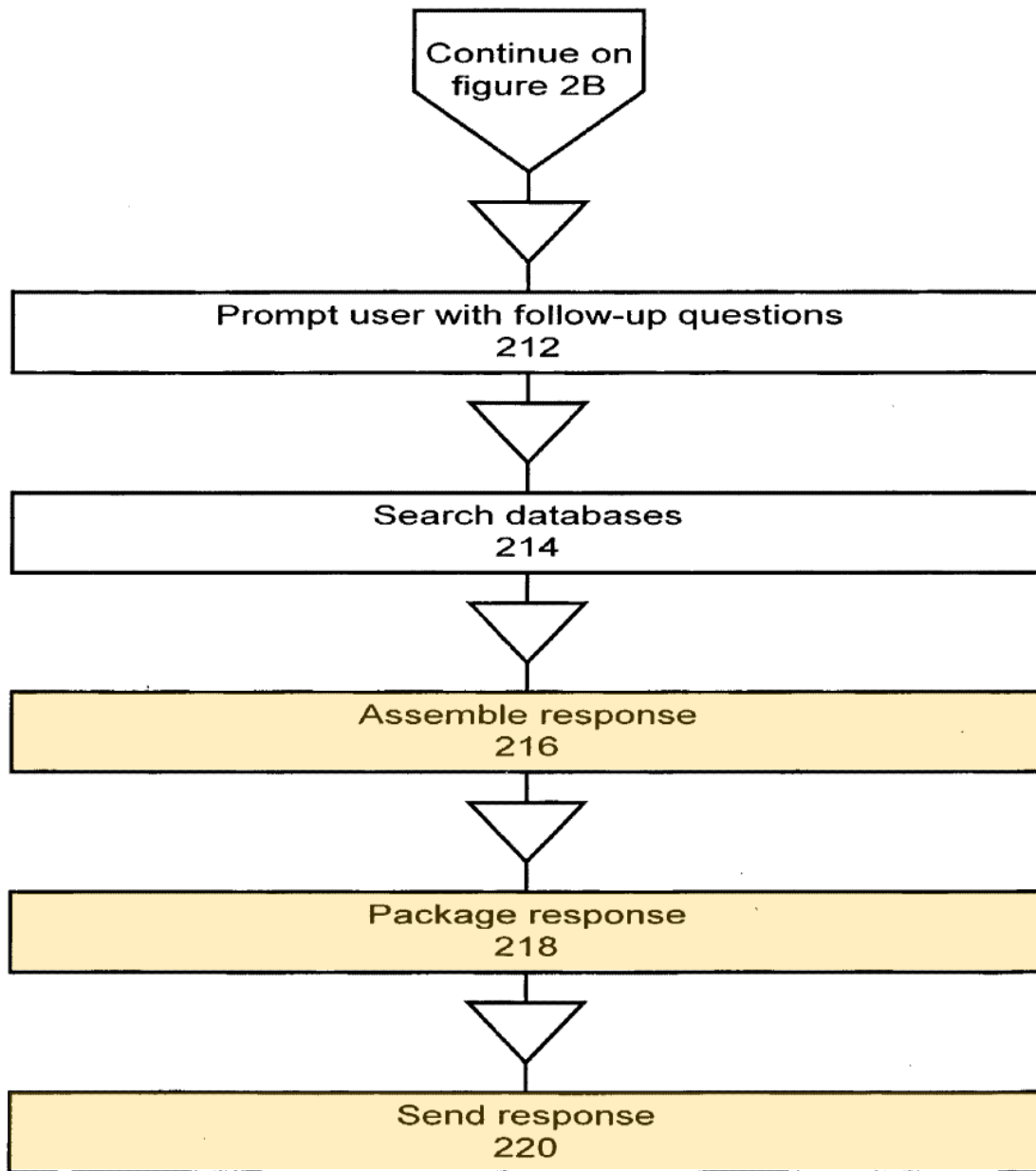


Figure 2B

EX1004, FIG. 2B (highlighting added).

2. Claim 2

a) Element [2.1]

[2.1] The method according to claim 1, wherein recognizing the words and phrases contained in the received utterance includes:

72. Khan renders claim 1 obvious and teaches this element. *See* Sections IX.A.1 (claim 1), IX.A.1.c) (element [1.4]).

b) Element [2.2]

[2.2] dynamically updating the information in the dictionary and phrase tables based on the dynamic set of prior probabilities or fuzzy possibilities;

73. One of ordinary skill would have found this element obvious in view of the teachings of Khan. As I discussed above in Section IX.A.1.b) (element [1.3]), Khan teaches and therefore at least renders obvious maintaining a dynamic set of prior probabilities usable at each stage of processing the utterance. Khan discloses the speech recognition system “access[ing] specialized databases containing specific grammar, dialect, and other speech recognition related information,” such as dictionaries and “phrase databases” (*dictionary and phrase tables*), for use in recognizing words and phrases in the received utterance. EX1004, 9:11-14, 13:14; *id.*, 8:30-9:5 (describing the “recognition engine” using “several associated databases, which allow it to detect phonemes in the request”), 9:15-20 (providing example recognition outputs), 11:32-34 (“databases may include grammars and dictionaries”). Khan further teaches *dynamically updating* databases so that the

system learns from requests over time. EX1004, 14:32-35 (system “uses a knowledge base of past learned data and uses each request to update its learning of specific requests”).

74. A skilled artisan thus would have been motivated to dynamically update the information in the dictionaries and phrase databases used for speech recognition based on the dynamic set of prior probabilities, because Khan already teaches dynamically updating “its learning of specific requests,” and the skilled artisan would have been motivated to similarly improve speech recognition over time by updating the probabilities for words and phrases in the dictionaries and phrase databases (*the information in the dictionary and phrase tables*). EX1004, 14:32-34; EX1005, 6:10-15 (describing dynamically updating model for use “in recognizing phone sequences in the user’s next utterance”), 6:51-54 (“probabilities of the selected words and phrases”); EX1011, 921-923 (describing how a “vocabulary can also be dynamically provided by the application” in a natural language-agent system); EX1024, 27:48-28:14 (describing a “phrase recognition algorithm designed to recognize an entire phrase of continuous or connected speech” and “updat[ing] the language model” for the phrase recognizer).

c) **Element [2.3]**

[2.3] determining an identity of the user based on voice characteristics of the utterance; and

75. Khan teaches this element. Khan teaches a voice interface that uses a “biometric voiceprint ... to help validate the user’s identity” (*determining an identity of the user based on voice characteristics of the utterance*). EX1004, 7:32-33. Voiceprints are like fingerprints and have long been used to verify the identity of a person speaking because “there are certain characteristics of an individual’s voice which cannot be disguised and which are as effective in identifying an individual as the ridges of his fingerprint.” EX1013, 2:4-7 (U.S. patent filed 1966 for a “Voice Verification System”). Khan teaches the user “speaking the username and password” to “access the system” and that the “biometric voiceprint may instead or in addition be used to help validate the user’s identity.” EX1004, 7:27-33. One of ordinary skill would have recognized that determining an identity of the user based on voice characteristics of the utterance is the same or at least encompasses validating an identity of the user based on a biometric voiceprint, because each identifies a user based on their voice and, as described by the ’209 patent, functions “as a security measure for applications, such as auctions or online shopping, where this is required.” EX1001, 16:56-67. Further, skilled artisans knew how to use various parameters characterizing a speaker’s voice to identify the speaker from an

utterance. For example, Newman describes how models of different speakers' "pronunciation of different phonemes," along with a scoring system based on parameters "that represent the frequency content" of the user speech, are used to identify a particular speaker. EX1014, 3:65-67, 4:55-57; *id.*, 3:40-8:33 (describing methods for a speaker identification system). One of ordinary skill thus would have been motivated, with a reasonable expectation of success, to determine an identity of the user based on voice characteristics because Khan teaches validating a user's identity based on a biometric voiceprint, and further because doing so was a well-known technique in the art.

d) Element [2.4]

[2.4] associating the recognized words and phrases and a pronunciation of the recognized words and phrases with the identity of the user, the determined context, and the at least one request when the recognized words and phrases satisfy a predetermined confidence level.

76. One of ordinary skill in the art would have found this element obvious in view of the teachings of Khan. Khan teaches associating recognized words and phrases with the identity of the user, the determined context, and the at least one request when the recognized words and phrases satisfy a predetermined confidence level. Specifically, Khan teaches "assigning a probability of correctness to the determined meaning" (*predetermined confidence level*) of recognized words and phrases, and "when the probability of correctness is at least equal to the

predetermined value, updating the database with the determined meaning of the request” (*the determined context, and the at least one request*), where the database was used to parse the request. EX1004, 26:5-15; *see also id.*, 14:32-34 (describing a “knowledge base of past learned data and us[ing] each request to update its learning of specific requests”). Khan also teaches discarding “[a]ny information that has an associated probability below a predetermined value” (*predetermined confidence value*) and specifically describes rejecting a candidate string of recognized words and phrases “as having an unduly low probability of occurrence.” EX1004, 16:11-14, 16:20-25. Regarding the database that is updated, Khan describes a “user profile” (*identity of the user*) stored in a “user preference database” including information about user preferences and historical choices, and “databases relating to past requests of users” that are used to determine the meaning of a request. EX1004, 18:8-13, 6:9-12, 21:9-10, 22:29-32.

77. Although Khan does not expressly disclose associating pronunciation with a user, Khan teaches using existing speech recognition systems, such as the Nuance system, to perform speech recognition. EX1004, 4:33-5:7. Skilled artisans would have known that speech recognition systems at the time, such as the Nuance system, included pronunciations for each word in dictionary files and that it was quite typical to try to improve the accuracy of speech recognition by training a recognizer on a specific user’s pronunciation. *E.g.*, EX1012, 85. The “recognition

package” for configuring the Nuance system includes recognition grammars and “[p]ronunciations for the words in those grammars.” EX1012, 10. The Nuance system allows users to “add spoken phrases to a dynamic grammar,” where “the recognition system listens to the user speak a phrase, generates pronunciation for that phrase by performing phonetic recognition of the utterance, and *returns the pronunciation*, along with a phrase identifier, to the application.” EX1012, 29 (emphasis added). Khan further teaches “the intelligent agent prompt[ing] the user to supply any essential information *that is not understood*,” which would have included information that the service engine could not confidently recognize due to the user pronunciation. EX1004, 17:31-33. One of ordinary skill would have been motivated to associate the pronunciation of the recognized words and phrases with the identity of the user, the determined context, and the at least one request, in order to improve the system’s response to subsequent utterances from that user. EX1014, 5:18-49 (describing adapting a model “to a particular known speaker”), 8:29-33 (updating speech model for the identified speaker with frames and recognized text). One of ordinary skill would have reasonably expected success in doing so, because the speech recognizers cited by Khan already took advantage of user pronunciation and the skilled artisan would have easily associated the pronunciation with the user identity and the user request in the existing databases.

3. **Claim 4**

a) **Element [4.1]**

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

78. Khan renders claim 1 obvious and teaches this element. *See* Sections IX.A.1 (claim 1), IX.A.1.d) (element [1.5]).

b) **Element [4.2]**

[4.2] 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;

79. One of ordinary skill would have found this element obvious in view of the teachings of Khan. Khan discloses that a “keyword/phrase parsing algorithm identifies the keywords embedded in each request,” and thus “us[ing] phrases and keywords from the request to understand the context of the request,” by “access[ing] domain specific keyword and phrase databases” (*matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords*). EX1004, 13:10-14; *id.*, 15:32-33 (“keywords match the travel domain”). Khan further discloses that the keywords are associated with “estimates on the context of the phrase.” EX1004, 13:14-16. Khan also discloses a “domain identity parsing algorithm” that “processes the request by identifying the keywords and types of phrases in the requests and suggests one or several domains with assigned

probability factors” (*matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords*). EX1004, 15:27-30. Furthermore, Khan teaches “us[ing] each request to update its learning of specific requests,” so that estimates of a context are improved over time (*the dynamically updated set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities*). EX1004, 14:32-34; *see also* EX1004, 26:5-27:20 (claiming updating the parsing databases based on confidence).

80. As I discussed above in Section IX.A.1.b) (element [1.3]), an obvious implementation of Khan maintains the dynamic set of prior probabilities or fuzzy possibilities in order to improve speech recognition and domain classification when processing subsequent requests. A skilled artisan would have found it obvious to dynamically update the keywords in the “domain specific keyword and phrase databases” based on the dynamic set of prior probabilities, because Khan already teaches dynamically updating “its learning of specific requests,” and a skilled artisan would have been motivated to similarly update the keywords with the predictable result of achieving better determinations of context and domain over time. Thus, the dynamic set of prior probabilities of Khan is used to dynamically update the set of keywords used for determining the context of the request contained in the utterance. In this way, the parsing algorithms of Khan for determining the context of the request learn words and phrases used idiosyncratically by a user over time and thus correctly

identify the meaning and context of the request intended by the user. Suppose hypothetically that a user says “I want to *jet* from New York to Seattle next weekend,” rather than “I want to *fly* from New York to Seattle next weekend,” in Khan’s example. *E.g.*, EX1004, 9:10-11. If the initial language model does not include “jet” as a keyword for the travel domain, the obvious implementation of Khan would have learned an increased probability for the word “jet” being associated with the travel domain through its dialogue with the user.

c) **Element [4.3]**

[4.3] 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

81. Khan teaches this element. Khan discloses “identifying the keywords and types of phrases in the request and suggest[ing] one or several domains with assigned probability factors” (*scoring one or more possible contexts for the matched keywords*). EX1004, 15:27-30. The scores are the “estimates on the context of the phrase” or the “probability factor” determined for each possible context. EX1004, 12:9-12, 13:10-16, 14:32-34, 15:27-30.

82. When scoring the one or more possible contexts, Khan discloses using “[d]atabases of user preferences and previously submitted requests of the user and of other users” (*a profile of the user*) to parse the recognized words and phrases of the utterance and determine the context of the request. EX1004, 11:30-12:5. Khan

further discloses using “domain based” / “domain specific” databases (*data content associated with the domain agents*) for the parsing and scoring of the contexts. EX1004, 12:10-11, 13:14-16. Khan further discloses an example where “template matching may be used” for evaluating possible domains (*contexts*) of the request, “whereby a user’s query is matched to the stored template best matching the query.” EX1004, 16:16-18. In this approach, “a template relat[es] to the domain of the request” and is associated with a respective “intelligent service agent,” and one of ordinary skill would therefore have understood the templates as *data content associated with the domain agents*. EX1004, 16:27-28. Further, the element recites “one or more of” and Khan teaches at least using a profile of the user, and thus renders claim 4 obvious regardless of how “context stack” is construed. *Supra* Section VIII.

d) Element [4.4]

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

83. Khan teaches this element. Khan teaches using “combining algorithms ... to select the domain of the request” (determining the context by selecting one or more of the possible contexts). EX1004, 16:8-11. The domain is selected “based on highest probability results,” where “[a]ny information that has an associated probability below a predetermined value may be discarded at this time” (selecting

one or more of the possible contexts having a score satisfying a predetermined confidence level). EX1004, 16:11-14. Khan further discloses “prompting the user with a question relating to the request” when the “probability of correctness” (score) assigned to “the determined meaning” (one or more of the possible contexts) is “below a predetermined value” (predetermined confidence level). EX1004, 26:5-27:20. Further, Khan teaches “when the probability of correctness is at least equal to the predetermined value, updating the database with the determined meaning of the request.” EX1004, 26:14-15; see also *id.*, 24:30-33 (describing “select[ing] the parsed product with the highest assigned degree of confidence”). One of ordinary skill would have understood that Khan thus discloses determining the context by selecting one or more of the possible contexts having a score satisfying a predetermined confidence level.

84. Khan discloses that other approaches may be used which also include scoring and selecting one or more possible contexts. *E.g.*, EX1004, 16:14-16. As an illustrative example, Khan discloses identifying the keywords “fly,” “from,” “to,” and “fare” in the example request, determining that “[t]hese keywords match the travel domain and hence the output of the module is to identify the travel domain with a 100% certainty.” EX1004, 15:31-34. Although the example in Khan only includes one domain with “100% certainty,” one of ordinary skill would have understood that requests may include words and phrases that match keywords and

phrases of different domains, with different respective probabilities. For example, the keyword phrase “next week” in Khan’s examples might match an e-mail/message domain, but with a much lower probability of correctness than in the travel domain because the other words and phrases in the utterance do not match other keywords in the e-mail/message domain, and so this possible context would be ignored. Conversely, if the utterance was “I want to send an e-mail to my friend about the meeting next week,” the keyword phrase “next week” might still match the travel domain, but that domain would have a lower certainty because other words in the utterance (e.g., “send,” “e-mail,”) do not match keywords for the travel domain; instead, the e-mail/message domain would have a higher certainty.

4. Claim 5

a) Element [5.1]

[5.1] The method according to claim 4, wherein determining the context for the at least one request contained in the utterance further includes:

85. Khan renders claim 4 obvious and teaches this element. *See supra* Sections IX.A.1.d) (element [1.5]), IX.A.3 (claim 4).

b) Element [5.2]

[5.2] requesting the user to verify the recognized words and phrases when none of the possible contexts have a score satisfying the predetermined confidence level; and

86. Khan teaches this element. Khan discloses that “the user is repeatedly prompted with a question relating to the request if the evaluated request contains insufficient understood data to allow the service engine to fulfill the request.” EX1004, 2:6-11. Specifically, after “pars[ing] the formatted information to determine its meaning,” Khan discloses that “[i]f the results are not understood” (*when none of the possible contexts has a score satisfying the predetermined confidence level*), “a follow-up question may be generated ... for dispatch to the user” (*requesting the user to verify the recognized words and phrases*). EX1004, 6:29-7:6. Khan further discloses that “initiat[ing] a dialog with the user to obtain further information” when required variables to process the request “are not available or have low associated confidence levels.” EX1004, 17:8-10; *id.*, 17:31-18:7. One of ordinary skill would have understood that such dialog seeking clarification would have included *requesting the user to verify the recognized words and phrases*, because the computed utterances could lead to significantly different outcomes, such as purchasing a flight “from new york to seattle next weekend” and “from newark to seattle next week.” EX1004, 9:15-20. If the system is not sufficiently confident in one of these recognitions, one of ordinary skill would have understood that the system would seek verification rather than proceed with processing.

c) **Element [5.3]**

[5.3] determining the context based on information provided by the user in one or more subsequent natural language speech utterances.

87. Khan teaches this element. Khan discloses, after “prompting the user with a question relating to the request,” parsing the subsequent utterance and “determining the meaning of the request” (*determining the context based on information provided by the user in one or more subsequent natural language speech utterances*) repeatedly “until the probability of correctness is at least equal to the predetermined value.” EX1004, 26:5-15; *see also id.*, 17:8-10 (“the intelligent agent will initiate a dialog with the user to obtain further information”).

5. **Claim 6**

a) **Element [6.1]**

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

88. Khan renders claim 1 obvious and teaches this element. *See supra* Sections IX.A.1 (claim 1), IX.A.1.f) (element [1.7]).

b) Element [6.2]

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

89. Khan teaches this element. As I discussed in Section IX.A.1.f), Khan teaches mapping the request with a domain-specific grammar to a domain-specific XML template (*the grammar used by the selected domain agent*). Khan teaches that the mapping of the input to the “XML template relating to the domain of the request” includes “required variables, secondary variables and search variables” (*one or more required values and one or more optional values for a request*). EX1004, 16:27-17:10. Khan teaches that “[t]he required variables are those necessary to conduct any relevant search for the information request to be processed,” while “[t]he secondary variables are not necessary but extremely helpful in pinpointing the information requested and the search variables identify the requested information.” EX1004, 16:34-17:7. Khan discloses *determining* these values by “retriev[ing]” the “template relating to the domain of the request.” EX1004, 16:31-32.

c) Element [6.3]

[6.3] 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. Khan teaches this element. Khan discloses that “the request is mapped by an intelligent service agent to a template relating to the domain of the request”

(extracting one or more criteria and one or more parameters from the keywords contained in the recognized words and phrases). EX1004, 16:27-28. When mapping the request to the template, Khan discloses extracting criteria and parameters from the request *using procedures sensitive to the determined context*, for example by determining a date range based on the keyword phrase “next weekend” in the request and determining that a “Fare” criterion is “Low” based on the keyword phrase “find cheapest fare” in the request. See EX1004, 15:14-15 (the request containing “next weekend” and “find cheapest fare” as keyword phrases), 17:19 (providing “Feb 00 11-13” as a date range for “DATE” variable), 17:29 (providing “<CRITERIA>Low</Criteria>” for “Fare”). One of ordinary skill would have understood that Khan discloses that mapping these keyword phrases to the template involves *procedures sensitive to the determined context* because doing so is dependent on the specific aspects of the template for the determined domain, such as specific criteria and parameters (e.g., a “Fare” criteria). EX1001, 18:15-21 (“context sensitive procedures for extracting the criteria or parameters from the user’s question or command”). This disclosure in Khan also satisfies the construction of “procedures sensitive to the determined context” as “procedures that differ based on which context has been determined,” because templates for different contexts will include different criteria and parameters, and thus the mapping of a request to a

particular template will differ based on the parameters and criteria of the template for the determined context. EX1004, 16:27-17:18.

d) Element [6.4]

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

91. Khan teaches this element. As noted above, Khan discloses an example where “Low” is identified as the criteria for “Fare” in the example request where “find cheapest fare” was a keyword phrase from the request and dates identified as variables for the keyword phrase “next weekend” (*inferring one or more criteria and one or more parameters*). EX1004, 15:14-15, 17:19, 17:29. One of ordinary skill would have understood that these criteria and parameters are inferred *using the dynamic set of prior probabilities* because Khan discloses using “probability factors” to “associate[] confidence levels” with the criteria and parameters, so these confidence levels would have been assigned to the criteria and parameters using the dynamic set of prior probabilities (i.e., learned probabilities that associate a domain with a keyword). EX1004, 17:7-10.

e) Element [6.5]

[6.5] 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.

92. Khan teaches this element. Khan discloses populating the “template relating to the domain of the request” with information from the request (*transforming the extracted and the inferred criteria and parameters into a format compatible with the grammar used by the selected domain agent*). EX1004, 16:27-28; *supra* Section IX.A.1.f). Khan provides an example request prior to the mapping (*transforming*):

```
<LISTQUERY>
<QUERYINT> i want to fly from new york to seattle next weekend find me
the cheapest fare available
<PROBFACT> 0.95 </PROBFACT>
<KEYWRD-PHP,ASE> fly </KEYWRD-PHRASE>
<KEYWRD-PHRASE> from new york </KEYWRD-PHRASE>
<KEYWRD-PHRASE> to seattle </KEYWRD-PHRASE>
<KEYWRD-PHRASE> next weekend </KEYWRD-PHRASE>
<KEYWRD-PHRASE> find cheapest fare </KEYWRD-PHRASE>
</QUERYINT>
</LISTQUERY>
<LISTDOMAIN>
TRAVEL <PROBFACT> 1.0 </PROBFACT>
</LISTDOMAIN>
```

EX1004, 15:7-16, 16:5-7.

93. After mapping (*transforming*) the extracted and inferred values to the template, “the request may consist of the following XML representation”

corresponding to the template relating to the domain (*the format compatible with the grammar used by the selected domain agent*):

```
<TITLE> XML TRAVEL REPRESENTATION </TITLE>
<TRAVELTEMP>
<MODE> AIR </MODE>
<LISTREQVAR>
<ORIG> new york </ORIG>
<DEST> seattle </DEST>
<DATE> Feb 00 11 -13 </DATE>
</LISTREQVAR>
<LISTSECVAR>
<CARR> Open </CARR>
<TIME> Open </TIME>
<CLSS> Open </CLSS>
<MEAL> Open </MEAL>
<ARPT> Open </ARPT>
</LISTSECVAR>
</LISTSRCVAR>
<V1> Fare <CRITERIA> Low </Criteria> </V1>
</TRAVELTEMP>
```

EX1004, 17:11-30. As shown, the representation (*transformed criteria and parameters*) includes the required variables such as origin, destination, and date (*all of the required variables*) and designates the secondary variables such as <TIME> and <CARR> (*one or more of the optional variables*) as “Open.” EX1004, 17:16-27 (indicating required variables as “LISTREQVAR,” origin as “ORIG,” destination as

“DEST,” date as “DATE,” secondary variables as “LISTSECVAR,” and various secondary variables left “Open”); *id.*, 17:5-7 (“secondary variables are not necessary”). Khan discloses populating one or more of the secondary variables (*one or more of the optional variables*) “(if not earlier supplied by the user) by accessing a user preference database.” EX1004, 18:8-9; *id.*, 18:9-13 (describing the intelligent agent extracting and inferring values for the secondary variables based on user preferences and historic records for the user).

6. Claim 7

a) Element [7.1]

[7.1] 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,

94. Khan renders claim 1 obvious, and one of ordinary skill would have found this element obvious in view of the teachings of Khan. *Supra* Section IX.A.1 (claim 1). Khan discloses that the request (*the at least one request contained in the utterance*) is “[a]n instruction to a service engine, such as an instruction to search for information relating to a query, an instruction to place an order, an instruction to make a reservation, or an instruction to perform several tasks, especially several related tasks” (*includes at least one question and at least one command*). EX1004, 3:8-10. While Khan does not explicitly mention utterances in command or question form, one of ordinary skill would have understood that any natural language engine

that processes a “query” would be expected to interpret queries in question or command form (e.g., “find a flight to New York on Sunday” versus “do you have a flight to New York on Sunday?”). *E.g.*, EX1010, 3 (describing similarity in statement and question), 198 (describing “sample user queries” in command and question form). Skilled artisans further knew that typical “utterances are composed of more than one sentence, so that in general, each input is a sequence of commands, questions, and/or assertions.” EX1032, 4:61-63. As Khan teaches handling multiple tasks, one of ordinary skill would have been motivated to implement Khan to handle a “complex request” including both a question and a command, in order “to fulfill the request” regardless of the query form. EX1004, 2:6-11, 27:21-28:8. One of ordinary skill would have reasonably expected success because skilled artisans knew how to “separate the utterance into its component instructions.” EX1032, 5:62-66; *id.*, 5:44-57, 6:4-65.

a) Element [7.2]

[7.2] wherein the invoked domain agent processes the formulated request by:

95. Khan teaches this element. *Supra* Section IX.A.1.g) (element [1.8]).

b) Element [7.3]

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

96. Khan teaches this element. Khan discloses a “[c]entral computer 100” (*local processing device*) including a processor and memory storing the service engine. EX1004, 3:33-4:10, FIG. 1A (reproduced below). The central computer is “connected to one or more of remote computers 166a through 166n” (*remote processing devices*) for processing requests. EX1004, 5:33-6:7. Khan discloses the service engine, via the intelligent service agent, performs “any other actions necessary to fulfill the user’s request,” such as “generat[ing] an order automatically” which is then “sent automatically to the applicable airline or other vendor” (*directing the at least one command to local or remote processing devices*). EX1004, 18:24-30; *id.*, 18:30-33 (describing how the service engine automatically performs other tasks according to a command in the request). EX1004, 18:30-33.

97. One of ordinary skill would have understood that the service engine performing such actions as *directing the at least one command to local or remote processing devices* because the command to perform one or more tasks is directed to and performed locally (at least in part by the intelligent service agent at the central computer, such as generating an order) and remotely (e.g., by “purchas[ing] goods

or services on behalf of a user” which would involve sending an order to remote computers for processing). EX1004, 18:24-33.

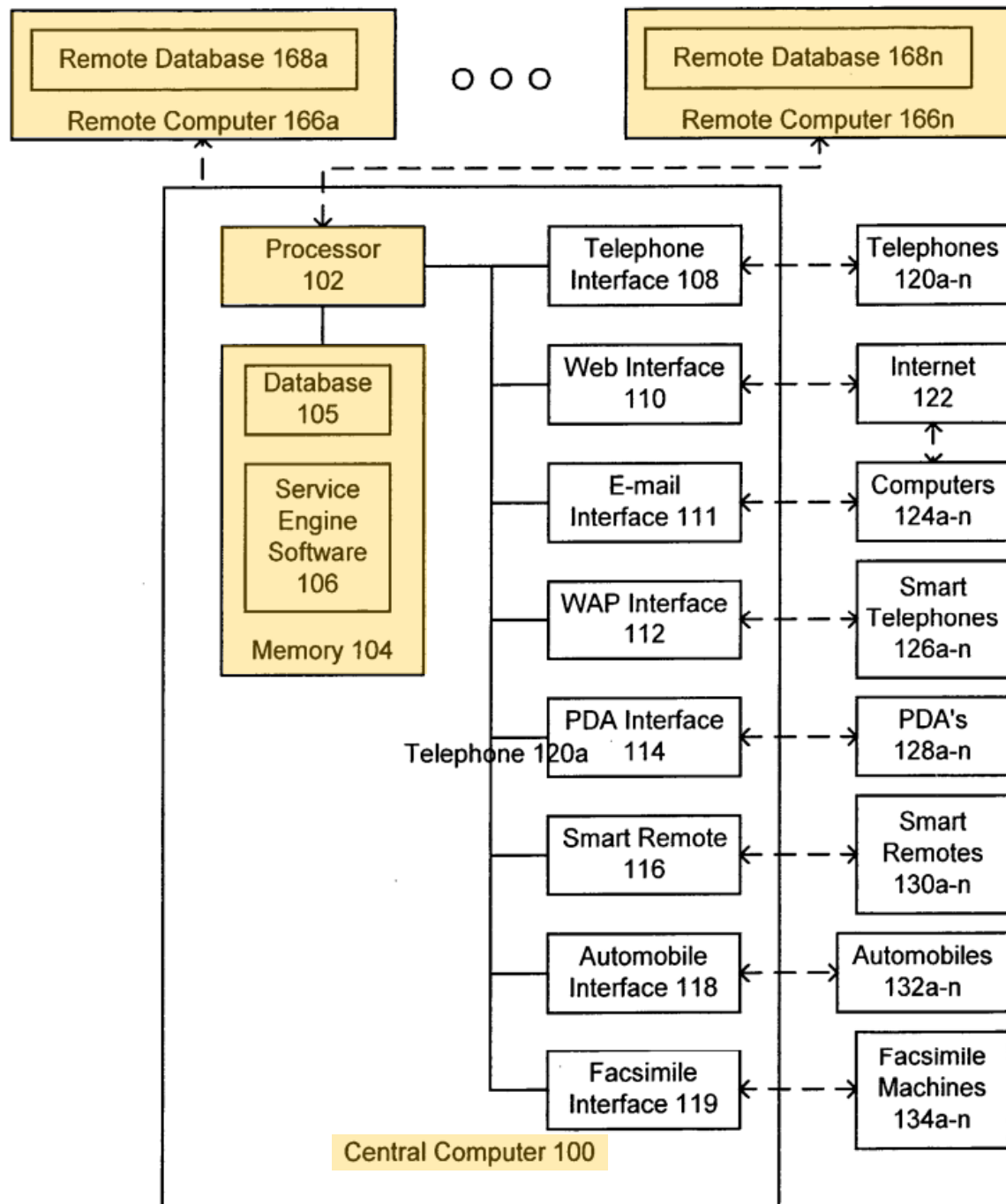


Figure 1A

EX1004, FIG. 1A (highlighting added).

c) **Element [7.4]**

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

98. Khan teaches this element. Khan discloses performing “any search necessary to fulfill the request” by “select[ing] and access[ing] one or more relevant knowledge bases or other databases (or part or all of the Web or Internet) and ... perform[ing] a search of each such source of data based on the XML request” (*querying local or network information sources to resolve information requested in the at least one question*). EX1004, 18:14-18. For the airfare example, Khan discloses selecting two databases, where “[d]epending on the architecture and format of each database, the algorithm generates appropriate searches, connects with remote databases, and retrieves information for the dates requested.” EX1004, 18:18-23.

7. **Claim 8**

a) **Element [8.1]**

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

99. Khan renders claim 7 obvious and teaches this element. *Supra* Section IX.A.6 (claim 7).

b) Element [8.2]

[8.2] 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;

100. One of ordinary skill would have found this element obvious in view of the teachings of Khan. For the airfare example, Khan discloses that the intelligent service agent uses a “smart search algorithm [to] select two databases to locate the least expensive domestic airline fares, one listing standard airline fares and one listing wholesale and discount travel operator fares” (*distinct information sources*). EX1004, 18:18-21. Then, “[d]epending on the architecture and format of each database, the algorithm generates appropriate searches, connects with remote databases, and retrieves information for the dates requested” (*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*). EX1004, 18:21-23. One of ordinary skill would have understood that the searches submitted to the different databases are *duplicate queries* because each search seeks the same information (e.g., the cheapest airfare between the designated origin and destination on the given dates). One of ordinary skill also would have found it obvious to implement the searches submitted to the different databases as *asynchronous queries* at least because the searches are independent of each other (e.g., the airfares returned from one database do not affect or depend on the airfares

returned from another database), and because the queries could be synchronous or asynchronous with predictable results. *E.g.*, EX1021, 1:57-2:8 (discussing synchronous versus asynchronous queries). Further, Khan teaches queries that “request information automatically (such as course catalogs and applications from colleges meeting certain criteria),” where there may be delayed feedback from the information source. EX1004, 18:30-33. One of ordinary skill thus would have understood that Khan obviously suggested *submitting asynchronous queries* because otherwise, the service engine would need to submit the searches sequentially (i.e., one at a time) and wait until results are received to continue processing, as the service engine would be limited to connecting to “at most one database at a time” if synchronous queries were used. EX1021, 1:57-67.

c) Element [8.3]

[8.3] asynchronously receiving a plurality of results of the queries from the information sources; and

101. One of ordinary skill would have found this element obvious in view of the teachings of Khan. As discussed *supra* Section IX.A.7.b) (element [8.2]), Khan discloses “generat[ing] appropriate searches, connect[ing] with remote databases, and retriev[ing] information for the dates requested” (*receiving a plurality of results of the queries*) when searching distinct databases (*from the information sources*) for the airfare example. EX1004, 18:21-23. In an obvious implementation of Khan as

discussed above, these “appropriate searches” are *a plurality of asynchronous queries* because they are submitted to each respective database independently and in parallel of each other, and further because in some instances a result may not be immediately returned by the information source. *E.g.*, EX1004, 18:30-33. One of ordinary skill would have understood that the “information for the dates requested” (*a plurality of results of the queries from the information sources*) would have been *asynchronously received* because the timing of receipt from each database may “[d]epend[] on the architecture and format of each database” rather than on the timing of receipt from another database. EX1004, 18:21-23.

d) Element [8.4]

[8.4] 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.

102. One of ordinary skill would have found this element obvious in view of the teachings of Khan. Khan discloses “assembl[ing] the output of the smart search module into a context specific response based on user preferences and the communications medium used by the user,” which involves scoring a relevance of each of the asynchronously received results using the dynamic set of prior probabilities to determine one or more best responses to the at least one question. EX1004, 7:15-17. The intelligent service agent “assists in identifying and collecting all information *relevant* to fulfilling the user’s request.” EX1004, 2:29-31 (emphasis

added); *id.*, 1:13-26 (service engine improves over existing approach that results in “both relevant hits ... and irrelevant hits”); 16:34-17:5 (“required variables are those necessary to conduct any *relevant* search for the information request to be processed”) (emphasis added). For the airfare example, Khan discloses determining “the least expensive fare (or only a few of the least expensive fares)” (*one or more best responses to the at least one question*) when assembling a response from the received results. EX1004, 19:5-12. Khan also discloses that the results may include “a substantially greater number of fares (perhaps relating to flights departing at different times or with different numbers of connections).” EX1004, 19:9-12. Khan discloses that “[t]he content and format of this response may be based on the interface through which the response will be returned to the user.” EX1004, 19:6-7. Consequently, the number of results returned may range from “only the least expensive fare” where a brief response is appropriate to “a substantially greater number of fares” where space is less limited. EX1004, 19:7-12.

103. One of ordinary skill implementing Khan would have found it obvious to rank the received results based on relevance to determine which results to provide to the user (*scoring a relevance of each of the asynchronously received results to determine one or more best responses to the at least one question*) so that only the most relevant result (e.g., “only the least expensive fare” satisfying all other parameters) can be returned, because Khan teaches prioritizing which results, or

“relevant hits,” to return to the user based on how close the result is to satisfying the parameters and criteria of the query (i.e., its relevance). EX1004, 1:13-26, 19:5-12; *id.*, 16:27-17:10. Skilled artisans knew how to score and rank the relevance of search results to identify “the most relevant search result,” as taught at least by Robbins, and would have easily and predictably implemented relevance scores in the service engine of Khan, especially because this is suggested by Khan. EX1026, [0006]; *id.*, [0046], [0053]-[0055]. Further, one of ordinary skill would have understood that Khan discloses scoring the relevance of these results *using the dynamic set of prior probabilities*, because the dynamic set of prior probabilities reflect “past learned data” relating to user requests and thus would enable the intelligent service agent to rank “flights departing at different times or with different numbers of connections” as more or less relevant depending on the historic preferences of the user and/or previous dialogue with the user. EX1004, 14:33-34, 19:9-12; *id.*, 6:9-12, 9:11-15, 11:32-12:5, 18:8-13 (disclosing that “the intelligent agent might identify airport preference” where “the user departed on ninety percent of previous recorded flights from LaGuardia Airport in New York”); *supra* Section IX.A.1.b) (element [1.3]). Scoring the relevance using the dynamic set of prior probabilities would also ensure that results using similar but non-exact terms matching the criteria/parameters would be appropriately weighted as relevant. *E.g.*, EX1026, [0053].

8. Claim 9

a) Element [9.1]

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

104. Khan renders claim 8 obvious and teaches this element. *Supra* Section IX.A.7 (claim 8).

b) Element [9.2]

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

105. One of ordinary skill would have found this element obvious in view of the teachings of Khan. Khan discloses that “the results generated by the service engine are assembled into a response” (*formulate a response to the question*), which includes determining one or more tokens. EX1004, 19:5-6. Specifically, for the airfare example, Khan discloses selecting “the least expensive fare (or only a few of the least expensive fares)” to “assemble[] into a response” (*determining one or more tokens required to formulate a response to the question*). EX1004, 19:5-12. One of ordinary skill would have understood that Khan thus discloses determining one or more tokens (e.g., a “fare” token) required to formulate a response to the question, because the response must include the identified information (e.g., the least expensive fare) to address the user’s request. Khan further discloses “identify[ing] the key information provided in the request to conduct information retrieval or

execute tasks” (EX1004, 16:9-11), where such key information is expressed in the query in terms of required variables, secondary variables, and search variables (EX1004, 16:31-17:8). One of ordinary skill would have found it obvious to use one or more such variables as one or more tokens required to formulate a response to the question, because doing so was a known technique that naturally and predictably allows natural language responses populated with appropriate values. *E.g.*, EX1012, 8 (describing that “[i]n the flight information example, you could use the tokens CITY and DATE”). For example, the skilled artisan would have understood that the “ORIG,” “DEST,” and “DATE” required variables for the airfare example would have corresponded to tokens used in the best response, so that the response would include: “The cheapest fare from *ORIG* to *DEST* on *DATE* is *FARE*,” where values for the tokens are substituted from the query results. *E.g.*, EX1004, 17:16-30 (depicting variables and criteria for the airfare example that may be used as tokens).

c) Element [9.3]

[9.3] 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;

106. One of ordinary skill would have found this element obvious in view of the teachings of Khan. As discussed above in Section IX.A.8.b), Khan discloses using “the key information provided in the request to conduct *information retrieval*” and that “the results generated by the service engine are assembled into a response”

(extracting one or more values for the determined tokens from the asynchronously received results). EX1004, 16:9-11 (emphasis added), 19:5-6. As Khan discloses, for the airfare example, that “only the least expensive fare (or only a few of the least expensive fares) ... might be returned in the response,” one of ordinary skill would have found it obvious that Khan would necessarily have extracted the values corresponding to these fares from the results received from the databases. EX1004, 19:7-12; 18:14-23 (describing searching databases to “retrieve[] information for the dates requested”). Khan further teaches that the retrieval of desired information from the databases “[d]epend[s] on the architecture and format of each database,” and that “[t]he content and format of this response may be based on the interface through which the response will be returned to the user.” EX1004, 18:14-23, 19:6-7. As the received results are in the format of the information source, one of ordinary skill would have understood that extracting the values from the received results requires using a format-specific method to extract only the desired information (*applying one or more scraping criteria to the asynchronously received results*), because the information to be extracted from the results may be structured significantly differently depending on the format of the information source (e.g., whether it is a certain type of database or a webpage accessed via “the Web or Internet”). EX1004, 18:14-23, 19:6-7.

107. Further, one of ordinary skill knew how to scrape (*extract*) desired information from an information source, as explained by Clendinning. EX1017, 6:7-23 (describing scraping “desired contents of the website” such as “the prices of products sold by the vendor” with a scraper “designed specifically to locate desired information on the third-party website and to interpret the format of this information”), 6:24-35 (describing how scraped output in XML format “allows the use of intelligent agents that seek out consistent information and then act on what they find”), 5:55-56 (“data may be retrieved by using known ‘scraper’ technology”). Similarly, Iizuka teaches “extracting required information item by item from different HTML documents” located via a search query, where the documents have different locations, structures, and attributes. EX1006, 4:60-65; *id.*, 4:44-51 (“retrieving required information from a plurality of semi-structured documents that ... have different document structures, presentation styles, and information elements”), 12:18-32 (describing query processing that “extract[s] information from the HTML documents with the use of the specified meta data”), 22:2-31 (describing an embodiment for information retrieval). One of ordinary skill implementing Khan would have been motivated, given Khan’s express teachings and known scraping techniques, to extract values for the tokens from the results with the predictable result of the response including the relevant information.

d) **Element [9.4]**

[9.4] 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

108. One of ordinary skill would have found this element obvious in view of the teachings of Khan. As the extracted values are retrieved from “one or more relevant knowledge bases” that include “the Web or Internet,” one of ordinary skill would have understood that Khan discloses evaluating the received results using the same or similar parsing methods used to parse and evaluate the utterance “to determine if they are understood” (*evaluating the extracted values using the dynamic set of prior probabilities*) because the results received from “the Web or Internet” would be in a natural language format similar to the utterance. EX1004, 6:29-7:6, 18:14-18; *id.*, 3:5-7 (describing “Natural Language Format”), 11:30-16:26 (describing parsing the request so that “information may be extracted”); Section IX.A.1.b) (element [1.3]). Khan further discloses there may be “insufficient understood data to allow the service engine to fulfill the request,” and that “any other actions necessary to fulfill the user’s request may be performed,” including “automatically using previously stored user information” (*resolving ambiguous, incomplete, or conflicting information*). EX1004, 22:18-19, 18:24-28. For example, Khan discloses that “the intelligent agent might identify airport preference” by using statistical information regarding user history. EX1004, 18:8-13 (identifying “that the

user departed on ninety percent of previous recorded flights from LaGuardia Airport in New York” to fill in *incomplete information*).

109. Further, one of ordinary skill knew that such a dynamic set of prior probabilities would be used to resolve ambiguities in extracted values of received results, where alternatives of the values are “listed and ranked by probabilities that may be learnable from data.” EX1011, 707; *id.*, 706-720 (describing text normalization techniques to address a “potential ambiguity” in text for a text-to-speech system); *see also* EX1010, 3-5 (explaining that “most or all tasks in speech and language processing can be viewed as resolving ambiguity at one of those levels,” such as phonetics and phonology, morphology, syntax, semantics, pragmatics, and discourse), 6 (“almost any speech and language processing problem can be recast as: ‘given N choices for some ambiguous input, choose the most probable one’”); EX1017, 7:6-22 (describing normalization of scraped information, because “a search for ‘XGA’ will not retrieve as a ‘hit’ data for a laptop computer in which screen size is specified as ‘1024x768,’ even though these two terms refer to the identical type of display”); EX1006, 22:16-26 (describing filtering search results and converting formats of retrieved data into an appropriate format for output). Thus, one of ordinary skill would have been motivated to implement Khan to evaluate the extracted values using the dynamic set of prior probabilities, wherein evaluating the extracted values includes resolving ambiguous, incomplete, or

conflicting information, at least because doing so was known in the art for resolving ambiguities during natural language processing, and further because Khan discloses to do so.

e) **Element [9.5]**

[9.5] 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. Khan teaches this element. Khan discloses selecting “only the least expensive fare (or only a few of the least expensive fares)” to “return[] in the response” (*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*). EX1004, 19:5-12.

9. **Claim 12**

a) **Element [12.1]**

[12.1] The method according to claim 9, wherein determining the one or more best responses to the at least one question further includes:

111. Khan renders claim 9 obvious and teaches this element. *See* Section IX.A.8 (claim 9).

b) **Element [12.2]**

[12.2] determining that the dynamic set of prior probabilities or fuzzy possibilities has failed to resolve the ambiguous, incomplete, or conflicting information; and

112. Khan teaches this element. Khan discloses that “the intelligent agent prompts the user to supply any essential information that is not understood” (*determining that the dynamic set of prior probabilities has failed to resolve the ambiguous, incomplete, or conflicting information*). EX1004, 17:31-33. As Khan also teaches that “any other actions necessary to fulfill the user’s request may be performed” after receiving and processing the information from the knowledge bases, one of ordinary skill would have understood that Khan determines whether the information is sufficient “to fulfill the user’s request,” which includes determining whether the information is understood or otherwise complete enough for the intelligent agent to act. EX1004, 18:24-25. For example, Khan discloses an example of automatically booking a flight “if the user had requested that the service engine reserve the least expensive seat on the applicable flight,” where sufficient “previously stored user information” is present for the intelligent agent to successfully do so. EX1004, 18:25-28. In this scenario, if the intelligent agent were unable to determine whether the user wanted to book the cheapest flight out of Kennedy or LaGuardia airport in New York City, it would do so by determining that it was unable to use the dynamic set of prior probabilities or fuzzy possibilities to resolve any ambiguous, incomplete, or conflicting information necessary to complete the purchase. EX1004, 17:8-10 (“If all of the required variables are not available or have low associated confidence levels, the intelligent agent will initiate

a dialog with the user to obtain further information”); *see also* EX1018, 23:32-37 (“Ambiguity may result from more than one correct interpretation of a submission (for example, a request for the flight schedules leaving New York City can be interpreted as leaving Kennedy or La Guardia airport and more information from the submitter would be required to resolve the ambiguity).”).

c) Element [12.3]

[12.3] resolving the ambiguous, incomplete, or conflicting information by requesting the user to provide additional information for resolving the ambiguous, incomplete, or conflicting information.

113. Khan teaches this element. Khan discloses “prompting the user with a question relating to the request” when “the evaluated request contains insufficient understood data to allow the service engine to fulfill the request” (*resolving ambiguous, incomplete, or conflicting information by requesting the user to provide additional information for resolving the ambiguous, incomplete, or conflicting information*). EX1004, 22:18-19; *see also id.*, 17:31-18:8 (“the intelligent agent prompts the user to supply any essential information that is not understood”), 18:24-33 (“any other actions necessary to fulfill the user’s request may be performed,” including “request[ing] information automatically”).

10. Claim 13

a) Element [13.1]

[13.1] The method according to claim 9, wherein presenting results of the processed request to the user includes:

114. Khan renders claim 9 obvious and teaches this element. *See* Section IX.A.8 (claim 9).

b) Element [13.2]

[13.2] selecting a format template for presenting the one or more best responses;

115. Khan teaches this element. Khan discloses selecting “[t]he content and format of this response ... based on the interface through which the response will be returned to the user” (*selecting a format template for presenting the one or more best responses*). EX1004, 19:6-7; *id.*, 19:7-12 (describing different format templates for different interfaces), 19:13-17 (describing packaging the response), 7:15-19 (describing “assembl[ing] the output of the smart search module into a context specific response based on user preferences and the communications medium used by the user,” followed by “packag[ing] the response” and “format conversion”); *see also* EX1011, 894-901 (describing “Response Content Generation” with templates and speech output).

c) Element [13.3]

[13.3] selecting a personality for presenting the one or more best responses;

116. One of ordinary skill would have found this element obvious in view of the teachings of Khan. Khan discloses presenting the results, such as “the least expensive fare (or only a few of the least expensive fares)” (*the one or more best responses*), via a speech/voice interface with “content and format” based on the voice interface. EX1004, 19:5-17; *id.*, 7:15-17. Khan does not provide further implementation details regarding how the response is assembled and “packaged into a collection of statements to be played to the user” via the voice interface. EX1004, 19:14-17.

117. One of ordinary skill understood, however, that standard text-to-speech (TTS) systems include a variety of voices corresponding to different personalities. *E.g.*, EX1019, 3:50-52 (explaining that prosody “parameters may be manipulated to create voice personalities” and that a TTS system “is supplied with nine different ‘Voices’ or personalities”); EX1027, 4:12-14 (disclosing a “voice user interface ... that includes multiple personalities”), 3:24-36 (defining “‘personality’ as used in the context of a voice user interface”); EX1010, 120-136 (discussing TTS systems). One of ordinary skill knew that a “speaking style of the voice ... can impart an overall tone to a communication,” where the speaking style (*personality*) reflects character and emotion. EX1011, 743-745 (depicting a module for prosody generation in text-to-speech systems, with a selected “Speaking Style” as an input to the module). Skilled artisans also knew that “fairly reliable tools exist[ed]” for the task of

predicting prosody for text-to-speech synthesizers, where prosody is “influenced by the *personality* of the speaker.” EX1016, 5:26-36 (emphasis added). Prosody is a combination of “physical, phonetic effects ... employed to express attitude, assumptions, and attention,” by varying pauses, pitch, rate/relative duration, and loudness. EX1011, 739-740. One of ordinary skill knew that if text “is spoken on a constant, uniform pitch with no pauses, or with uniform pauses between words, it sounds highly unnatural.” EX1011, 740. Hartono teaches selecting a personality for an agent to communicate with a user, and thus “express similar levels of assertiveness and/or responsiveness by varying the words used, the speed at which the words are delivered, the tone and/or the pitch of the words.” EX1007, 33:1-10; *id.*, 32:14-25 (“The output of the natural language agent 102 is preferably adaptive to the personality of the user 106 by first identifying the personality type, personality trait or characteristics of the user and utilizing that identification for responding to the user.”). One of ordinary skill thus would have been motivated to implement Khan with selectable personalities for presenting the results to the user for the presented results to sound more natural to the user, as known in the art such as Hartono, and would have reasonably expected success doing so because existing systems already provided personality modules.

d) **Element [13.4]**

[13.4] determining an order for presenting the tokens contained in the response;
and

118. Khan teaches this element. As discussed above in Section IX.A.8.b), Khan renders obvious determining one or more tokens required to formulate a response to the question. Khan also discloses assembling the results into a response that is returned to the user via the voice interface. EX1004, 19:5-18. One of ordinary skill would have understood that assembling the results into a response includes selecting a template with attributes (*tokens*) “interspersed between words of natural language.” EX1008, 1:50-53 (explaining in 2000 that “[t]he majority of current conversational systems perform natural language generation (NLG) with templates”). One of ordinary skill knew that a template could provide “a fixed way of rendering [a set of attribute-value pairs] into natural language,” and would have understood selecting a template as *determining an order for presenting the tokens contained in the response* because selecting the template sets the order for presenting the tokens. EX1008, 1:59-61. One of ordinary skill also knew that “[t]he ability to reorder words and attributes is important for the purposes of generating natural language,” and that templates can be dynamically generated, with a “dynamic reordering of words and attributes such that a most appropriate phrase for a given

situation is found” (*determining an order for presenting the tokens contained in the response*). EX1008, 2:66-3:12. For example, Ratnaparkhi explains that:

For example, in a conversation concerning the domain of air travel where information on departure times of flights out of Boston is being sought, the phrase “A flight from Boston that leaves at 3 PM” might be a more appropriate response that is more akin to what would be said in natural language than the phrase “A 3PM flight that leaves from Boston” even though it expresses the same information.

EX1008, 2:67-3:7; *id.*, 3:18-52 (describing using grammar rules and a scoring system to generate natural language phrases for attribute-value pairs). One of ordinary skill would have been motivated to implement Khan with dynamic reordering of tokens, in order to achieve a natural language output while maintaining flexibility in the types of requests that a user may submit. EX1004, 1:16-17 (teaching that system should provide better results than “canned responses”), 3:8-10 (describing requests); EX1008, 7:63-67 (explaining that dynamic reordering “eliminates the need for a programmer to specify every possible ordering of various words and phrase fragments” for a natural language generation system). One of ordinary skill would have been further motivated to implement Khan with dynamic reordering and thus achieve more natural responses in order to facilitate effective human-computer interaction in “noisy, or stressful situations” where “the

requirement for more natural sounding speech is thought to be important” (EX1020, 1:29-32), because Khan teaches using automobile communication systems and smart remote controls as voice interfaces, where the environment may be noisy or stressful. EX1004, 1:22-26, 1:31-32, 2:15-25. The skilled artisan would have reasonably expected to succeed in doing so because Khan already provides the elements for doing so (e.g., response assembly module, the dynamic set of prior probabilities) and the implementation does not require any reworking of the Khan system. Thus, this element is obvious in view of Khan.

e) Element [13.5]

[13.5] performing variable substitutions and transformations on the best values selected for the tokens, wherein performing the variable substitutions and transformations includes using the dynamic set of prior probabilities or fuzzy possibilities to vary a terminology of the best values.

119. One of ordinary skill would have found this element obvious in view of the teachings of Khan. The response assembly module of Khan prepares the “context specific response based on user preferences and the communications medium used by the user.” EX1004, 7:15-17; *id.*, 19:5-7. To make the response context specific and suitable for voice output, skilled artisans knew that preparing text for text-to-speech (TTS) involved performing text normalization, which is “the conversion from the variety of symbols, numbers, and other nonorthographic entities of text into a common orthographic transcription suitable for subsequent phonetic conversion”

(performing variable substitutions and transformations on the best values selected for the tokens). EX1011, 692 (emphasis added); *see also* EX1008, 4:38-42 (describing substituting attributes / tokens “with their respective natural language values”).

120. One of ordinary skill further knew that the normalization of text including abbreviations, for example, is “context driven and probabilistic in nature” *(using the dynamic set of prior probabilities or fuzzy possibilities to vary a terminology of the best values)*. EX1011, 693. For example, if the user requested an address and the best value obtained by the intelligent agent was “123 Example Dr.,” one of ordinary skill would have understood that the response assembly module of Khan would use the dynamic set of prior probabilities or fuzzy possibilities to substitute “Dr.” with “Drive” because this would be more probable than “Doctor” in the address or travel domain. *E.g.*, EX1011, 693 (Table 14.1 depicting ambiguity of “Dr. Smith” having alternatives of “doctor or drive”). Similarly, one of ordinary skill knew that “*varying the words used*” would make the computer-generated response seem more natural, as taught by Hartono. EX1007, 33:1-10 (emphasis added). A skilled artisan implementing Khan thus would have used the known techniques of variable substitutions and transformations, including by varying a terminology, with the predictable result of making responses more natural and less like the “canned

responses” disparaged by Khan. EX1004, 1:16-17. Thus, this element is obvious in view of Khan and the knowledge of one of ordinary skill.

11. Claim 14

a) Element [14.1]

[14.1] The method according to claim 13, wherein presenting results of the processed request to the user further includes:

121. Khan renders claim 13 obvious and teaches this element. *See* Section IX.A.10 (claim 13).

b) Element [14.2]

[14.2] formatting the best responses in accordance with the selected format template, wherein a text to speech engine reads the formatted best responses to create a system generated speech utterance that presents the best responses to the user.

122. Khan teaches this element. Khan discloses selecting a format template for presenting the one or more best responses, which includes selecting a “format of this response” for output, and further assembling the results into the response (*formatting the best responses in accordance with the selected format template*). EX1004, 19:5-12; Section IX.A.10.b) (element [13.2]). Khan further discloses that “the assembled response is packaged based on the particular interface through which the response will be transmitted,” where responses returned via voice interfaces are “packaged into a collection of statements to be played to the user” (*wherein a text to speech engine reads the formatted best responses to create a system generated*

speech utterance that presents the best responses to the user). EX1004, 19:13-18; *id.*, 7:15-19. Skilled artisans would have understood that the output packaging module that “packages the response for transmission to the user” (EX1004, 7:17-19) included a text-to-speech (TTS) engine for generating the collection of statements from the assembled, formatted response, because “a text-to-speech system that conveys spoken information” is a basic subsystem of a spoken language system. EX1011, 4, 6-7 (depicting basic architecture of a TTS system and explaining that “[t]he development of TTS synthesis can be traced back to the 1930s”); EX1007, 23:4-13 (Hartono describing a “text-to-speech synthesizer” using “commercial off the shelf technology”). One of ordinary skill thus would have found it obvious to use a TTS engine to generate the utterance “to be played to the user,” because Khan suggests doing so and TTS engines were a basic element of a natural language system. EX1004, 19:13-18.

**X. GROUND 2: CLAIMS 1,2, 4-9, AND 12-14 ARE
OBVIOUS OVER KHAN IN VIEW OF RICCARDI**

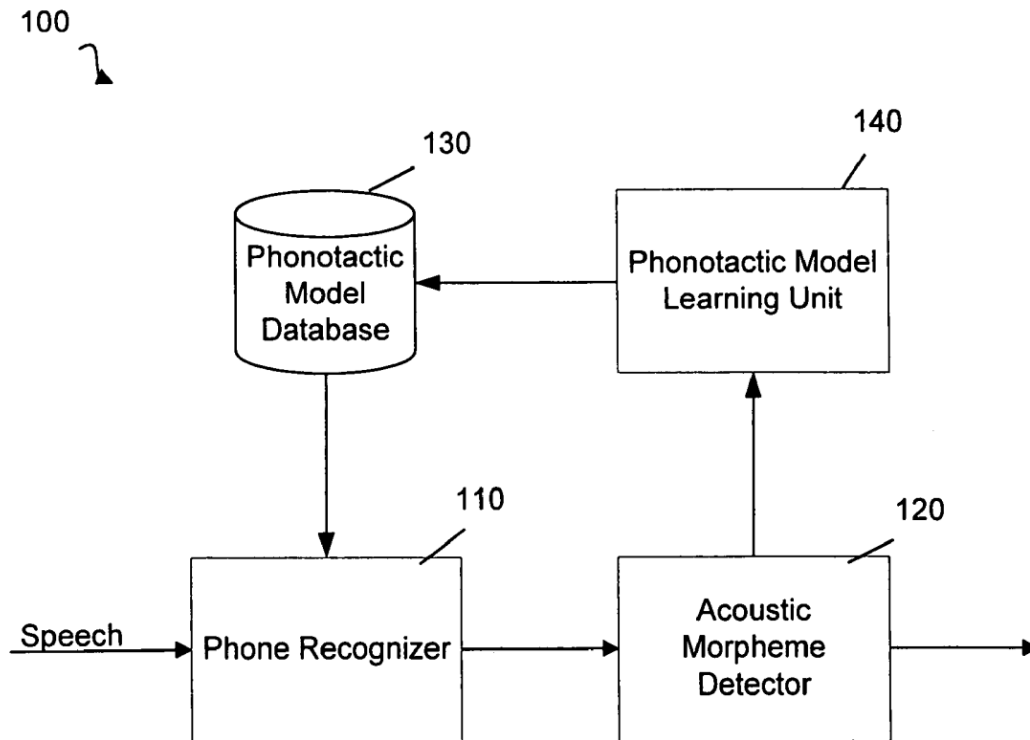
123. To the extent Patent Owner argues that it was not within one of ordinary skill’s knowledge to maintain a dynamic set of prior probabilities usable at each stage of processing, Ground 2 expressly relies on Riccardi to confirm the obviousness of this element, and of the Challenged Claims. *See* Section IX.A.1.b) (element [1.3]).

A. Riccardi (EX1005)

124. Riccardi is a patent that issued from a U.S. patent application filed September 21, 2001, and claims priority to a provisional application filed September 27, 2000, and thus I understand that it qualifies as prior art under at least Section 102(e). EX1005, Cover. Riccardi is a continuation-in-part of applications filed October 18, 2000, claiming priority to a provisional application filed on November 5, 1999. *Id.* Riccardi is analogous art to the '209 patent partly because they are both in the same field of endeavor of speech recognition and understanding. EX1001, 1:38-48; EX1005, 1:22-23.

125. Riccardi discloses a method for speech recognition based on a phonotactic model with “unsupervised learning of acoustic and lexical units from out-of-domain speech data.” EX1005, 2:24-26. In Riccardi, a “phone recognizer 110” uses a phonotactic model to recognize “the existence of one or more phone in the user’s input speech,” an “acoustic morpheme detector 120 detects the acoustic morphemes present in the recognized phones,” and a “phonotactic model learning unit 140 generates a new phonotactic model based on the detected acoustic morphemes,” which is then “use[d] by the phone recognizer 110 in recognizing phone sequences in the user’s next utterance.” EX1005, 3:58-61, 4:4-6, 4:21-24, 6:9-15, FIG. 1. This allows for “fast adaptation of language probabilities to a new

domain,” where “the relevant language statistics of task classification systems may be learned.” EX1005, 2:26-30.



EX1005, FIG. 1.

B. The Khan-Riccardi Combination

126. One of ordinary skill would have been motivated, with a reasonable expectation of success, to implement the service engine of Khan with the adaptive learning methods of Riccardi to achieve improved recognition accuracy and a flexible speech recognition system. The combination of Khan and Riccardi includes the service engine of Khan with a speech recognition system that recognizes words in user speech, as taught by Khan (EX1004, 8:30-9:7) and Riccardi (EX1005, 3:34-

5:22), using a probabilistic model that dynamically improves over time, as taught by Riccardi (EX1005, 2:24-30, 6:10-39).

127. As I explained above for Ground 1, Riccardi discloses an “adaptive spoken dialog system,” which “transforms a prior probability distribution from one domain to another without speech transcription (on-line adaptation),” thus allowing the system to “react and recover to novel events” and providing improved speech recognition “in the user’s next utterance.” EX1005, 6:9-20; *id.*, 6:25-7:19 (describing dynamic adaptation using “at least two stochastic modeling processes”). One of ordinary skill would have been motivated to include this dynamic speech recognition functionality in the Khan service engine, which complements the adaptive learning of requests taught by Khan, so that the service engine “adapt[s] to unseen speech and language events,” thereby improving the accuracy and flexibility of the service engine. EX1005, 2:22-53.

128. One of ordinary skill would have reasonably expected success implementing this functionality in the Khan service engine because (1) Khan already provides a set of prior probabilities for speech recognition and parsing (*see above* Section IX.A.1.b)), (2) Khan discloses using “generic speech recognition software” (EX1004, 4:33-5:7), indicating that its system was designed to accommodate various speech processing modules, including the adaptive speech recognition of Riccardi, and (3) skilled artisans knew how to update probabilistic models used for natural

language processing (EX1024, 13:44-14:9). For at least these reasons in combination with the reasons provided above in Ground 1, the Challenged Claims are further obvious over Khan and Riccardi.

**XI. GROUND 3: CLAIMS 1,2, 4-9, AND 12-14 ARE
OBVIOUS OVER KHAN IN VIEW OF RICCARDI AND IIZUKA**

129. To the extent Patent Owner argues that extracting one or more values for the determined tokens from the asynchronously received results is not obvious in view of Khan and the knowledge of one of ordinary skill, Ground 3 expressly relies on Iizuka (discussed as a background reference in Ground 1) to confirm the obviousness of this element, and of the Challenged Claims. *See* Section IX.A.8.c) (element [9.3]).

130. Iizuka (EX1006) issued from a patent application filed June 9, 1999, and therefore qualifies as prior art under Section 102(e). EX1006, Cover.

131. As I explained above in Ground 1, Iizuka teaches “extracting required information item by item from different HTML documents” located via a search query, where the documents have different locations, structures, and attributes. EX1006, 4:60-65; *id.*, 4:44-51, 12:18-32, 22:2-31. Khan teaches using “the key information provided in the request to conduct information retrieval,” and a skilled artisan would have naturally turned to Iizuka for implementation details for “flexibly retrieving necessary information” from various information sources, regardless of the structure or style of the original information sources, as taught by Iizuka, because

this disclosure is consistent with the teachings of Khan regarding “retriev[ing] information” from remote databases “[d]epending on the architecture and format of each database.” EX1006, 29:45-59; EX1004, 18:21-23. One of ordinary skill would have easily incorporated the teachings from Iizuka with the obvious implementation of Khan and Riccardi described in Grounds 1 and 2 by programming the “Smart Search Module” of the service engine, which already “contains multiple algorithms designed to interface with and search across multiple knowledge bases.” EX1004, 7:12-14, FIG. 1B. For at least these reasons in combination with the reasons provided above in Grounds 1 and 2, the Challenged Claims are further obvious over Khan, Riccardi, and Iizuka.

XII. GROUND 4: CLAIMS 1,2, 4-9, AND 12-14 ARE OBVIOUS OVER KHAN IN VIEW OF RICCARDI, IIZUKA, AND HARTONO

132. To the extent Patent Owner argues that selecting personalities and varying language in natural language responses, along with using text-to-speech synthesizers, is not obvious in view of Khan and the knowledge of one of ordinary skill, Ground 4 expressly relies on Hartono (discussed as a background reference in Ground 1), to confirm the obviousness of these elements, and of the Challenged Claims. *See* Sections IX.A.10.c) (element [13.3]), IX.A.10.e) (element [13.5]), IX.A.11.b) (element [14.2]).

133. Hartono is an international patent application filed August 20, 1999, and published March 2, 2000, and thus is prior art at least under 102(a). EX1007,

Cover. Much like Khan, Hartono discloses an “adaptive natural language interface” where agents fulfill user requests. EX1007, 1:19-20, 4:9-13. As discussed above, Hartono discloses selecting a personality used by an agent communicating with the user, and “varying the words used” in the natural language response. EX1007, 32:14-33:13. Hartono explains that this allows that this allows responses to be “adaptive to the personality of the user,” and “express[es] similar levels of assertiveness and/or responsiveness by varying the words used, the speed at which the words are delivered, the tone and/or the pitch of the words.” EX1007, 32:14-17, 33:4-10. Hartono further discloses a “text-to-speech synthesizer” for generating a speech utterance for the user. EX1007, 23:4-13. Hartono discloses “us[ing] commercial off the shelf technology to communicate messages to the user 106 by speech.” EX1007, 23:7-9.

134. One of ordinary skill implementing the service engine of Khan would have been motivated to incorporate the selection of a personality, the varying of terminology in a response, and the use of text-to-speech synthesizers as taught by Hartono for the responses provided to the user in order to achieve more natural responses and enhance the communication between the user and the service engine. EX1007, 1:25-2:19 (describing goal of “[m]aking computers more user-friendly” and “[n]atural or spoken language [as] an efficient method ... to communicate”). Further, the skilled artisan would have naturally turned to Hartono for

implementation details on generating the response “to be played to the user” (EX1004, 19:14-17), because the adaptive teachings in Hartono inform how to “deliver[] the output to the user ... using simplified emotional response.” EX1007, 33:1-4. The skilled artisan would have reasonably expected success because these features were available as “commercial off the shelf technology” that “utilize[s] intonation to make the synthesized speech sound more natural to the user,” and could easily be incorporated into the service engine of Khan. EX1007, 23:7-10.

135. For these same reasons, one of ordinary skill would have been motivated to incorporate the teachings of Hartono into the obvious implementation of Khan, Riccardi, and Iizuka described in Grounds 1-3. For at least these reasons, the Challenged Claims are further obvious in view of Khan, Riccardi, Iizuka, and Hartono.

I, Paul Jacobs, do hereby declare and state, that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: 8/6/2025, 2025


Signature

Paul S. Jacobs

Paul@jaketech.com
mobile: (703) 307-0467

4105 Faith Ct.
Alexandria VA 22311

Education

Ph.D. in Computer Science
S.M. in Applied Mathematics
A.B. in Applied Mathematics

University of California at Berkeley (1985)
Harvard University (1981)
Harvard University (1981)

Work History

Founder and President, Jake Technologies, Inc. (7/02 -)

- Founded technology services company focused on strategic technology development, product evaluation, and advising corporate counsel and law firms on issues related to intellectual property. Serve as expert consultant in patent litigation, research, and related matters.
- Appointed by the office of the Secretary of Commerce to the Patent Public Advisory Committee, advising the Under Secretary and Director of the U.S.. Patent and Trademark Office from 2012 – 2015.

Chief Technology Officer, Primus Knowledge Solutions, Inc. (6/01 – 7/02)

- Played senior executive role in integration of AnswerLogic into Primus, a publicly-held CRM software company focusing on knowledge capture and delivery in support center and web self-service applications.
- As CTO, responsible for technology “evangelism” both inside of and outside of Primus, including articulating company strategy and vision to analysts, customers, and prospects and working with product and product management teams.

President, Chief Operating Officer, and Chief Technology Officer, AnswerLogic, Inc. (10/99 – 5/01)

- Joined start-up at close of Series A financing, led through Series B financing and initial product launch and made sale of company to Primus Knowledge Solutions.
- Led technology and general management of over 90 employees, including over 50 linguists and over 30 engineers. Only senior executive retained after Primus acquired AnswerLogic.

Managing Vice President, Electronic Commerce, SRA International (9/98 – 10/99)

- Managed NetOwl product line and related IsoQuest technologies for parent company; SRA derived over \$10 million in product licenses and millions more in associated services from NetOwl and related products, with over 100 customers. SRA, which soon went public, was ranked in Business Week’s Top 20 Private IT firms in 1999, cited for e-mail software based on NetOwl.

President and Chief Executive Officer, IsoQuest, Inc. (7/96 – 9/98)

- Founded and led advanced Internet-focused knowledge management software company, backed by SRA International. IsoQuest’s main product, NetOwl, became the leading engine for automatic information extraction as well as forming the basis for the parent company’s product offering. In 1998, with IsoQuest at break-even, SRA re-integrated the subsidiary for strategic reasons.
- Responsible for all major financial, strategy, staffing and management decisions
- Conceived product strategy, launched and sold over \$2 million of new product (NetOwl) to 20 clients in first two years; brought company to break-even.

APPENDIX A

- Personally guided marketing, sales, and product management until executives in each area were hired.
- Landed major accounts, including Thomson Corporation, Knight Ridder, Bloomberg, Lexis/Nexis, Disclosure, NewsEdge, and Infoseek.

Director of Product Marketing, SRA International (7/95 – 6/96)

- Led launch of (\$200M) consulting firm's first software product (NameTag)
- Developed strategic marketing, branding, product management, and marketing communications programs, hired and managed agencies and contractors and led product marketing initiatives
- Successfully presented business plan and laid groundwork for formation of new company (IsoQuest)

Director of Media Information Technologies, SRA International (8/94 – 6/95)

- Developed and implemented long-range technology transfer plan
- Supported SRA business units (media, intelligence, others) in new initiatives and major sales opportunities
- Led search engine re-design and deployment for new venture (Picture Network International)

Computer Scientist, GE Corporate Research and Development (8/85-8/94)

- Led GE's initiatives in natural language text processing. Built world's leading team in automatic information extraction from free text. Received numerous management awards. Whitney Gallery of Technical Achievers
- Principal investigator for four years in U.S. government's Tipster program (Tipster was cited with the "Hammer" award as a national reinvention laboratory)
- Proposed, won, and deployed advanced text management projects in GE Information Services, Aircraft Engines, Aerospace, Capital Services, and external contracts with ARPA and other agencies.
- Completed GE Technical Management Course
- Spent one year as Visiting Associate Professor at University of Pennsylvania (Computer and Information Sciences, Institute for Research in Cognitive Science)
- Consulted with Infonautics, Inc. (start-up in Wayne, PA) – helped map out initial business plan and technical approach to Homework Helper, their first product.

Professional Activities

Author of over 50 published journal articles and technical papers. Holder of two U.S. Patents.

Author of *Text-Based Intelligent Systems* (Lawrence Erlbaum Associates, 1992).

Adjunct Professor, College of Information Studies (iSchool), University of Maryland. Currently teach Information Architecture. Delivered 2008 commencement address.

Member of Patent Public Advisory Committee (PPAC), US Dept. of Commerce, 2012-2015

Member of USACM (Public Policy Committee of Major Computer Trade Association)

Member of Intellectual Property Committee of the USACM

Executive Committee, Association for Computational Linguistics, 1997-9

Other Activities

Accomplished skier, distance runner. Iron Man Triathlete. Winner, Golden Gate Marathon, 1983, 1984. Enjoy exotic travel, adventure sports, hiking with my family.

APPENDIX A

PAUL S. JACOBS

PUBLICATIONS, PROFESSIONAL AND SCHOLARLY ACTIVITIES

Refereed Journal Articles

P. Jacobs. PHRED: A generator for natural language interfaces. *Computational Linguistics* 11 (4), pp. 219-242, 1985. Also appearing in L. Bolc and D. McDonald (eds.), *Natural Language Generation Systems*, Springer Verlag, 1988

P. Jacobs. Knowledge intensive natural language generation. *Artificial Intelligence* 33 (3), pp. 325-378, 1987.

L. Rau and P. Jacobs. $NL \cap IR$: Natural language for information retrieval. *International Journal of Intelligent Systems* 4 (3), pp. 319-343, 1989.

L. Rau, P. Jacobs, and U. Zernik. Information extraction and text summarization using linguistic knowledge acquisition. *Information Processing and Management* 25 (4), pp. 419-428, 1989.

P. Jacobs and L. Rau. SCISOR: A system for extracting information from on-line news. In *Communications of the Association for Computing Machinery* 33 (11), pp. 88-97, November 1990.

P. Jacobs. TRUMP: A transportable language understanding program. *International Journal of Intelligent Systems* 7 (3), pp. 245-276, 1992.

P. Jacobs and L. Rau. Innovations in text interpretation. *Artificial Intelligence* 63 (1-2), pp. 143-191, 1993.

P. Jacobs. Using statistical methods to improve knowledge-based news categorization. *IEEE Expert*, 8(2), pp. 13-23, Apr. 1993.

Refereed Conference Proceedings

P. Jacobs. Generation in a natural language interface. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, 1983.

P. Jacobs and L. Rau. Ace: Associating language with meaning. In *Proceedings of the Sixth European Conference on Artificial Intelligence*, Pisa, Italy, 1984. Reprinted in T. O'Shea (ed.), *Advances in Artificial Intelligence*, North Holland, 1986.

P. Jacobs. The KING natural language generator. In *Proceedings of the Seventh European Conference on Artificial Intelligence*, Brighton, England, 1986. Reprinted in B. du Boulay, D. Hogg, and L. Steels (eds.), *Advances in Artificial Intelligence II*, North Holland, 1987.

P. Jacobs. Knowledge structures for natural language generation. In *Proceedings of the 11th International Conference on Computational Linguistics*, pp. 554-559, Bonn, 1986.

P. Jacobs. Language analysis in not-so-limited domains. In *Proceedings of the Fall Joint Computer Conference*, Dallas, 1986.

APPENDIX A

- P. Jacobs. A knowledge framework for natural language analysis. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, Milan, 1987.
- D. Besemer and P. Jacobs. FLUSH: A flexible lexicon design. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pp. 186-193, Palo Alto, 1987.
- P. Jacobs. Achieving bidirectionality. In *Proceedings of the 12th International Conference on Computational Linguistics*, pp. 267-274, Budapest, 1988.
- P. Jacobs. Concretion: Assumption-based understanding. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, 1988.
- P. Jacobs. Why text planning isn't planning. In *Proceedings of the AAAI Workshop on Text Planning and Generation*, St. Paul, Minnesota, 1988.
- P. Jacobs and U. Zernik. Acquiring lexical knowledge from text: A case study. In *Proceedings of the National Conference on Artificial Intelligence*, St. Paul, Minnesota, 1988.
- P. Jacobs and L. Rau. Natural language techniques for intelligent information retrieval. In *Proceedings of the 11th International Conference on Research and Development in Information Retrieval*, Grenoble, France, 1988.
- P. Jacobs. Extensible natural language interfaces. In *Proceedings of the 17th ASIS Mid-Year Meeting*, Ann Arbor, 1988.
- P. Jacobs and L. Rau. A friendly merger of conceptual expectations and linguistic analysis in a text processing system. In *Proceedings of the Fourth IEEE Conference on Artificial Intelligence Applications*, pp. 351-356, San Diego, 1988.
- L. Rau and P. Jacobs. Integrating top-down and bottom-up strategies in a text processing system. In *Proceedings of the Second Conference on Applied Natural Language Processing (ANLP)*, pp. 129-135, Austin, 1988.
- P. Jacobs and L. Rau. The GE NLToolset: A software foundation for intelligent text processing. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki, 1990.
- U. Zernik and P. Jacobs. Tagging for learning: Collecting thematic relations from Corpus. In *Proceedings of the 13th International Conference on Computational Linguistics*, pp. 34-39, Helsinki, 1990.
- P. Jacobs, G. Krupka, L. Rau, N. Sondheimer, and U. Zernik. Generic text processing. In *Proceedings of the Third DARPA Speech and Natural Language Workshop*, 359-364, Somerset, PA, 1990.
- P. Jacobs. To parse or not to parse: Relation-driven text skimming. In *Proceedings of the 13th International Conference on Computational Linguistics*, pp. 194-198, Helsinki, 1990.

APPENDIX A

L. Rau and P. Jacobs. Creating segmented databases from free text for text retrieval. In *Proceedings of the 14th International Conference on Research and Development in Information Retrieval*, pp. 337-346, Chicago, 1991.

P. Jacobs, G. Krupka, and L. Rau. Lexico-semantic pattern matching as a companion to parsing in text understanding. In *Proceedings of the Fourth DARPA Speech and Natural Language Workshop*, Asilomar, CA, 1991.

P. Jacobs. Parsing run amok: Relation-driven control for text analysis. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 315-321, San Jose, 1992.

P. Jacobs. Joining statistics with NLP for text categorization. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pp. 178-185, Trento, Italy, 1992.

P. Jacobs, G. Krupka, and L. Rau. A Boolean approximation method for query construction and topic assignment in TREC. In *Proceedings of the Second Annual Symposium on Document Analysis and Retrieval*, Las Vegas, 1993.

P. Jacobs. Word sense acquisition for multilingual text interpretation. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94)*, Vol. II, pp. 665-671, 1994.

P. Jacobs. Browsing vs. Surfing: The Next Generation of Search and Retrieval. In *Proceedings of the National On-Line Conference*. New York, 1998.

Books and Book Chapters

P. Jacobs. KING: A knowledge-intensive natural language generator. In G. Kempen (ed.), *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kluwer Academic Publishers, 1987.

P. Jacobs. Two hurdles for natural language systems. In R. Freedle, (ed.), *Artificial Intelligence and the Future of Testing*, pp. 257-277, Lawrence Erlbaum Associates, Hillsdale, NJ, 1990.

P. Jacobs. Making sense of lexical acquisition. In U. Zernik, (ed.), *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*, pp. 29-44, Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.

P. Jacobs. Integrating language and meaning in structured inheritance networks. In J. Sowa, (ed.), *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, pp. 527-542, Morgan Kaufman Publishers, San Mateo, CA, 1991.

P. Jacobs. Text power and intelligent systems. In *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, pp. 1-8, Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

P. Jacobs (ed). *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

APPENDIX A

P. Jacobs. Why Text Planning (Still) Isn't Planning. In A. Ortony, J. Slack, and O. Slack (eds.), *Communication from an Artificial Intelligence Perspective*, Springer Verlag, Heidelberg, 1992.

P. Jacobs. Information extraction. In R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue (eds.), *Survey of the State of the Art in Human Language Technology*, pp. 230-232, Cambridge University Press, 1997.

Other Published Articles

P. Jacobs. Generation in the UNIX Consultant System, in D. Roesner (ed.), *International Workshop on Language Generation*, Institut fuer Informatik, Universitaet Stuttgart, 1983.

P. Jacobs. Teaching language to computers. *IEEE Potentials*, October, 1986.

P. Jacobs and L. Rau. Software for intelligent text processing. In *Proceedings of the 13th International Conference on Computational Linguistics (project notes)*, Helsinki, Vol. 3, 1990, pp 373-375.

G. Krupka, L. Iwanska, P. Jacobs, and L. Rau. GE NLToolset test results and analysis. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, San Diego, 1991.

G. Krupka, P. Jacobs, L. Rau and L. Iwanska. Description of the GE NLToolset system as used in MUC-3. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, San Diego, 1991.

P. Jacobs, G. Krupka, and L. Rau. The right tools for the job: Multiple language analysis strategies in the GE NLToolset. In *Proceedings of the Workshop on Fully Implemented Natural Language Systems*, Trento, Italy, March, 1992.

P. Jacobs, G. Krupka, L. Rau, T. Kaufmann and M. Mauldin. GE-CMU: Description of the TIPSTER/SHOGUN system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, McLean, VA, 1992.

L. Rau, G. Krupka, P. Jacobs, I. Sider and L. Childs. GE NLToolset test results and analysis. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, McLean, VA, 1992.

P. Jacobs, G. Krupka, L. Rau, M. Mauldin, T. Mitamura, T. Kitani, I. Sider, and L. Childs. GE-CMU: description of the SHOGUN system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pp. 109-120, 1993.

P. Jacobs, G. Krupka, L. Rau, M. Mauldin, T. Mitamura, T. Kitani, I. Sider, and L. Childs. The TIPSTER/SHOGUN Project. In *Proceedings of the TIPSTER Text Phase I Final Meeting*, pp. 209-221, Morgan Kaufmann, 1993.

P. Jacobs. GE in TREC-2: Results of a Boolean approximation method for routing and retrieval. In *Proceedings of the Second Text Retrieval Conference (TREC-2)*, D. Harman (ed.), pp. 191-199, NIST Special Publication 500-215, 1994.

APPENDIX A

P. Jacobs. Natural language processing: A brief history for skeptics. *ServerWorld*, February 2001.

Technical Reports Not Published Elsewhere

P. Jacobs. *A Knowledge-Based Approach to Language Production*. Ph. D. Thesis, University of California, Berkeley. Available as Computer Science Division Technical Report 86/254, 1985.

P. Jacobs. A cooperative natural language directory system. General Electric Corporate Research and Development, Report 88CRD025, 1988.

Research Grants

Principal investigator of Tipster Phase I (GE, 1991, \$2.8 million) and Phase II (GE, 1993) \$10s of millions in commercial financing for advanced research and development from 1994 through 2000.

Professional Societies and Committees

Member of IEEE, ACM, AAAI, and ACL for many years.

Program committees:

- AAAI National Conference on Artificial Intelligence

- ACL Annual Conference

- ACL Conference on Applied Natural Language Processing

- ACM SIGIR Conference on Research and Development in Information Retrieval

- ARPA Speech and Natural Language Conference

- Others

Program chair, AAAI Symposium on Text-Based Intelligent Systems, 1990 (led to the publication of the book, *Text-Based Intelligent Systems* and to an increase in cross-disciplinary work in computational linguistics and information retrieval)

Program chair, ACL Conference on Applied National Language Processing, Stuttgart, Germany, 1994.

Executive Committee of the ACL, 1995-1998

USACM Public Policy Committee, Intellectual Property Subcommittee, 2005 -

U.S. Patents

5,243,520 Sense Discrimination System and Method, Sept. 7, 1993

5,251,129 Method for Automated Morphological Analysis of Word Structure, Oct. 5, 1993