

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

DIALECT, LLC,
Patent Owner

Case IPR2024-00748
U.S. Patent No. 8,015,006 B2
Issue Date: September 6, 2011

Title: SYSTEMS AND METHODS FOR PROCESSING NATURAL LANGUAGE SPEECH
UTTERANCES WITH CONTEXT-SPECIFIC DOMAIN AGENTS

DECLARATION OF CHRISTOPHER M. SCHMANDT

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

TABLE OF CONTENTS

	Page
I. INTRODUCTION AND QUALIFICATIONS44.....	12
A. Qualifications and Experience	12
B. Materials Considered	16
II. PERSON OF ORDINARY SKILL IN THE ART	18
III. STATEMENT OF LEGAL PRINCIPLES	20
A. Claim Construction.....	20
B. Obviousness (§ 103)	22
IV. THE '006 PATENT	28
A. Overview of the Specification.....	28
B. The Challenged Claims.....	31
V. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS.....	33
A. Brief Summary of Prior Art	33
1. Prior Art Common to All Grounds.....	33
(a) Coffman (EX1003)	33
(b) Kanevsky (EX1004)	38
(c) Kneser (EX1005).....	41
(d) Zadrozny (EX1006).....	42
2. Additional Prior Art for Certain Dependent Claims	45
(a) Maes '101 (EX1010).....	45
(b) Franceschi (EX1011)	46
(c) Chai (EX1012).....	48
(d) Redfern (EX1013).....	52
B. Ground 1: Claims 1-3 Are Obvious Over Coffman in View of Kanevsky, Kneser and Zadrozny.....	53

TABLE OF CONTENTS
(continued)

	Page
1. Claim 1	53
(a) “receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;” (1[a])	55
(b) “recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables, wherein recognizing the one or more words or phrases contained in the utterance includes:” (1[b])	63
(i) “dynamically updating the information in the one or more dictionary and phrase tables based on a dynamic set of prior probabilities or fuzzy possibilities;” (1[b][i])	71
(ii) “determining an identity associated with a user that spoke the utterance based on voice characteristics associated with the utterance; and” (1[b][ii])	78
(iii) “associating the one or more recognized words or phrases and a pronunciation associated with the one or more recognized words or phrases with the determined identity and the request contained in the utterance in response to the one or more recognized words or phrases satisfying a predetermined confidence level;” (1[b][iii])	81

TABLE OF CONTENTS

(continued)

Page

(c)	“parsing, at a parser coupled to the processing device, the one or more recognized words or phrases to determine a meaning associated with the utterance and a context associated with the request contained in the utterance, wherein the one or more recognized words or phrases are further associated with the determined context in response to the one or more recognized words or phrases satisfying the predetermined confidence level;” (1[c]).....	89
(d)	“formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context;” (1[d]).....	94
(e)	“processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance; and” (1[e])	108
(f)	“presenting the generated response to the utterance via the speech unit.” (1[f])	110
2.	Claim 2: “The method according to claim 1, wherein parsing the one or more recognized words or phrases to determine the context associated with the request includes:”	111
(a)	“matching the one or more recognized words or phrases to a set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities;” (2[a])	111
(b)	“scoring multiple possible contexts for the one or more recognized words or phrases matched to the set of keywords using a profile associated with the determined user identity, data content associated with the domain agent, or a context stack that includes one or more recent contexts; and” (2[b]).....	112

TABLE OF CONTENTS
(continued)

	Page
(c) “selecting one of the scored multiple possible contexts having a highest score that satisfies the predetermined confidence level to be the determined context.” (2[c])	116
3. Claim 3: “The method according to claim 2, wherein parsing the one or more recognized words or phrases to determine the context associated with the request further includes:”	116
(a) “requesting the user to verify the one or more recognized words or phrases in response to determining that none of the scored multiple possible contexts satisfy the predetermined confidence level; and” (3[a]).....	117
(b) “determining the context based on information that the user provides in one or more subsequent natural language speech utterances.” (3[b]).....	118
C. Ground 2: Claim 4 Is Obvious Over Ground 1 Prior Art (Coffman, Kanevsky, Kneser and Zadrozny) in Further View of Maes ’101 and Franceschi.....	118
1. Claim 4: “A method for processing natural language speech utterances with context-specific domain agents, comprising:”	118
(a) “receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;” (4[a])	119
(b) “recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables, wherein recognizing the one or more words or phrases contained in the utterance includes:” (4[b])	119

TABLE OF CONTENTS

(continued)

Page

(i)	“requesting a user that spoke the utterance to spell the one or more words or phrases in response to the one or more recognized words or phrases failing to satisfy a predetermined confidence level; and” (4[b][i]).....	119
(ii)	“updating the information in the one or more dictionary and phrase tables based on a phonetic alphabet spelling associated with the one or more words or phrases, wherein the user provides the phonetic alphabet spelling in one or more subsequent natural language speech utterances;” (4[b][ii]).....	129
(c)	“parsing, at a parser coupled to the processing device, the phonetic alphabet spelling associated with the one or more words or phrases to determine a meaning associated with the utterance and a context associated with the request contained in the utterance;” (4[c]).....	130
(d)	“formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context;” (4[d]).....	131
(e)	“processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance; and” (4[e])	131
(f)	“presenting the generated response to the utterance via the speech unit.” (4[f])	131
D.	Ground 3: Claim 5 Is Obvious Over Ground 1 Prior Art (Coffman, Kanevsky, Kneser and Zadrozny) in Further View of Chai.	132

TABLE OF CONTENTS
(continued)

	Page
1. Claim 5: “A method for processing natural language speech utterances with context-specific domain agents, comprising:”	132
(a) “receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;” (5[a])	132
(b) “recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables;” (5[b]).....	132
(c) “parsing, at a parser coupled to the processing device, information relating to the utterance to determine a meaning associated with the utterance and a context associated with the request contained in the utterance, wherein the parsed information includes the one or more recognized words or phrases;” (5[c])	133
(d) “formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context, wherein formulating the request in accordance with the grammar used by the domain agent includes:” (5[d]).....	133
(i) “determining one or more required values and one or more optional values associated with formulating the request in the grammar used by the domain agent;” (5[d][i])	134

TABLE OF CONTENTS

(continued)

Page

(ii)	“extracting one or more criteria and one or more parameters from one or more keywords contained in the one or more recognized words or phrases, wherein the parser extracts the one or more criteria and the one or more parameters using procedures sensitive to the determined context;” (5[d][ii]).....	143
(iii)	“inferring one or more further criteria and one or more further parameters associated with the request using a dynamic set of prior probabilities or fuzzy possibilities; and” (5[d][iii]).....	145
(iv)	“transforming the one or more extracted criteria, the one or more extracted parameters, the one or more inferred criteria, and the one or more inferred parameters into one or more tokens having a format compatible with the grammar used by the domain agent, wherein the one or more tokens include all the required values and one or more of the optional values associated with formulating the request in the grammar used by the domain agent;” (5[d][iv]).....	148
(e)	“processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance; and” (5[e])	150
(f)	“presenting the generated response to the utterance via the speech unit. ” (5[f])	150
E.	Ground 4: Claims 6 and 9 Are Obvious Over Ground 1 Prior Art (Coffman, Kanevsky, Kneser and Zadrozny) in Further View of Redfern.	150

TABLE OF CONTENTS
(continued)

	Page
1. Claim 6: “A method for processing natural language speech utterances with context-specific domain agents, comprising:”	150
(a) “receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request, wherein the request includes at least one command and at least one question;” (6[a]).....	151
(b) “parsing, at a parser coupled to the processing device, information relating to the utterance to determine a meaning associated with the utterance and a context associated with the request contained in the utterance;” (6[b]).....	158
(c) “formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context;” (6[c])	159
(d) “processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance, wherein processing the formulated request with the domain agent includes:” (6[d]).....	160
(e) “directing the at least one command to one or more local or remote processing devices to execute the at least one command; and” (6[e]).....	160
(f) “querying one or more local or network information sources to resolve information requested in the at least one question, wherein querying the one or more local or network information sources further includes:” (6[f]).....	162

TABLE OF CONTENTS
(continued)

	Page
(g) “submitting multiple asynchronous queries to the one or more local or network information sources, wherein the multiple queries include one or more duplicate queries submitted to distinct information sources;” (6[g]).....	163
(h) “asynchronously receiving multiple results to the multiple asynchronous queries from the one or more local or network information sources; and” (6[h])	166
(i) “scoring a relevance associated with the multiple asynchronously received results using a dynamic set of prior probabilities or fuzzy possibilities to determine one or more best responses to the at least one question, wherein scoring the relevance associated with the multiple asynchronously received results to determine the one or more best responses includes:” (6[i]).....	167
(j) “determining one or more tokens required to formulate a response to the at least one question;” (6[j])	171
(k) “applying one or more scraping criteria to the multiple asynchronously received results to extract one or more values for the asynchronously received results;” (6[k])	174
(l) “evaluating the one or more extracted values using the dynamic set of prior probabilities or fuzzy possibilities to resolve ambiguous, incomplete, or conflicting information associated with the one or more extracted values; and” (6[l]).....	176
(m) “selecting one or more best values for the one or more required tokens from the one or more evaluated values, wherein the one or more best responses include the one or more best values for the one or more required tokens; and” (6[m]).....	178

TABLE OF CONTENTS
(continued)

	Page
(n) “presenting the generated response to the utterance via the speech unit, wherein the response presented via the speech unit includes results associated with the one or more local or remote processing device executing the at least one command and the one or more best responses to the at least one question.” (6[n])	180
2. Claim 9: “The method according to claim 6, wherein scoring the relevance associated with the multiple asynchronously received results to determine the one or more best responses further includes:”	181
(a) “determining that the dynamic set of prior probabilities or fuzzy possibilities failed to resolve the ambiguous, incomplete, or conflicting information associated with the one or more extracted values; and” (9[a])	182
(b) “requesting a user that spoke the utterance to provide additional information relating to the request, wherein the domain agent uses the additional information relating to the request to resolve the ambiguous, incomplete, or conflicting information associated with the one or more extracted values.” (9[b]).....	183
VI. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS	184
VII. CONCLUSION	185

I, Christopher M. Schmandt, declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

A. Qualifications and Experience

1. I retired several years ago after a 40-year career at the Massachusetts Institute of Technology (“MIT”); for most of that time I was employed as a Principal Research Scientist at the Media Laboratory. In that role I also served as faculty for the MIT Media Arts and Sciences academic program. I was a founder of the Media Laboratory, a research lab which now spans two buildings.

2. I received my B.S. degree in Electrical Engineering and Computer Science from MIT in 1978, and my M.S. in Visual Studies (Computer Graphics) also from MIT. I was employed at MIT since 1980, initially at the Architecture Machine Group which was an early computer graphics and interactive systems research lab. In 1985, I helped found the Media Laboratory and continued to work there until retirement. I was director of a research group titled “Living Mobile.” My research spanned distributed communication and collaborative systems, with an emphasis on multi-media and user interfaces, with a strong focus on speech-based systems. I have over 70 published conference and journal papers and one book in the field of speech technology and user interaction.

3. For the first fifteen years of my career, my research emphasized speech

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

recognition and speech user interfaces. I built the first conversational computer system utilizing speech recognition and synthesis (“Put That There”) starting in 1980. Put That There parsed spoken input into a semantic structure representing a command or intent, and could ask questions of the user to evoke responses which would be further parsed and melded into the context of the current command. Some information could be gleaned from non-speech channels, specifically where a user was pointing. I continued to innovate speech user interfaces using recognition, text-to-speech synthesis, and recorded audio in a wide variety of projects. I built one of the first graphical user interfaces for audio editing, employing keyword recognition on voice memos in 1982 (“Intelligent Ear”). I built the first research-grade unified messaging system, which combined text and voice messages into a single inbox, with speech recognition over the phone for remote access, and a graphical user interface for desktop access in 1983 (“Phone Slave”). Again, Phone Slave included a parser which could accept fragmentary input and build up a complete representation of the user’s utterance with additional input. Especially in its follow up multi-user system, Conversational Desktop (1985), a number of contextual cues could also aid the dialog understanding, including date, time, calendar entries, and with whom a user was speaking over a teleconference channel. Along with my students we built the first system for real time spoken driving directions, including

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

speech-accessible maps of Cambridge, Massachusetts in 1987 (“Back Seat Driver”). Back Seat Driver had to translate a route from a map form in which each block was a separate street segment with intersection information contained in nodes, into spoken directions, using words like we use with each other to tell the driver where to go. We built some of the earliest speech-based personal assistants for managing messages, calendar, contacts, etc. (Conversational Desktop 1985, Chatter 1993, MailCall 1996). We built quite a few systems employing speech recognition in handheld mobile devices (ComMotion 1999, Nomadic Radio 2000, Impromptu 2001, and Symphony 2004, for example). We applied speech recognition to large bodies of everyday conversations captured with a wearable device and utilized as a memory aid (Memory Prosthesis 2004). We used speech recognition on radio newscasts to build a personalized version of audio newscasts (Synthetic News Radio, 1999) and also investigated adding speech recognition to a mouse-based window system a few years earlier.

4. In addition to this work running my academic research lab, I helped found and then design speech user interfaces for InTouch Systems, which built natural language based personal digital assistants accessible over the telephone. In this context I helped solve a number of problems in obtaining adequate speech recognition performance in the face of personal information, such as contact lists,

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

containing words which were difficult for a system to understand. The system also maintained dialog context, such that while reading email, for example, commands related to email were anticipated, such as finding important messages or messages from some particular person. When a user was accessing a calendar, a different grammar was invoked, emphasizing creation and retrieval of calendar entries by date or time.

5. I was later awarded the prestigious Association for Computing Machinery (ACM) Computer Human Interface (CHI) Academy membership specifically for those years of work pioneering speech user interfaces.

6. In the course of my research, I built a number of speech recognition client/server distributed systems, with the first being in 1985. Much of the initial motivation for a server architecture was that speech recognition required expensive digital signal processing hardware that we could not afford to put on each computer, so a central server with the required hardware was used. Later versions of the speech recognition server architecture allowed certain computers to perform specialized tasks serving a number of client computers providing voice user interfaces, either on screens or over telephone connections. Because of my early work with distributed speech systems, I served for several years in the mid-1990s with a working group on the impact of multimedia systems on the Internet reporting to the Internet

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

Engineering Task Force (IETF) and later the Internet Activities Board (IAB). This work impacted emerging standards such as Session Initiation Protocol (SIP).

7. In my faculty position I taught graduate level courses in speech technology and user interaction design, and directly supervised student research and theses at the Bachelors, Masters, and PhD level. I oversaw the Masters and PhD thesis programs for the entire Media Arts and Sciences academic program during my more senior years. I also served on the Media Laboratory intellectual property committee for many years.

8. My Curriculum Vitae is attached as **Exhibit A**.

9. I have been retained by counsel for Petitioner to provide my expert opinion in connection with the above-captioned proceeding as set forth herein.

B. Materials Considered

10. The analysis that I provide in this Declaration is based on my education, research, and experience, as well as the documents I have considered. In forming my opinions, I have read and considered U.S. Patent No. 8,015,006 B2 (“’006 patent”) [EX1001] and its prosecution history. I have cited to the following documents in my analysis below:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

Exhibit No.	Description of Document
1001	U.S. Patent No. 8,015,006 B2 to Robert Kennewick et al. (filed May 30, 2008, issued September 6, 2011) (“ 006 ” or “ 006 patent ”)
1003	International Patent Application Publication No. WO 00/20962 to Daniel Coffman, Stephane H. Maes et al. (filed October 1, 1999, published April 13, 2000) (“ Coffman ”) ¹
1004	U.S. Patent No. 5,897,616 to Dimitri Kanevsky and Stephane Herman Maes (filed June 11, 1997, issued April 27, 1999) (“ Kanevsky ”)
1005	U.S. Patent No. 6,157,912 to Reinhard Kneser et al. (filed March 2, 1998, issued December 5, 2000) (“ Kneser ”)
1006	Wlodek Zadrozny et al., “Natural Language Understanding with a Grammar of Constructions”, Proc. 15th Int’l Conf. on Computational Linguistics (Vol. 2) (1994) (“ Zadrozny ”)
1007	International Patent Application Publication No. WO 00/21075 to Stephane H. Maes et al. (filed October 1, 1999, published April 13, 2000) (“ Maes ”) ²
1008	Excerpts from Dafydd Gibbon et al., <i>Handbook of Multimodal and Spoken Dialogue Systems : Resources, Terminology, and Product Evaluation</i> (2000) (“ Gibbon ”)
1009	Excerpts from <i>Microsoft Computer Dictionary</i> (5th ed. 2002)
1010	U.S. Patent No. 6,073,101 to Stephane Herman Maes (filed January 28, 1997, issued June 6, 2000) (“ Maes ’101 ”)
1011	U.S. Patent No. 6,321,196 B1 to Carlos Antonio Franceschi (filed July 2, 1999, issued November 20, 2001) (“ Franceschi ”)

¹ I understand that Coffman also issued as U.S. Patent No. 7,137,126.

² I understand that Maes also issued as U.S. Patent No. 7,003,463.

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

Exhibit No.	Description of Document
1012	U.S. Patent No. 6,829,603 B1 to Joyce Yue Chai, Wlodek Zadrozny et al. (filed February 2, 2000, issued December 7, 2004) (“ Chai ”)
1013	U.S. Patent No. 6,078,914 to Darren M. Redfern (filed December 9, 1996, issued June 20, 2000) (“ Redfern ”)
1014	U.S. Patent Application Publication No. 2003/0144846 to Lawrence A. Denenberg and Christopher M. Schmandt (filed January 31, 2002, published July 31, 2003) (“ Denenberg ”)
1015	Excerpts from <i>Merriam-Webster’s Collegiate Dictionary</i> (10th ed. 1994)
1016	Excerpts from Richard F. E. Sutcliffe et al., <i>Industrial Parsing of Software Manuals</i> (1996) (“ Sutcliffe ”)
1017	Christopher Schmandt, “The Intelligent Ear: A Graphical Interface to Digital Audio” in Proc. Int’l Conf. Cybernetics and Society, pp.393-397 (1981)
1018	Christopher Schmandt et al., “The Intelligent Voice-Interactive Interface,” Proc. Human Factors in Comp. Sys., pp.363-366 (1982)
1019	Christopher Schmandt et al., “A Robust Parser and Dialog Generator for a Conversational Office System” in Proc. AVIOS Conf. (1986)

II. PERSON OF ORDINARY SKILL IN THE ART

11. I understand that, under the patent laws in effect before the America Invents Act (“AIA”) of 2011, an assessment of claims of a patent filed before the AIA took effect should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date (i.e., the “time the invention

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

was made”). I have preliminarily for purposes of my Declaration only assumed that date to be June 3, 2002, the filing date of the earliest patent application to which the ’006 patent claims priority (provisional appl. ser. no. 60/384,388).³

12. I have also been advised that to determine the appropriate level of a person having ordinary skill in the art, the following factors may be considered: (1) the types of problems encountered by those working in the field and prior art solutions thereto; (2) the sophistication of the technology in question, and the rapidity with which innovations occur in the field; (3) the educational level of active workers in the field; and (4) the educational level of the inventor.

13. The ’006 patent states that it “relates to the retrieval of online information or processing of commands through a speech interface.” (’006, 1:20-22.) In my opinion, a person of ordinary skill in the art would have possessed a bachelor’s degree in electrical engineering or computer science, and two years of work experience in the field of speech- and natural language-interfaces for interacting with and directing computer systems and software, including speech

³ I express no opinion on whether the ’006 patent is actually entitled to the benefit of the provisional application to which it claims priority (60/384,388). This issue is not relevant here because all of the prior art references relied upon predate that date. In the event entitlement to the provisional filing date may later become relevant, for example based on arguments later presented by Patent Owner, I reserve my right to provide opinions on that issue at that time.

recognition and processing. A person could also have qualified as a person of ordinary skill in the art with some combination of (1) more formal education (such as a master's of science degree) and less technical experience, or (2) less formal education and more technical or professional experience.

14. My opinions regarding the level of ordinary skill in the art are based on, among other things, my over 40 years of experience in the field of computer science, my understanding of the basic qualifications that would be relevant to an engineer or scientist tasked with investigating methods and systems in the relevant area, and my familiarity with the backgrounds of colleagues, co-workers, and employees, both past and present.

15. Although my qualifications and experience exceed those of the hypothetical person having ordinary skill in the art defined above, my analysis and opinions regarding the '006 patent have been based on the perspective of a person of ordinary skill in the art as of June 2002.

III. STATEMENT OF LEGAL PRINCIPLES

A. Claim Construction

16. I understand that a purpose of claim construction is to determine what a person of ordinary skill in the art would have understood the claim terms to mean. Claim terms are generally given their ordinary and customary meaning, which is the

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

meaning that the term would have to a person of ordinary skill in the art in question as of the effective filing date.

17. 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 disputed term appears, but in the context of the entire patent, including the specification. I understand that the patent specification, under the legal principles, has been described as the single best guide to the meaning of a claim term, and is thus highly relevant to the interpretation of claim terms. And I understand for terms that do not have a customary meaning within the art, the specification usually supplies the best context of understanding the meaning of those terms.

18. I further understand that other claims of the patent in question, both asserted and unasserted, can be valuable sources of information as to the meaning of a claim term. Because the claim terms are normally used consistently throughout the patent, the usage of a term in one claim can often illuminate the meaning of the same term in other claims. Differences among claims can also be a useful guide in understanding the meaning of particular claim terms.

19. I understand that the prosecution history can further inform the meaning of the claim language by demonstrating how the inventors understood the invention and whether the inventors limited the invention in the course of prosecution, making

the claim scope narrower than it otherwise would be. Extrinsic evidence, such as dictionaries, may also be consulted in construing the claim terms.

20. I understand that, in *Inter Partes* Review (IPR) proceedings, a claim of a patent shall be construed using the same claim construction standard that would be used to construe the claim in a civil action filed in a U.S. district court (which I understand is called the “*Phillips*” claim construction standard), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.

21. I have been instructed by counsel to apply the “*Phillips*” claim construction standard for purposes of interpreting the claims in this proceeding, to the extent they require an explicit construction. The description of the legal principles set forth above thus provides my understanding of the “*Phillips*” standard as provided to me by counsel.

B. Obviousness (§ 103)

22. I understand that a patent claim is obvious if, as of the effective filing date, it would have been obvious to a person having ordinary skill in the field of the technology (the “art”) to which the claimed subject matter belongs.

23. I understand that the following factors should be considered in

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

analyzing obviousness: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims; and (3) the level of ordinary skill in the pertinent art. I also understand that certain other facts known as “secondary considerations” such as commercial success, unexplained results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others may be utilized as indicia of nonobviousness. I understand, however, that secondary considerations should be connected, or have a “nexus,” with the invention claimed in the patent at issue.

24. I understand that a reference qualifies as prior art for obviousness purposes when it is analogous to the claimed invention. The test for determining what art is analogous is: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the inventor’s endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved.

25. I understand that a person of ordinary skill in the art is assumed to have knowledge of all prior art. I understand that one skilled in the art can combine various prior art references based on the teachings of those prior art references, the general knowledge present in the art, or common sense. I understand that a motivation to combine references may be implicit in the prior art, and there is no

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

requirement that there be an actual or explicit teaching to combine two references. Thus, one may take into account the inferences and creative steps that a person of ordinary skill in the art would employ to combine the known elements in the prior art in the manner claimed by the patent at issue. I understand that one should avoid “hindsight bias” and ex post reasoning in performing an obviousness analysis. But this does not mean that a person of ordinary skill in the art for purposes of the obviousness inquiry does not have recourse to common sense.

26. I understand that when determining whether a patent claim is obvious in light of the prior art, neither the particular motivation for the patent nor the stated purpose of the patentee is controlling. The primary inquiry has to do with the objective reach of the claims, and that if those claims extend to something that is obvious, then the entire patent claim is invalid.

27. I understand one way that a patent can be found obvious is if there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent’s claims. I understand that a motivation to combine various prior art references to solve a particular problem may come from a variety of sources, including market demand or scientific literature. I understand that a need or problem known in the field at the time of the invention can also provide a reason to combine prior art references and render a patent claim invalid for

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

obviousness. I understand that familiar items may have obvious uses beyond their primary purpose, and that a person of ordinary skill in the art will be able to fit the teachings of multiple prior art references together like the pieces of a puzzle. I understand that a person of ordinary skill is also a person of at least ordinary creativity. I understand when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this finite number of predictable solutions leads to the anticipated success, I understand that the invention is likely the product of ordinary skill and common sense, and not of any sort of innovation. I understand that the fact that a combination was obvious to try might also show that it was obvious, and hence invalid, under the patent laws. I understand that if a patent claims a combination of familiar elements according to known methods, the combination is likely to be obvious when it does not more than yield predictable results. Thus, if a person of ordinary skill in the art can implement a predictable variation, an invention is likely obvious. I understand that combining embodiments disclosed near each other in a prior art reference would not ordinarily require a leap of inventiveness.

28. I understand that obviousness may be shown by demonstrating that it would have been obvious to modify what is taught in a single piece of prior art to

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

create the patented invention. Obviousness may also be shown by demonstrating that it would have been obvious to combine the teachings of more than one item of prior art. I understand that a claimed invention may be obvious if some teaching, suggestion, or motivation exists that would have led a person of ordinary skill in the art to combine the invalidating references. I also understand that this suggestion or motivation may come from the knowledge of a person having ordinary skill in the art, or from sources such as explicit statements in the prior art. I understand that when there is a design need or market pressure, and there are a finite number of predictable solutions, a person of ordinary skill may be motivated to apply common sense and his skill to combine the known options in order to solve the problem.

29. I understand the following are examples of approaches and rationales that may be considered in determining whether a piece of prior art could have been combined with other prior art or with other information within the knowledge of a person having ordinary skill in the art:

- (1) Some teaching, motivation, or suggestion in the prior art that would have led a person of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- (2) Known work in one field of endeavor may prompt variations of it for use in the same field or a different field based on design incentives or other

market forces if the variations would have been predictable to a person of ordinary skill in the art;

(3) Combining prior art elements according to known methods to yield predictable results;

(4) Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;

(5) Applying a technique or approach that would have been “obvious to try” (choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success);

(6) Simple substitution of one known element for another to obtain predictable results; or

(7) Use of a known technique to improve similar products, devices, or methods in the same way.

30. I understand that, when determining whether a claimed combination is obvious, the correct analysis is not whether one of ordinary skill in the art, writing on a blank slate, would have chosen the particular combination of elements described in the claim. Instead, I understand the correct analysis considers whether one of ordinary skill, facing the wide range of needs created by developments in the field of endeavor, would have seen a benefit to selecting the combination claimed.

31. I understand that the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference. The test for obviousness, in other words, is not whether the references could be physically combined but whether the claimed inventions are rendered obvious by the teachings of the prior art as a whole.

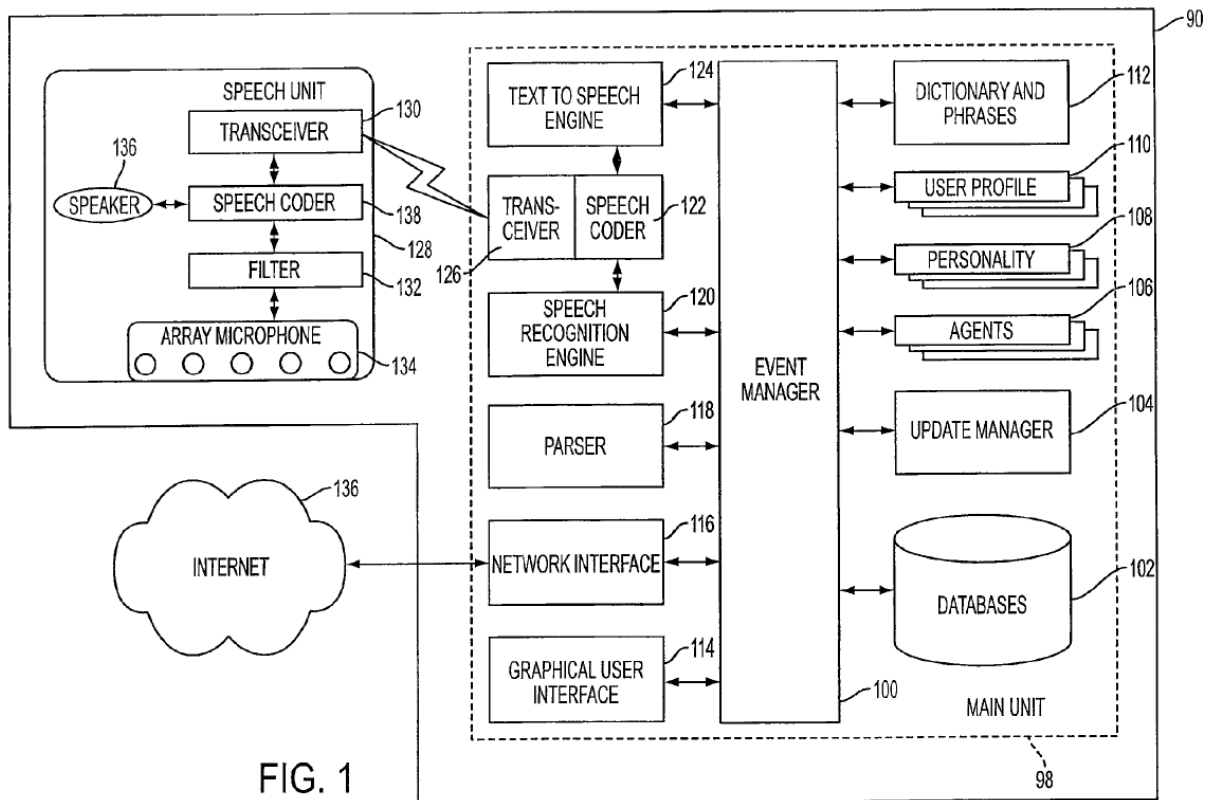
IV. THE '006 PATENT

A. Overview of the Specification

32. The '006 patent states that “[t]he present invention relates to the retrieval of online information or processing of commands through a speech interface.” ('006, 1:20-22.)

33. The '006 patent admits that “[m]uch work covering multiple methods has been done in the fields of natural language processing and speech recognition,” that “[s]peech recognition has steadily improved in accuracy and today is successfully used in a wide range of applications,” and that “[n]atural language processing has previously been applied to the parsing of speech queries.” ('006, 1:45-50.) According to the '006 patent, however, “current systems are generally unable to provide a complete environment for users to make natural language speech queries and receive natural sounding responses.” ('006, 1:50-53.) The '006 patent purports to “overcome...drawbacks of prior systems.” ('006, 1:59-61.)

34. Figure 1 of the '006 patent, reproduced below, "is an overall diagrammatic view according to one embodiment of the invention" ('006, 8:54-55):



('006, Fig. 1.) As shown in the figure above, system **90** includes, among other things, speech unit **128**, speech recognition engine **120**, dictionary and phrases **112**, parser **118**, and agents **106**. Speech unit **128** includes a microphone to receive a spoken utterance from a user. ('006, 10:30-32.) After initial processing, the speech is passed to speech recognition engine **120** for processing "using data in the dictionary and phrases module **112** and received from the agents **106**." ('006, 10:56-60.) Then, "[t]he recognized words and phrases are processed by the parser **118**,

which transforms them into complete commands and questions using data supplied by an agent[s] **106**.” (’006, 10:61-63.) “The agent[s] **106** then process the commands or questions.” (’006, 10:63-64.) In order to select the correct agent to process the command or question, “parser **118** determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history.” (’006, 17:49-51.) “For example, a question with the keywords ‘temperature’ implies a context value of weather for the question.” (’006, 17:53-55.) “Once the context for the question or command has been determined, the parser **118** can invoke the correct agent **156**, **150**.” (’006, 18:12-14.) The agent returns results to the user. (’006, 11:5-6.)

35. With respect to “agents,” the ’006 patent states:

Agents are autonomous executables that receive, process and respond to user questions, queries and commands. The agents provide complete, convenient and re-distributable packages or modules of functionality, typically for a specific domain or application. Agents can be complete packages of executable code, scripts, links to information, and other forms of communication data, required to provide a specific package of functionality, usually in a specific domain. In other words, an agent includes everything that is needed to extend the functionality of the invention to a new domain. Further, agents and their associated data can be updated remotely over a network as new behavior is added or new information becomes available. Agents can use system resources and

the services of other, typically more specialized, agents. Agents can be distributed and redistributed in a number of ways including on removable storage media, transfer over networks or attached to emails and other messages.

(’006, 2:56-3:5.)

36. I discuss additional aspects of the ’006 patent as needed in the analysis below.

B. The Challenged Claims

37. This Declaration addresses claims 1-6 and 9 of the ’006 patent. Independent claim 1 recites:

1. A method for processing natural language speech utterances with context-specific domain agents, comprising:
 - [a] receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;
 - [b] recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables, wherein recognizing the one or more words or phrases contained in the utterance includes:
 - [b][i] dynamically updating the information in the one or more dictionary and phrase tables based on a dynamic set of prior probabilities or fuzzy possibilities;

- [b][ii] determining an identity associated with a user that spoke the utterance based on voice characteristics associated with the utterance; and
- [b][iii] associating the one or more recognized words or phrases and a pronunciation associated with the one or more recognized words or phrases with the determined identity and the request contained in the utterance in response to the one or more recognized words or phrases satisfying a predetermined confidence level;
- [c] parsing, at a parser coupled to the processing device, the one or more recognized words or phrases to determine a meaning associated with the utterance and a context associated with the request contained in the utterance, wherein the one or more recognized words or phrases are further associated with the determined context in response to the one or more recognized words or phrases satisfying the predetermined confidence level;
- [d] formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context;
- [e] processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance; and
- [f] presenting the generated response to the utterance via the speech unit.

(’006, 25:49-26:21 (Claim 1) (bracketed notation (e.g., “[a]”) added).)

38. I address the claims further in my detailed analysis in **Part V** below.

V. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS

39. I have reviewed and analyzed the prior art references and materials listed in **Part I.B** above. In my opinion the claims of the '006 patent would have been obvious to a person of ordinary skill in the art based on the following combinations of the prior art.

Ground	Claims	Basis for Challenge under § 103
1	1-3	Coffman (EX1003), Kanevsky (EX1004), Kneser (EX1005), and Zadrozny (EX1006)
2	4	Ground 1 Prior Art + Maes '101 (EX1010) and Franceschi (EX1011)
3	5	Ground 1 Prior Art + Chai (EX1012)
4	6, 9	Ground 1 Prior Art + Redfern (EX1013)

40. I am informed by counsel that each of the references cited in the grounds above qualifies as prior art to the challenged claims because each reference was filed and/or published before the '006 patent priority date (June 3, 2002).

A. Brief Summary of Prior Art

1. Prior Art Common to All Grounds

(a) Coffman (**EX1003**)

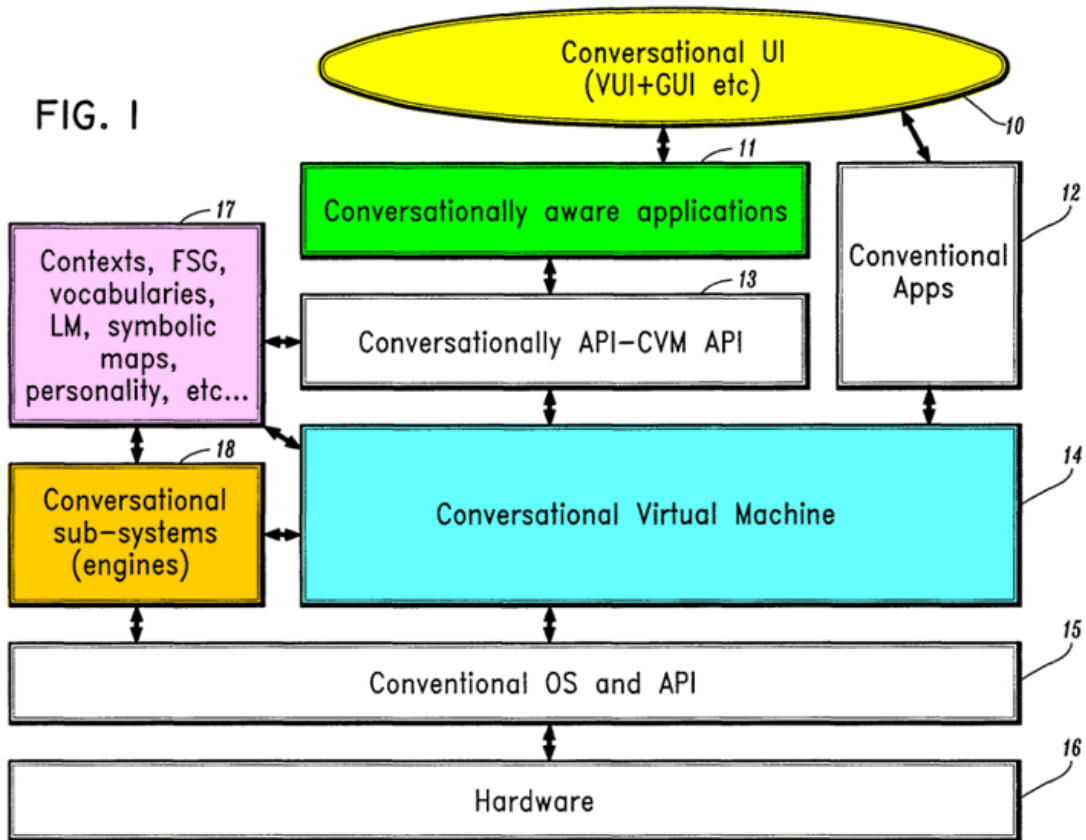
41. **Coffman**, International Patent Application Publication No. WO 00/20962 to Daniel Coffman et al. is entitled "Conversational Computing via Conversational Virtual Machine." The face page of Coffman identifies International

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

Business Machines as the applicant. (Coffman, face page (field 71).) I am informed that Coffman qualifies as prior art because it published on April 13, 2000, before the earliest priority date of the '006 patent.

42. Coffman is a strong reference that discloses or suggests the majority of the limitations of the challenged claims. And as explained below, it also incorporates by reference Kanevsky [EX1004] and Maes [EX1007] and shares links to Zadrozny [EX1006], Maes '101 [EX1010], Franceschi [EX1011] and Chai [EX1012], such as common inventors and/or assignment (to IBM).

43. Coffman describes a “conversational computing system.” Generally speaking, a conversational system is a computing system with either purely voice input/output or a multi-modal system with voice input/output (as well as other input/output means). (Maes, 1:12-13.) Coffman’s system “provides the capability to have a natural dialog with NLU [natural language understanding], NLG [natural language generation] and mixed initiative across multiple applications, with multiple domains.” (Coffman, 36:29-31.) Figure 1 of Coffman provides a high-level depiction of the system’s components:



(Coffman, Fig. 1 (highlighting added).) As shown above, the system includes, among other things, “a universal coordinated multi-modal conversational user interface (CUI) (10),” “a plurality of conversationally-aware applications (11),” “conversational kernel (14),” “conversational arguments (17),” and “conversational engines (18).” (Coffman, face page (Abstract).)

44. The conversational virtual machine (or CVM) is “[a] key component” of the conversational system that “allows a computer or any other interactive device to converse with a user.” (Coffman, 10:23-25, 11:10-13.) To do this, the CVM uses conversational subsystems “including speech recognition, speaker recognition, text-

to-speech, natural language understanding and natural dialog generation engines to understand and generate dialog between user and machine.” (Coffman, 5:1-4.) The **conversational subsystems** “are responsible for converting voice requests into queries and converting outputs and results into spoken messages using the appropriate **data files 17** (e.g., contexts, finite state grammars, vocabularies, language models, symbolic query maps etc.)” (Coffman, 13:8-11.)

45. “The **conversationally aware applications 11** are applications that are specifically programmed for operating with a **CVM core layer (or kernel) 14** via conversational application APIs 13.” (Coffman, 12:28-30.) A **conversationally aware application** operates with a **CVM** by providing its **data files** to the CVM for use by the **conversational subsystems**—an application hooks into the CVM API by “registering all its conversational capabilities and requirements” including “[a]ctive vocabulary, finite state grammar and language models to control the application” and “[s]ymbolic mapping if NLU is supported or at list [sic; least] a context state list.” (Coffman, 57:23-27; *see also id.*, 36:31-32 (“each application will provide the CVM 401 its own parsing and translation arguments”).)

46. The CVM also “allows the user to run multiple tasks on a machine.” (Coffman, 11:10-13.) The CVM is able to do this by using a “context stack,” which enables the system to determine the appropriate conversationally-aware application

for handling user input. (Coffman, 42:16-17 (“This stack structure allows to maintain non-ambiguous conversational interactions with multiple tasks, threads or processes.”).) The context stack “comprises all the information associated with an application.” (Coffman, 21:23-24.) “Such information includes all the variable, states, input, output and queries to the backend that are performed in the context of the dialog and any extraneous event that occurs during the dialog.” (Coffman, 21:24-26.) Coffman explains:

In general, the context stack 405 accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 (or at least identifiers of such conversational arguments) for the different engines that may be needed for input understanding (e.g., files or arguments that the engines use for performing their respective tasks such as active FSG, topic, vocabulary, HMM (hidden markov models), voiceprints, language models or possible queries for a speech input). In other words, the term “context” refers to the state of each discourse (whether active or nonnative), which keeps track of the past history of the discourse, its current state, and the specific characteristics and full query arguments of the corresponding task (e.g, vocabulary file, language model, parsing, tags, voiceprint, TTS rules, grammar, NLU etc. of each active task/process) along with any data needed for input understanding.

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

(Coffman, 29:20-32.) To identify the appropriate conversationally-aware application to handle a user's input (such as a user's spoken utterance), the system searches the context stack for a context that is an acceptable match:

Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active.

(Coffman, 42:1-5.) "The first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other application is stored to activate if the recognized query is rejected by the user."

(Coffman, 37:2-4; *see also id.*, 42:18-19 ("If the request is complete, it will be executed, pending possible request for confirmation by the user, e.g. when it is irreversible.").)

47. I discuss additional aspects of Coffman as needed in my analysis of the challenged claims below.

(b) Kanevsky (EX1004)

48. **Kanevsky**, U.S. Patent No. 5,897,616 to Dimitri Kanevsky and Stephane Herman Maes is entitled "Apparatus and Methods for Speaker

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

Verification/Identification/Classification Employing Non-Acoustic and/or Acoustic Models and Databases.” I am informed that Kanevsky qualifies as prior art because it was filed on June 11, 1997 and issued on April 27, 1999, before the earliest priority date of the ’006 patent.

49. I note that Coffman expressly incorporates Kanevsky by reference. (Coffman, 37:10-13.) I am informed by counsel that under the patent laws, when a document is “incorporated by reference” into a host document, the incorporated document’s material effectively becomes part of the host document as if it were explicitly contained therein. Therefore, I am informed that Kanevsky is properly considered part of the disclosures of Coffman for purposes of applying it as prior art. I further note that both references also share a named co-inventor, Stephane Herman Maes, and identify IBM as the assignee/applicant. (Coffman, face page; Kanevsky, face page.)

50. My analysis points to Kanevsky primarily for Kanevsky’s further details about speech recognition. Claim 1**[b][i]** of the ’006 patent recites **“dictionary and phrase tables”** and a **“set of prior probabilities.”** As mentioned above, Coffman describes a conversational computing system that uses speech recognition and also describes how a conversationally-aware application provides data files for controlling the application, including active vocabulary, finite state

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

grammar and language models. Coffman describes using the language models as part of “converting voice requests into queries” (Coffman, 13:8-11) but does not appear to further explain what a language model is or how it is used for speech recognition. That is not surprising because language models and their use for speech recognition were well known. This is confirmed by Kanevsky, which explains that a speech recognition engine may be implemented using “Hidden Markov Models, mixtures of Gaussian probabilities, mel cepstral vectors as acoustic features, a 20K or 64K vocabulary and a trigram language model.” (Kanevsky, 13:50-56.) Kanevsky describes such a speech recognition engine as an “existing component” and explains that “[s]uch systems, for example, are disclosed in IBM’s speech dictation engine ‘Simply Speaking’ and in ‘Transcription of Radio Broadcast News with the IBM Large Vocabulary Speech Recognition System’, P.S. Gopalakrishnan [sic, Gopalakrishnan] et al., Proceeding Speech Recognition Workshop, Arden House, Drapa (1996).” (Kanevsky, 13:48-61.)

51. As I discuss in my analysis below, a “trigram language model” was a well-known type of “n-gram language model” comprising sequences of three words (i.e., “trigrams”) and the probabilities of their occurrence based (at least initially) on some previously collected representative set of text, called training data. As I explain below, the probabilities associated with the trigrams in a language model are

“prior probabilities.”

52. I discuss additional aspects of Kanevsky as needed in my analysis of the challenged claims below.

(c) Kneser (EX1005)

53. **Kneser**, U.S. Patent No. 6,157,912 to Reinhard Kneser et al. is entitled “Speech Recognition Method with Language Model Adaptation.” I am informed that Kneser qualifies as prior art because it was filed on March 2, 1998 and issued on December 5, 2000, before the earliest priority date of the ’006 patent.

54. My analysis points to Kneser for its teachings regarding adaptive language models. Claim 1 [b][i] of the ’006 patent recites a “**dynamic set of prior probabilities**.” As I noted above, a language model, such as a trigram language model disclosed by Kanevsky, comprises a set of prior probabilities. An “adaptive” language model is a language model whose probability information can change as the model is used, such that the “prior probabilities” “adapt” to reflect that usage history. Although neither Coffman nor Kanevsky appears to specifically mention using them, adaptive n-gram language models and their advantages were well known, which Kneser confirms, explaining:

These language model values represent the probabilities that given word sequences of limited length are present; they have been obtained from a comparatively large quantity of training texts, which may also

be formed by written texts. Long-term changes in the use of language, for example subject matter or style, can accordingly not be taken into account with these language model values [probabilities]. Since the quality of the adaptation of the language model values has a considerable influence on the reliability of the speech recognition, it is desirable to adapt the language model values as well as possible to the given text by means of a small quantity of updated text. This adaptation may take place during the process of recognizing a given text or alternatively subsequent to the recognition of part of a text, after recognition errors have been manually corrected.

(Kneser, 1:22-29; *see generally id.*, 1:13-37.)

55. As explained below, based on Kneser's teachings, it would have been obvious for the language models described by the combination of Coffman and Kanevsky to be "adaptive," thereby disclosing a "**dynamic set of prior probabilities.**"

56. I discuss additional aspects of Kneser as needed in my analysis of the challenged claims below.

(d) Zadrozny (EX1006)

57. **Zadrozny** is a paper authored by Wlodek Zadrozny et al. entitled "NL Understanding with a Grammar of Constructions" and published in volume 2 of the Proceedings of the 15th International Conference on Computational Linguistics in

1994. I am informed that Zadrozny qualifies as prior art because its 1994 publication date is before the earliest priority date of the '006 patent.

58. In another link between prior art references, Zadrozny is cited in Kanevsky as an example of a semantic analyzer. (Kanevsky, 14:22-25.) Zadrozny also identifies a number of its co-authors (including Wlodek Zadrozny) as being associated with IBM Research Division. (Zadrozny, p.1.)

59. My analysis points to Zadrozny primarily for its teachings regarding using a grammar associated with an application to perform natural language understanding and formulate an application query. Claim 1[d] of the '006 patent recites “**formulating**, at the parser, **the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context.**” Zadrozny describes a “grammar of constructions” associated with an application (such as a calendar application) for performing natural language understanding to translate a recognized utterance into “a form that can be executed by the application”:

We present an approach to natural language understanding based on a computable grammar of constructions. A construction consists of a set of features of form and a description of meaning in a context. A grammar is a set of constructions. This kind of grammar is the key element of MINCAL, an implemented natural language speech-enabled

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

interface to an on-line calendar system. . . .

With respect to parsing, grammars of constructions can be parsed like “standard” grammars, except that the set of features is richer. Given a string (representing a sentence, a fragment of a discourse or a paragraph), the parser assigns it a construction.... Here the message of an utterance must be understood in the context of an intended action. This is done in two steps. First, the system determines the intended action and its parameters, using domain knowledge (meetings+time+places). Second, once all the parameters have been extracted from the dialog, the system executes the action. To do this, the program uses application-specific knowledge to translate the action and its parameters into a form that can be executed by the application (Xdiary).

(Zadrozny, pp.1-2.) As explained above in the excerpt above, Zadrozny describes a technique using a “grammar of constructions” to formulate an utterance as an intended action and parameters and then translate the action and parameters into a form for the application to execute. Zadrozny illustrates the technique with the example utterance “I want you to arrange a conference in my office at 5.” (Zadrozny, p.3.) Using a grammar of constructions, the utterance is “analyzed as *sent(assert, svoc)*, an assertive sentence consisting of a subject, verb, an object and a complement” and formulated as an action (“schedule”) and series of parameters:

```
[    [    action_name schedule]
      [    event_name
```

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

```
[    a conference]
[  event_time
  [    [    minute 0]
    [    hour
      [    5 am_or_pm]
  event_place
  my office]
```

(Zadrozny, pp.3-4.) As Zadrozny explains, this action and parameters are then translated into a form for the application to execute (i.e., an application call/query).

60. As explained below, it would have been obvious to incorporate Zadrozny's teachings regarding a grammar of constructions into Coffman's conversational system in order to translate a user's recognized spoken utterance into an application-specific query.

61. I discuss additional aspects of Zadrozny as needed in my analysis of the challenged claims below.

2. Additional Prior Art for Certain Dependent Claims

(a) Maes '101 (EX1010)

62. **Maes '101**, U.S. Patent No. 6,073,101 to Stephane Herman Maes is entitled "Text Independent Speaker Recognition for Transparent Command Ambiguity Resolution and Continuous Access Control." I am informed that Maes '101 qualifies as prior art because it was filed on January 28, 1997 and issued on June 6, 2000, both dates before the earliest priority date of the '006 patent.

63. I note that Coffman and Maes '101 name a common co-inventor, Stephane Herman Maes of Danbury, Connecticut, and the same assignee/applicant, IBM. (Maes '101, face page; Coffman, face page.)

64. My analysis points to Maes '101 primarily for its teachings regarding prompting a user to spell a word when its recognition fails to meet a confidence level. Claim 4[b][i] recites “**requesting a user that spoke the utterance to spell the one or more words or phrases in response to the one or more recognized words or phrases failing to satisfy a predetermined confidence level.**” Maes '101 teaches that “unrecognized utterances can be defined by the user during enrollment or later enrollment correction by prompting the user to spell the word whenever speech recognition is unsuccessful.” (Maes '101, 6:52-55.) Maes '101 discloses that, by spelling words, the system's performance can be improved and adapted to the user, by adding the spelled word to the speech recognizer's vocabulary, possibly associated only with particular users. (Maes '101, 6:55-66.)

65. I discuss additional aspects of Maes '101 as needed in my analysis of the challenged claims below.

(b) Franceschi (EX1011)

66. **Franceschi**, U.S. Patent No. 6,321,196 B1 to Carlos Antonio Franceschi is entitled “Phonetic Spelling for Speech Recognition.” I am informed

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

that Franceschi qualifies as prior art because it was filed on July 2, 1999 and issued on November 20, 2001, both dates before the earliest priority date of the '006 patent.

67. Franceschi, like many of the other prior art references I rely on in my analysis, identifies IBM as the assignee. (Franceschi, face page.)

68. My analysis points to Franceschi for its teachings regarding a user providing a phonetic alphabet spelling to a speech recognition system. While claim 4[b][i] of the '006 patent simply recites “**requesting a user...to spell,**” 4[b][ii] refers to “**a phonetic alphabet spelling**” provided by the user. Franceschi explains the motivations and benefits of phonetic alphabet spelling, and further, that such functionality was routine and conventional, noting that commercially-available speech recognition systems already included it:

[I]t frequently occurs that the best approach for an unrecognized, difficult, or new word is for the speaker to spell it....

However, many of the letters sound very similar and may be confused for each other, even for human listeners. It is therefore known in the art to spell a word phonetically which is meant to indicate using a commonly understood word for each letter in the word to be spelled. For example, one may phonetically spell the work [sic; word] “key” by saying “kilo echo yankee.” One may also use a phonetic alphabet when selecting items from a list by using a word from the phonetic alphabet rather than saying the corresponding letter name.... Various such alphabets have been developed and used over the years with human

listeners, primarily by the military, for clearly communicating over sometimes noisy or unreliable radio or telephone links. The developers of speech recognition devices and programs have likewise incorporated a phonetic spelling feature in their products for word spelling.

(Franceschi, 1:38-52, 2:8-13.) Franceschi purports to describe improved techniques for enabling a user to provide phonetic alphabet spelling in a speech recognition systems. (Franceschi, e.g., 2:46-50.) As I explain below, it would have been obvious for the user to provide a “**phonetic alphabet spelling**” of one or more words and phrases in view of Franceschi.

69. I discuss additional aspects of Franceschi as needed in my analysis of the challenged claims below.

(c) Chai (EX1012)

70. **Chai**, U.S. Patent No. 6,829,603 B1 to Joyce Yue Chai et al. is entitled “System, Method and Program Product for Interactive Natural Dialog.” I am informed that Chai qualifies as prior art because it was filed on February 2, 2000, before the earliest priority date of the ’006 patent, and issued on December 7, 2004.

71. In yet another link between prior art references, Chai lists Wlodek Zadrozny, lead author of Zadrozny [EX1006], as a co-inventor and identifies IBM as the assignee. (Chai, face page.)

72. My analysis points to Chai for its teachings regarding required and

optional values in an application request. Claim 5[d][i] recites “**determining one or more required values and one or more optional values associated with formulating the request in the grammar used by the domain agent.**” As noted above, Zadrozny discloses a grammar (“grammar of constructions”) for formulating an application request (such as a request for a calendar application to schedule a conference). Zadrozny, however, does not appear to expressly disclose that an application request would have “**one or more required values and one or more optional values.**” But Chai similarly describes filling “slots” in “action templates” based on natural language input in order to determine the application action to execute. Chai explains:

One or more presentation manager modules present information from the computer system to one or more users over one or more network interfaces and accept queries from one or more of the users using one or more known input/output modalities (e.g. Speech, typed in text, pointing devices, etc.). A natural language parser parses one or more natural language phrases received over one or more of the network interfaces by one or more of the presentation managers into one or more logical forms (parsed user input), each logical form having a grammatical and structural organization. A dialog manager module maintains and directs interactive sessions between each of the users and the computer system. The dialog manager receives logical forms from one or more of the presentation managers and sends these to a

taxonomical mapping process which matches the items of interest to the user against the content organization in the content database to match business categories and sends modified logical forms back to the dialog manager. In a preferred embodiment, these modified logical forms are matched against a set of predefined action templates, the best matching template is selected, corresponding action requests are sent to an action manager, action results are received from the action manager, and presentation requests are sent to one or more of the presentation managers for presenting the system response to the user. The action manager module receives action requests from the dialog manager, executes the action (e.g. Retrieving the latest price of a stock or all blue pants with cuffs), and sends the action results to the dialog manager.

(Chai, 4:51-5:11.)

73. For example, in an online shopping example, Chai describes processing two user requests, “I want to buy one 30 inch wide White Color Cooktop with Electric PowerSource and with Electric:Radiant Glass CookTop Surface and of KenMore company” followed by “I want to buy one Black CookTop of previous type.” (Chai, 12:28-32.) Chai explains that the user’s input is used to fill various slots of matched action templates, with some slots being “**required**”—for example, a value for the “item name” slot is required:

Before taking discourse context into consideration, the following slots 807 in matching templates 806 with “buy” transaction are filled.

First Template (Used for Pants and Shirts):

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

TRANSACTION: buy

TEMPLATE SLOTS:

Item name: CookTop

sku:

quantity: 1

size:

color: Black

company:

REQUIRED SLOTS: item name, quantity, company

(Chai, 12:32-44; *see also id.*, 12:45-56 (second template), 12:21-22 (“choosing the template with the most number of required slots filled”).) On the other hand, referring to the example action template above, because a value for the “color” slot is not identified as being required, a person of ordinary skill in the art would have understood it to be optional. Chai therefore discloses both “**one or more required values**” (e.g., the value for the “item name” slot in Chai’s shopping example) and “**one or more optional values**” (e.g., the value for the “color” slot in Chai’s example). I implemented these techniques that Chai describes in my own work, such as my Put That There system, starting in 1979. (*See, e.g.*, C. Schmandt et al., “The Intelligent Voice-Interactive Interface” (1982), pp.364-365 (describing matching input words against command templates); C. Schmandt et al., “A Robust Parser and Dialog Generator for a Conversational Office System” (1986) [EX1019], pp.003 (section 3 describing “slot-and-filler approach”), 007 (“although there are slots for both name and time, for example, some commands may require only one or the other

of them”).)

74. As I explain below, it would have been obvious in view of Chai to “**determin[e] one or more required values and one or more optional values**” for a request in the grammar used a conversationally-aware application.

75. I discuss additional aspects of Chai as needed in my analysis of the challenged claims below.

(d) Redfern (EX1013)

76. **Redfern**, U.S. Patent No. 6,078,914 to Darren M. Redfern is entitled “Natural Language Meta-Search System and Method.” I am informed that Redfern qualifies as prior art because it was filed on December 9, 1996 and issued on June 20, 2000, both dates before the earliest priority date of the ’006 patent.

77. My analysis of claims 6 and 9 points to Redfern for its teachings regarding a meta-search application. These claims recite, among other things, that a user’s request “**includes at least one command and at least one question**” and that the domain agent processes the formulated request by “**directing the at least one command to one or more local or remote processing devices**” and “**querying one or more local or network information sources to resolve information requested in the at least one question.**” (See ’006, claim 6.)

78. Redfern describes a meta-search application where a user can provide

a single natural language query to the application, and the application will formulate queries to multiple search engines, such as the well-known Lycos, AltaVista and Excite web-based search engines. (Redfern, e.g., 2:52-3:9, 4:8-34, 8:66-9:12, 15:43-65.) The application then scores the results returned by the search engines and extracts relevant information to present to the user. (Redfern, e.g., 2:52-3:9, 15:19-41, 16:47-60.)

79. As I explain in my analysis below, it would have been obvious for one of Coffman's conversationally-aware applications to take the form of a meta-search application as described by Redfern, where the user could instruct the application to query certain search engines/information sources to answer a question.

80. I discuss additional aspects of Redfern as needed in my analysis of the challenged claims below.

B. Ground 1: Claims 1-3 Are Obvious Over Coffman in View of Kanevsky, Kneser and Zadrozny.

1. Claim 1

81. For reference, claim 1 recites:

1. A method for processing natural language speech utterances with context-specific domain agents, comprising:
 - [a] receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;

- [b] recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables, wherein recognizing the one or more words or phrases contained in the utterance includes:
- [c] dynamically updating the information in the one or more dictionary and phrase tables based on a dynamic set of prior probabilities or fuzzy possibilities;
- [d] determining an identity associated with a user that spoke the utterance based on voice characteristics associated with the utterance; and
- [e] associating the one or more recognized words or phrases and a pronunciation associated with the one or more recognized words or phrases with the determined identity and the request contained in the utterance in response to the one or more recognized words or phrases satisfying a predetermined confidence level;
- [f] parsing, at a parser coupled to the processing device, the one or more recognized words or phrases to determine a meaning associated with the utterance and a context associated with the request contained in the utterance, wherein the one or more recognized words or phrases are further associated with the determined context in response to the one or more recognized words or phrases satisfying the predetermined confidence level;
- [g] formulating, at the parser, the request contained in the utterance in

accordance with a grammar used by a domain agent associated with the determined context;

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

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

(’006, 25:49-26:21 (Claim 1) (bracketed notation (e.g., “[a]”) added).)

82. The preamble recites “[a] **method for processing natural language speech utterances with context-specific domain agents, comprising.**” As shown below, the claimed method is disclosed and rendered obvious by the prior art.

(a) **“receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;” (1[a])**

83. The specification of the ’006 patent describes a “**speech unit**” as comprising a microphone and speaker, with other optional elements such as a filter, speech coder, and transceiver. (’006, 10:28-36 (“The speech unit 128 includes one or more microphones, for example array microphone 134, to receive the utterances of the user. The speech received at the microphone 134 may be processed by filter 132 and passed to the speech coder 138 for encoding and compression. In one embodiment, a transceiver module 130 transmits the coded speech to the main unit 98.”), 10:36-39 (“Coded speech received from the main unit 98 is detected by the

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

transceiver 130, then decoded and decompressed by the speech coder 138 and annunciated by the speaker 136.”), 10:51-53 (“In an alternative embodiment, wires connect the speech unit 128 and the main unit 98, eliminating the need for one speech coder 138.”), Fig. 1 (showing speech unit including, among other things, microphone 134 and speaker 136).) The speech unit is connected, via wires or wirelessly (e.g., transceiver), to a “main unit” processing device. (’006, 10:34-36, 10:51-53, Fig. 1.)

84. A person of ordinary skill in the art would have understood and found it obvious that Coffman’s system would have **“a speech unit coupled to a processing device”** as claimed. Coffman similarly describes user devices for its conversational computing system having a microphone and speaker—i.e., a **“speech unit”**—for receiving speech input from a user and providing audio output (e.g., via text-to-speech). (Coffman, e.g., 53:8-10 (“Voice input from a microphone (e.g. the standard voice input) can be arguments of function calls with the sequence of words, phones, or user identity or queries (symbolic language representation provided by NLU).”), 37:14-22 (“Spoken input from a speech client (e.g., telephone 600) is subject to a speech recognition process 604 and other input (e.g., keyboard, mouse clicks etc) are subject to NLU processing 605.”), 35:2-3 (“listen to the output of a task”), 39:12 (“audio only output”), 39:30 (“voice output”), Fig. 6 (telephone 600, “speech synthesis,” “speech out”).) Coffman further teaches that these components

are coupled to components for processing input from and output to them (“**processing device**”), such as conventional processing components (e.g., CPU) of a desktop computer or PDA for running a conversational virtual machine (“CVM”) and related software. (Coffman, 10:26-28 (“A CVM platform can also be implemented with PvC (pervasive computing) clients as well as servers.”), 12:5-7 (“Referring now to Fig. 1, a block diagram illustrates a conversational computing system (or CVM system) according to an embodiment of the present invention, which may be implemented on a client device or a server.”), 13:1-3 (“The CVM system may be built on top of a conventional OS and APIs 15 and conventional device hardware 16 and located on a server or any client device (PC, PDA, PvC).”), 28:29-31 (“In general, the CVM 401 operates by converting conversational and conventional input streams into multiple actions and produces sorted output to a user through conversational and/or conventional output streams.”), 20:26-29 (describing prioritizing “CPU cycles”), 45:29-32 (“CPU load” and “CPU cycles”), Figs. 1 & 6.)

85. Coffman further discloses “**receiving, at a speech unit coupled to a processing device, a ... speech utterance.**” Coffman describes receiving spoken input from a person using, for example, a telephone or microphone. (Coffman, e.g., 53:8-10 (“Voice input from a microphone (e.g. the standard voice input) can be arguments of function calls with the sequence of words, phones, or user identity or

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

queries (symbolic language representation provided by NLU).”), 37:14-22 (“Spoken input from a speech client (e.g., telephone 600) is subject to a speech recognition process 604 and other input (e.g., keyboard, mouse clicks etc) are subject to NLU processing 605.”), 41:2-5 (“Based on the modality (pointer, keyboard, file, speech), the task dispatcher 402 redirects the stream to the appropriate conversational subsystems or conventional subsystem with speech inputs being transcribed/understood. The output of these subsystems is run down the context stack 405 to extract the active query and complete it.”), 53:22-26 (“The input process described above in accordance with one aspect of the present invention considers that speech or any other input stream is included as a classical input/output stream that is susceptible to all forms of processing typically reserved for character or binary input.”).)⁴ Figure 6 of Coffman, reproduced below, illustrates receiving a user generated speech utterance via telephone 600:

⁴ All underlining added to quoted material unless noted.

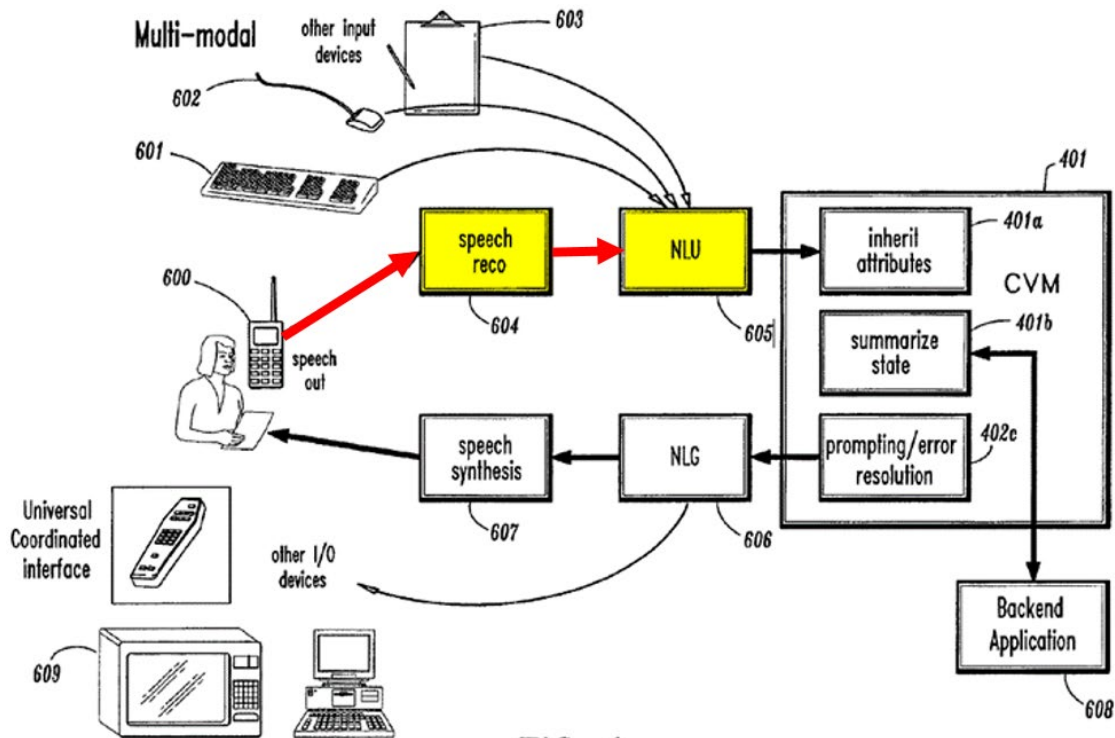


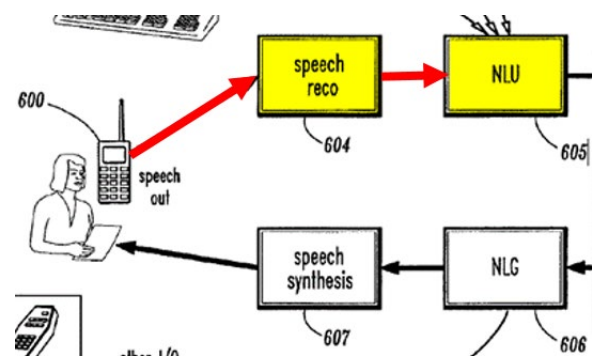
FIG. 6

(Coffman, Fig. 6 (annotations added); *see also id.*, 37:14-22 (“Spoken input from a speech client (e.g., telephone 600)...”).)

86. Coffman further discloses receiving spoken input that is a “**natural language speech utterance.**” In the context of the ’006 patent, and speech recognition and processing generally, “**natural language**” generally speaking refers to the forms of language used by a person for communication, as opposed to artificial or formal languages such as logics, algebras, and programming languages. (See, e.g., Dafydd Gibbon et al., *Handbook of Multimodal and Spoken Dialogue Systems : Resources, Terminology, and Product Evaluation* (2000) (“Gibbon”) [EX1008], p.444 (definition of “natural language”).) For example, English is a natural

language, while C++ and Java are programming languages. Consistent with this understanding, Coffman discloses several examples of natural language speech utterances, such as “I would like to open this,” “it should be synchronized with my PDA,” and “e-mail from my mother.” (Coffman, 6:32-7:3 (“For example, a mouse click or pointer/stylus pointing action followed by the command ‘I would like to open this’ is disambiguated into a set of attribute value pairs: Command: open, Object: Windows or task selected by the last mouse click.”), 48:1-3 (“Since user identity can be associated in the context of each discourse, command ambiguity can be immediately and transparently resolved (e-mail from my mother is correctly understood, independently of the user).”), 56:25-57:2 (“grocery list,” “it should be synchronized with my PDA,” and “Do it!”).)

87. Coffman’s receipt (and use) of spoken “**natural language**” input is also confirmed by Coffman explaining that the spoken input, once recognized, is subjected to natural language parsing and understanding (“NLU”) processes, as illustrated by Figure 6 (excerpted and annotated at right). (Coffman, e.g., Fig. 6, 53:22-26 (“User inputs can be represented by their transcriptions or their mappings into a symbolic language after parsing and NLU.”), 53:8-10 (“Voice input from a



microphone (e.g. the standard voice input) can be arguments of function calls with the sequence of words, phones, or user identity or queries (symbolic language representation provided by NLU).”), 37:14-22 (“Spoken input from a speech client (e.g., telephone 600) is subject to a speech recognition process 604 and other input (e.g., keyboard, mouse clicks etc) are subject to NLU processing 605.”), 6:6-8 (“In another aspect, the CVM provides the capability to have natural dialog with NLU, NLG, contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains).”), 5:1-4 (“In one aspect of the present invention, the CVM utilizes conversational subsystems (which may be local or distributed) including speech recognition, speaker recognition, text-to-speech, natural language understanding and natural dialog generation engines to understand and generate dialog between and user and machine.”), 37:29-32 (“The key conversational aspect is in the input procedure wherein commands can be entered in NLU (to provide natural language understanding of input queries) or in FSG mode (for constrained input according to rules: grammar and vocabulary, as opposed to free natural input).”).)

88. Coffman further discloses and renders obvious that the speech utterance “**contains a request**,” explaining that the system “convert[s] voice requests into queries.” (Coffman, 13:8-13 (“In general, conversational sub-systems 18 are

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

responsible for converting voice requests into queries and converting outputs and results into spoken messages using the appropriate data files 17 (e.g., contexts, finite state grammars, vocabularies, language models, symbolic query maps etc.)”).) The example spoken utterances from Coffman that I noted above—“I would like to open this,” “it should be synchronized with my PDA,” and “e-mail from my mother”—are examples of spoken utterances that “**contain[] a request.**” (Coffman, 6:32-7:3 (“I would like to open this”), 48:1-3 (“e-mail from my mother”)⁵, 56:25-57:2 (“grocery list,” “it should be synchronized with my PDA,” and “Do it!”).)

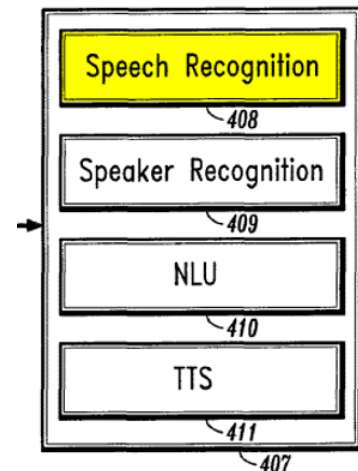
⁵ This would be understood as part of a spoken request to “open” or “find” the specified e-mail in an e-mail reading system such as described at Coffman, 56:12.

- (b) “recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables, wherein recognizing the one or more words or phrases contained in the utterance includes:” (1[b])

89. Coffman discloses “a speech recognition engine coupled to the processing device.” Coffman explains that its system includes various

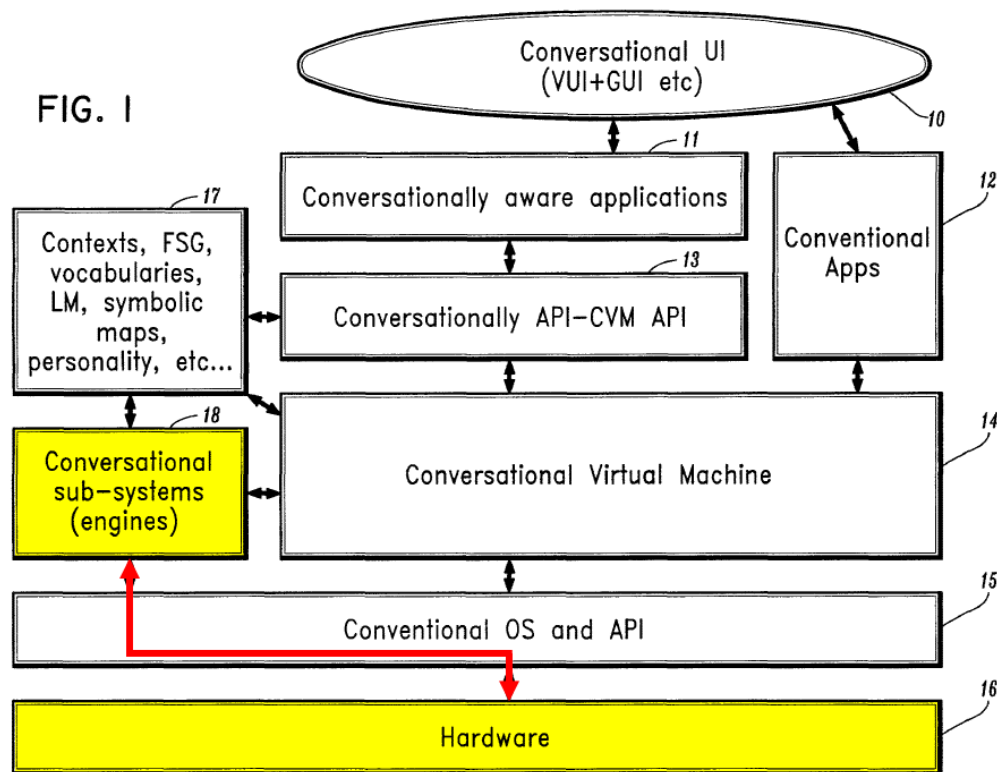
“conversational subsystems,” including a “speech recognition engine” (illustrated at right (Fig. 4 (excerpt))).

(Coffman, 10:28-30 (“In general, the CVM provides conversational APIs and protocols between conversational subsystems (e.g. speech recognition engine, text-to speech etc.) and conversational and/or conventional



applications.”), 29:1-4 (“The conversational system 400 further comprises a plurality of conversational resource subsystems (engines) 407 including, for example, a speech recognition system 408”), Fig. 4.) A person of ordinary skill in the art would have understood that Coffman’s speech recognition engine, as a software component running on a conventional device such as a desktop computer or PDA, would be “**coupled to the processing device.**” (Coffman, 13:1-3 (“The CVM system may be built on top of a conventional OS and APIs 15 and conventional device hardware 16 and located on a server or any client device (PC, PDA, PvC).”),

20:26-29 (describing prioritizing “CPU cycles”), 45:29-32 (“CPU load” and “CPU cycles”), Figs. 1 & 6.) For example, Figure 1 of Coffman depicts conversational sub-systems (engines) 18—which includes a “speech recognition engine”—coupled to hardware 16:



(Coffman, Fig. 1 (annotations added); *see also id.*, 13:1-3, 29:1-4.)

90. Coffman further makes clear that its speech recognition engine “**recogniz[es] one or more words or phrases contained in the utterance,**” explaining that “[s]poken input from a speech client (e.g., telephone 600) is subject to a speech recognition process 604 and other input (e.g., keyboard, mouse clicks etc) are subject to NLU processing 605.” (Coffman, 37:20-22; *see also id.*, Fig. 6

(depicting input to and output from “speech reco” 604), 38:9-11 (“Each input stream is converted into its ASCII transcription and aligned with input time- marks via speech recognition processing 604.”).)

91. A person of ordinary skill in the art would have also understood and found it obvious that the “**recognizing**” in Coffman’s system would be performed “**using information in one or more dictionary and phrase tables.**” As explained below, the “**one or more dictionary and phrase tables**” correspond in Coffman to n-gram language models used for speech recognition.

92. In particular, Coffman explains that each conversationally-aware application registers with a conversational virtual machine (“CVM”), and provides its language model(s) for controlling the application. (Coffman, e.g., 57:22-58:11, 7:12-14, 13:8-13.) By way of background, in a speech recognition context, a “language model” generally speaking refers to a formal representation of the structure of a natural language. (See, e.g., Gibbon [EX1008], p.430 (definition of “language model”).) “In most speech recognisers language models consist of an n-gram model, i.e. probabilities of the occurrence of words, words pairs, sequences of three words, etc.” (Gibbon, p.430; see also *id.*, pp.444 (defining “n-gram language model” as “a stochastic language model that is based on a n-gram grammar, i.e. on conditional probabilities depending only on the (n-1) immediate predecessor

words.”), 479 (defining “stochastic language model” as “a language model that assigns probabilities to the allowed word sequences; typically all word sequences have a non-zero probability.”).) A “trigram” language model therefore refers to an n-gram language model that uses the probabilities of the occurrence of sequences of three words (i.e., the “n” in “n-gram” is three). (Gibbon, p.488 (defining “trigram” as “a sequence of three words”).)

93. Coffman does not provide a detailed discussion of how a language model is used for speech recognition. This is not surprising, because speech recognition using language model(s) was well known and I understand that a patent need not teach, and preferably omits, what is well known in the art. Coffman does expressly incorporate by reference U.S. Patent No. 5,897,616 (Kanevsky [EX1004]) (Coffman, 37:10-13), which further explains that a speech recognition engine may be implemented using “a trigram language model.” (Kanevsky, 13:50-56.) Kanevsky describes such a speech recognition engine as an “existing component” and explains that “[s]uch systems, for example, are disclosed in IBM’s speech dictation engine ‘Simply Speaking’ and in ‘Transcription of Radio Broadcast News with the IBM Large Vocabulary Speech Recognition System’, P.S. Gopalakrishnan [sic, Gopalakrishnan] et al., Proceeding Speech Recognition Workshop, Arden House, Drapa (1996).” (Kanevsky, 13:48-61.)

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

94. Under the proposed combination of Coffman and Kanevsky, therefore, the speech recognition process to identify one or more trigrams corresponding to the utterance discloses “**recognizing...one or more words or phrases contained in the utterance**” because the trigrams comprise words and sequences of words (i.e., phrases). (See Gibbon [EX1008], p.488 (defining “trigram” as “a sequence of three words”).) And moreover, recognizing trigrams from the user’s spoken input (“**words or phrases contained in the utterance**”) under the proposed combination “**us[es] information in one or more dictionary and phrase tables.**” This is because a person of ordinary skill in the art would have understood and found it obvious to organize the n-gram language model(s) used by the Coffman’s conversational system as a **table**. For example, Sutcliffe [EX1016] explains that “[m]ost large-vocabulary English speech recognition systems use a word N-gram language model of English grammar: syntactic knowledge is captured in a large table of word bigrams (pairs), trigrams (triples), ... N-grams (see surveys of large-vocabulary speech recognition systems, e.g. HLT Survey, 1995; comp.speech, 1996).” (Richard F. E. Sutcliffe et al., *Industrial Parsing of Software Manuals* (1996) (“**Sutcliffe**”) [EX1016], p.27.) U.S. Patent Application Publication No. 2003/0144846 (“Denenberg”) (on which I am named as a co-inventor) similarly explains that “[a]nother common recognizer is a natural language speech recognizer

based upon the N-gram language model which works from tables of probabilities of sequences of words.” (Denenberg [EX1014], [0040].) Thus, a trigram language model itself comprises “**one or more dictionary and phrase tables**” because it is organized as a table and includes a list of words and sequences of words to refer to during speech recognition. (*See, e.g., Merriam-Webster’s Collegiate Dictionary* (10th ed. 1994) [EX1015], p.322 (defining “dictionary” as encompassing “a list (as of items of data or words) stored in a computer for reference (as for information retrieval or word processing)”).)

95. **Rationale and Motivation to Combine (Coffman and Kanevsky).** It would have been obvious to a person of ordinary skill in the art to combine Coffman and Kanevsky, predictably resulting in the conversational system of Coffman in which language models provided by the conversationally-aware applications and used by the system for speech recognition were n-gram language models, such as a trigram language model. Combining the teachings of Coffman and Kanevsky in this manner would have therefore resulted in “**recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables**” as claimed.

96. Both Coffman and Kanevsky are analogous references in the same field

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

of speech recognition and processing as the '006 patent. ('006, 1:20-22; Coffman, e.g., 1:7-8 (“systems and methods for conversational computing”), Figs. 1, 4 & 6; Kanevsky, e.g., 1:15-21 (“automatic speech recognition, text-independent speaker identification, natural language understanding techniques”).) Coffman and Kanevsky would have also been reasonably pertinent to a number of problems facing the patent inventors, including how to implement a system for a person to interact with a computer system using speech and techniques for performing speech recognition.

97. A person ordinary skill in the art would have been motivated to combine Coffman and Kanevsky. To begin with, as noted above, Coffman expressly incorporates Kanevsky by reference. (Coffman, 37:10-13.) Therefore, I am informed that Kanevsky is properly considered part of the disclosures of Coffman for purposes of applying it as prior art. I further note that both references also share a named co-inventor, Stephane Herman Maes, and identify IBM as the assignee/applicant. (Coffman, face page; Kanevsky, face page.) Therefore, a person of ordinary skill in the art would have naturally consulted both of these references to understand the features and functionality of Coffman’s conversational system, or to actually implement or adapt the system and features it provides. A person of ordinary skill in the art would have regarded Coffman and Kanevsky as a collective

body of work, with Kanevsky complementing Coffman's disclosures. Because Coffman incorporates Kanevsky by reference, and both references share a common co-inventor and assignee, and in light of their overlapping subject matter, a person of ordinary skill in the art would have found it odd to even question their combinability, or whether they can properly be considered together to understand Coffman's conversational system.

98. Moreover, as explained, Coffman already discloses that a conversationally-aware application provides its "language models" to the conversational system in connection with processing a user's input. A person of ordinary skill in the art would have understood that a language model in this context likely refers to an n-gram language model (*e.g.*, Gibbon [EX1008], pp.430 (definition of "language model"), 444 (definition of "n-gram language model")), and Kanevsky provides express confirmation, explaining that a speech recognition engine can be implemented using a trigram language model (Kanevsky, 13:50-56). Moreover, n-gram language models were one of a number of finite ways of implementing speech recognition, and the use of an n-gram language model as taught by Kanevsky would have involved the application of a known technique to implement or improve Coffman's speech recognition functionality in the same way.

99. A person of ordinary skill in the art would have had a reasonable

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

expectation of success. The implementation and use of n-gram language models for speech recognition was well-known and would have involved the application of conventional techniques and technologies. For example, as noted, the '006 patent itself acknowledges that “[s]peech recognition has steadily improved in accuracy and today is successfully used in a wide range of applications.” ('006, 1:45-48.) Kanevsky describes a speech recognition engine using a trigram language model as an “existing component,” disclosed in, for example, “IBM’s speech dictation engine ‘Simply Speaking’ and in ‘Transcription of Radio Broadcast News with the IBM Large Vocabulary Speech Recognition System’, P.S. Gopialakrishnan [sic, Gopalakrishnan] et al., Proceeding Speech Recognition Workshop, Arden House, Drapa (1996).” (Kanevsky, 13:48-61.) A person of ordinary skill in the art would have found the disclosures of Coffman and Kanevsky sufficiently detailed to implement the techniques on which I rely, without undue experimentation.

100. As explained for the following limitations, the prior art also discloses and renders obvious that **“recognizing the one or more words or phrases contained in the utterance includes”** the following steps.

- (i) **“dynamically updating the information in the one or more dictionary and phrase tables based on a dynamic set of prior probabilities or fuzzy possibilities;” (1[b][i])**

101. As explained below, 1[b][i] describes nothing more than the

conventional use of an “adaptive” n-gram language model (such as bigram or trigram) for speech recognition. In an adaptive n-gram language model, the probabilities associated with the n-grams are dynamically updated as the speech recognition system is used. (*See, e.g.*, Gibbon [EX1008], p.371 (defining “adaptive language model” as a language model that “adapts [its] probabilities to the most recent history, say the last 100 to 1000 predecessor words.”).)

102. As explained for 1[b], **“the dictionary and phrase tables”** under the proposed combination correspond to one or more trigram language models provided by conversationally-aware applications to a conversational system and used for speech recognition. A trigram language model, as explained, refers to an n-gram language model that uses the probabilities of the occurrence of sequences of three words (i.e., the “n” in “n-gram” is three). (Gibbon, p.488 (defining “trigram” as “a sequence of three words”); *see also id.*, pp.430 (“In most speech recognisers language models consist of an n-gram model, i.e. probabilities of the occurrence of words, words pairs, sequences of three words, etc.”), 440 (defining “n-gram language model” as “a stochastic language model that is based on a n-gram grammar, i.e. on conditional probabilities depending only on the (n-1) immediate predecessor words.”).)

103. An n-gram language model (such as a trigram language model) includes

a “**set of prior probabilities**”—the set of probabilities associated with the trigrams in the trigram language model used by the speech recognition engine and maintained by the conversational system as a result of application registration. A person of ordinary skill in the art would have understood and found it obvious that the probabilities associated with the trigrams in the language model are “**prior probabilities**” because they each correspond to the probability of observing some phenomenon estimated from previously collected training data, but independent of future observations. (*See, e.g.,* Gibbon [EX1008], p.460 (defining “prior probability”); *see also* Sutcliffe [EX1016], p.27 (explaining that a language model is “extracted or learnt from a corpus, a representative set of texts in the domain of the speech recognizer.”).)

104. With respect to a “**dynamic set of prior probabilities**,” this is disclosed and rendered obvious through the conventional use of an “adaptive” n-gram language model for speech recognition. As noted, in an “adaptive” n-gram language model, the probabilities associated with the n-grams (such as bigram or trigram) are dynamically updated as the speech recognition system is used. (*See, e.g.,* Gibbon [EX1008], p.371 (defining “adaptive language model” as a language model that “adapts [its] probabilities to the most recent history, say the last 100 to 1000 predecessor words.”).) Coffman itself suggests modifying language models

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

on a per-user basis, describing collection of user-specific information in order to perform “more robust recognition,” and explaining that “any information can be accumulated to recognize the user,” including “the usual phrasing of a query, the type of query phrased, command frequency (often used, not often used), preferred applications, time or usage, etc.” (Coffman, 37:5-10.) Although neither Coffman nor Kanevsky appears to expressly detail using them, adaptive n-gram language models and their advantages were well known. The use of adaptive n-gram language models would have been obvious over U.S. Patent No. 6,157,912 (**Kneser** [EX1005]). Kneser discloses that it is advantageous to adapt a language model by updating its probabilities, explaining:

These language model values represent the probabilities that given word sequences of limited length are present; they have been obtained from a comparatively large quantity of training texts, which may also be formed by written texts. Long-term changes in the use of language, for example subject matter or style, can accordingly not be taken into account with these language model values [probabilities]. Since the quality of the adaptation of the language model values has a considerable influence on the reliability of the speech recognition, it is desirable to adapt the language model values as well as possible to the given text by means of a small quantity of updated text. This adaptation may take place during the process of recognizing a given text or alternatively subsequent to the recognition of part of a text, after

recognition errors have been manually corrected.

(Kneser, 1:22-33; *see generally id.*, 1:13-37.)

105. **Rationale and Motivation to Combine (Coffman and Kanevsky with Kneser)**. It would have been obvious to a person of ordinary skill in the art to further combine Coffman and Kanevsky with Kneser's teachings regarding an adaptive language model, predictably resulting in language models in Coffman's conversational system having a "**dynamic**" set of prior probabilities. Therefore, under the proposed combination, because the prior probabilities are included in the trigram language model(s) ("**the dictionary and phrase tables**"), 1[b][i] simply corresponds to dynamically updating the trigram language model(s) in accordance with model adaptation as taught by Kneser.

106. Like Coffman and Kanevsky, Kneser is an analogous reference in the same field of speech recognition and processing as the '006 patent. ('006, 1:20-22; Kneser, e.g., 1:5-11 ("The invention relates to the automatic recognition of speech in which a language model with language model values are used, which values are determined from a comparatively long training text, while an adaptation of the language model is carried out for certain cases, in particular for certain types of texts.")) Kneser would have also been reasonably pertinent to a number of problems facing the patent inventors, including techniques for performing speech recognition

and how to implement a speech recognition system that uses a “dynamic” set of prior probabilities.

107. A person ordinary skill in the art would have been motivated to combine Kneser with Coffman and Kanevsky. Kneser, as shown above, provides express motivation for incorporating “adaptive” language model techniques into Coffman’s conversational system—improving speech recognition reliability and performance. Coffman itself confirms this motivation in the context of collecting user-specific information in order to perform “more robust recognition”:

It is to be appreciated that conversational biometrics can be used to collect any context and meta information on the user not only to customize or adapt for purposes of user preferences or to authorize a query, but also to use the information to perform more robust recognition. Accordingly, any information can be accumulated to recognize the user. Namely, the usual phrasing of a query, the type of query phrased, command frequency (often used, not often used), preferred applications, time or usage, etc.

(Coffman, 37:5-10.) Adaptation based on usual phrasing and frequency of commands, as Coffman describes above, could be easily accomplished by adjusting the probabilities in the n-gram language model, and adaptive language models as taught by Kneser provide a way in which Coffman’s system could advantageously improve speech recognition reliability and performance. The use of adaptive

language models was well known in the art (*e.g.*, Gibbon [EX1008], p.371 (defining “adaptive language model”)), and a person of ordinary skill in the art would have appreciated that using adaptive language models in Coffman’s conversational system for speech recognition would have improved a similar system in the same way.

108. A person of ordinary skill in the art would have had a reasonable expectation of success. As noted, the use of adaptive language models was well known in the art, and implementing the language models used in Coffman’s conversational system as adaptive language models would have involved conventional techniques and technologies. A person of ordinary skill in the art would have found the disclosures of Coffman, Kanevsky, and Kneser sufficiently detailed to implement the techniques on which I rely, without undue experimentation.

109. Therefore, the combination of Coffman, Kanevsky and Kneser discloses and renders obvious “**dynamically updating the information in the one or more dictionary and phrase tables**” (dynamically updating probability information in trigram language model(s)) “**based on a dynamic set of prior probabilities or fuzzy possibilities**” (based on adaptive prior probability information).

- (ii) **“determining an identity associated with a user that spoke the utterance based on voice characteristics associated with the utterance; and” (1[b][ii])**

110. Coffman discloses **“determining an identity associated with a user that spoke the utterance”** because its conversational system includes functionality for “speaker recognition” and “speaker identification,” such as “text-independent speaker identification” using a user’s command or request. For example, Coffman explains that “the CVM utilizes conversational subsystems (which may be local or distributed) including speech recognition, speaker recognition, text-to-speech, natural language understanding and natural dialog generation engines to understand and generate dialog between and user and machine.” (Coffman, 5:1-4.) Figure 4 of Coffman depicts conversational system 400 with speaker recognition system 409:

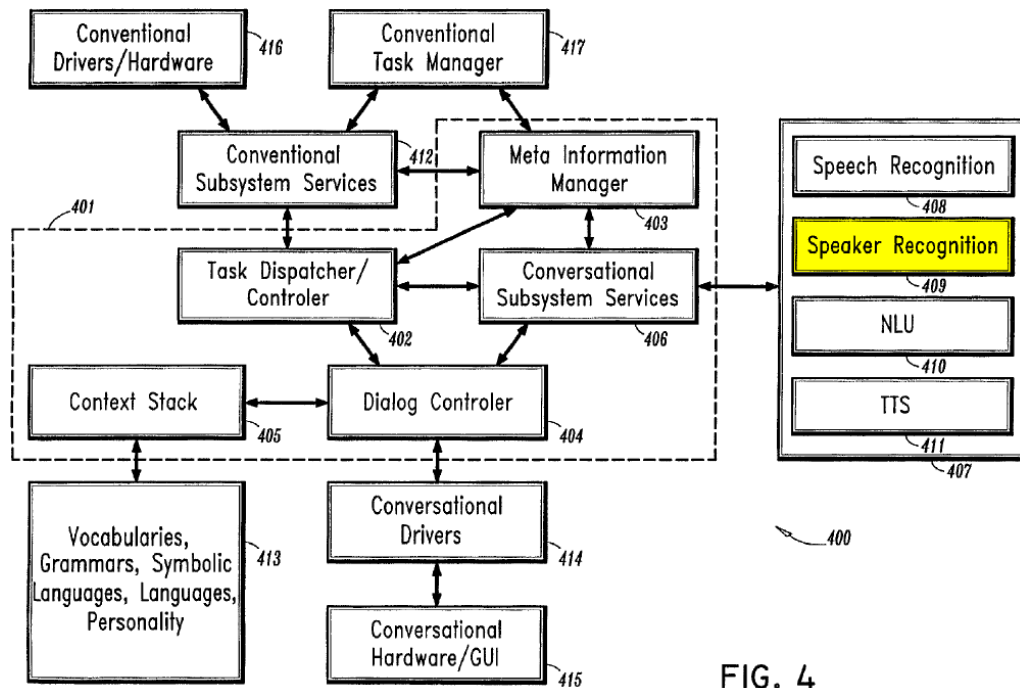


FIG. 4

(Coffman, Fig. 4 (highlighting added); *see also id.*, 29:1-4, 15:13-16:24 (listing various speaker-identification-related “conversational foundation classes” including “Acoustic_Identify_speaker,” “Acoustic_Verify_speaker” and “Speech_biometrics_identify.”).) Coffman further discloses that “[i]n a speech-based system, each command can be used to perform text-independent speaker identification.” (Coffman, 47:25-27; *see also id.*, 8:16-18 (“A user is automatically identified whenever a query to a resource is made.”).) “Text-independent speaker identification,” as its name suggests, is a form of speaker identification using some features of a speech signal to determine the identity of the speaker of a given utterance that does not require the utterance to comprise specific linguistic content.

(See Gibbon [EX1008], pp.474 (defining “speaker recognition” as “[a]ny decision-making process that uses some features of the speech signal to determine some information on the identity of the speaker of a given utterance”), 486 (defining “text-independent speaker recognition system” as “[a] speaker recognition system for which the linguistic content of test speech utterance varies across trials”).) The very fact that text-independent speaker identification is *text-independent* implies that the identification does not rely on specific words but instead uses voice characteristics.

111. Kanevsky, which Coffman specifically cites for speaker identification (Coffman, 37:5-13), further describes text independent speaker recognition that involves comparing “**voice characteristics**” of a sample of the user’s utterance with a model to verify the user’s identity. (Kanevsky, e.g., 6:66-7:8, 6:12-15 (“The identity claim may be made from other information provided by the speaker or voice characteristics...”), 2:6-8 (describing prior art systems that “employ voice-based verification” using “acoustic characteristics of the utterances spoken by the user”).) I discussed the rationale and motivation to combine Coffman and Kanevsky in my analysis of 1[a], which applies here as well. A person of ordinary skill in the art would have therefore understood and found it obvious to “**determin[e] an identity associated with a user that spoke the utterance based on voice characteristics associated with the utterance.**”

- (iii) **“associating the one or more recognized words or phrases and a pronunciation associated with the one or more recognized words or phrases with the determined identity and the request contained in the utterance in response to the one or more recognized words or phrases satisfying a predetermined confidence level;” (1[b][iii])**

112. As explained below, Coffman’s process of matching recognized words and phrases from the utterance to identify an acceptable context discloses **“associating the one or more recognized words or phrases and a pronunciation associated with the one or more recognized words or phrases with the determined identity and the request contained in the utterance.”**

113. Coffman explains that the conversational system includes a “context stack” where the state and history of each active process, such as an application process, is tracked in its own “context”:

The core of the CVM 401 is the context stack 405 which operates and is managed under the control of the dialog controller 404 (it is to be understood that the context stack 405 id [sic; is] directly related to the global history and meta information repository discussed above). In general, the context stack 405 accumulates the context (i.e., full query arguments list of attribute value n-uples^[6], and state/mode) of each

⁶ An “n-uple” refers to a collection of “n” elements, where “n” is a number. Therefore, for example, a “4-uple” (or “quadruple”) refers to a collection of four

active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 (or at least identifiers of such conversational arguments) for the different engines that may be needed for input understanding (e.g., files or arguments that the engines use for performing their respective tasks such as active FSG, topic, vocabulary, HMM (hidden markov models), voiceprints, language models or possible queries for a speech input). In other words, the term “context” refers to the state of each discourse (whether active or normative), which keeps track of the past history of the discourse, its current state, and the specific characteristics and full query arguments of the corresponding task (e.g., vocabulary file, language model, parsing, tags, voiceprint, TTS rules, grammar, NLU etc. of each active task/process) along with any data needed for input understanding. It is to be appreciated that, in a distributed CVM (as explained below), the context stack (as well as other CVM components) may be directly associated with networked services (i.e., distributed over the network) (as described above with respect to the context and global history).

(Coffman, 29:18-30:3; *see also id.*, 30:4-10 (“More specifically, each new task, process, or thread creates a new stack entry and is associated with a discourse. Each

elements. Coffman’s “attribute value n-uple” refers to an n-uple having “n” attribute-value pairs. Coffman’s example, “Command: open, Object: Windows or task selected by the last mouse click” (Coffman, 6:32-7:3), is a 2-uple (or “dupe”) because it includes two attribute-value pairs.

application may be associated with multiple discourses (e.g. the application management discourse and the application content navigation discourses). Each context associated with a given discourse comprises the latest requests made to the corresponding process/task/thread as well as the latest output. The context of a given discourse is also associated with, e.g., any active grammars, vocabularies and symbolic language which maps the actual query.”.)

114. Coffman explains that each context can be associated with a user identity and that a context is also associated with the user’s request (as I explain in more detail below, under the proposed combination, the claimed “**associating**” takes place following speaker identification based on the recognized words and phrases satisfying a predetermined confidence level). With respect to “**the identity of the user**,” Coffman explains that “user identity can be associated in the context of each discourse,” so that “command ambiguity can be immediately and transparently resolved (e-mail from my mother is correctly understood, independently of the user).” (Coffman, 48:1-3; *see also id.*, 46:26-31 (“the stack of contexts should contain the identity of the user as the most recently authenticated identity”), 48:11-14 (“[W]ith respect to conversational multi-users and conversational security, it is to be appreciated that dialogs, categories, meta-information, and access to resources can be a function of the identity of the user and its associated meta-information

history. And conversely, the conversational information collected on a query can be used to recognize the user.”.) With respect to “**the request contained in the utterance,**” Coffman teaches that the recognized words and phrases are used to match a context and its associated formulated request (e.g., in symbolic language or application call/query):

Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active. Until the active command is completed, or until a new command is provided, the selected context is marked active, and pushed to the top of the context stack 405.

(Coffman, 42:1-7; *see also id.*, 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other application is stored to activate if the recognized query is rejected by the user.”).) Therefore, as a result of identifying a context, “**the one or more recognized words or phrases**” are associated with the user’s identity and the request associated with the context (“**with the determined identity and the request contained in the utterance**”).

115. 1[b][iii] further recites “**associating ... a pronunciation associated with the one or more recognized words or phrases with the one or more recognized words or phrases with the determined identity and the request contained in the utterance.**” This would have been obvious over the proposed combination in at least two separate ways. **First**, Coffman explains that application registration involves providing baseforms (pronunciation rules) for the active vocabulary to the conversational system. (Coffman, e.g., 57:22-27, 60:30-61:2, 24:30-25:3.) A person of ordinary skill in the art would have therefore understood that successful recognition of words and phrases would have involved successfully matching the user’s utterance to the pronunciation rules. Therefore, the subsequent matching of the recognized words and phrases to a context would also result in associating the pronunciations specified by the baseforms of those words and phrases with the user’s identity and the request associated with the context.

116. **Second**, as explained above for 1[b][ii], identifying the user based on voice characteristics of the user’s utterance is disclosed by and obvious over Coffman and Kanevsky. A person of ordinary skill in the art would have understood that a user’s pronunciation of words and phrases would be reflected in the voice characteristics of their utterance. For example, Kanevsky describes using information about a user’s voice such as “prototypes, prosody, speech rate, accent,

etc.” to identify the user. (Kanevsky, 9:39-42, 10:34-38, 11:53-58; *see also, e.g.*, Gibbon [EX1008], pp.493 (defining “voice characteristic” as “[t]hose aspects of speech which remain relatively constant over longer stretches of speech, and constitute the background against which segmental and prosodic variation is produced and perceived (e.g. mean pitch level, mean loudness, mean tempo, harshness, creak, whisper, tongue body orientation, dialect)”)), 461 (defining “prosody” as covering “properties of pronunciation”); Merriam-Webster’s Collegiate Dictionary (10th ed. 1994) [EX1015], p.319 (defining “dialect” as “a regional variety of language distinguished by features vocabulary, grammar, and pronunciation from other regional varieties and constituting together with them a single language”).) Therefore, a person of ordinary skill in the art would have understood and found it obvious that, by performing speaker identification in accordance with 1[b][ii], the user’s pronunciation of the recognized words and phrases would have been “associated” **“with the determined identity and the request contained in the utterance.”**

117. 1[b][iii] further recites that the claimed “associating” occurs “**in response to the one or more recognized words or phrases satisfying a predetermined confidence level.**” This also would have been obvious over Coffman and Kanevsky. The concept of a confidence level has been well known

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

in the field of speech recognition and processing since at least the 1970s, and I personally implemented this concept in my work starting in 1979. (*See, e.g.*, C. Schmandt, “The Intelligent Ear: A Graphical Interface to Digital Audio” (1981) [EX1017], pp.394-395 (describing “degree of certainty for recognition of each word”).) Coffman itself mentions “confidence and ambiguity results” (Coffman, 27:16), applications providing a confidence level for its interpretation of an input (Coffman, 31:26-28) and “provid[ing] feedback to the user for incomplete or ambiguous queries/commands” (Coffman, 40:16-17) but does not detail how confidence results may be used by the system. But it would have been obvious in view of Kanevsky to perform the claimed “**associating**” under the proposed combination only when speaker identification based on the recognized words and phrases satisfied a predetermined confidence level. Doing so would help ensure that the system did not incorrectly associate an input with a particular user, context and query and would further the security objectives described in both Coffman and Kanevsky. (*E.g.*, Coffman, 37:5-14 (describing conversational biometrics for query authorization), 8:10-15 (describing preventing unauthorized access), 46:17-32 (describing user authentication and authorization), 48:1-3, 48:11-14.) For example, Kanevsky describes calculating a score from text-independent speaker recognition, comparing the score to “a predetermined threshold value,” and allowing access only

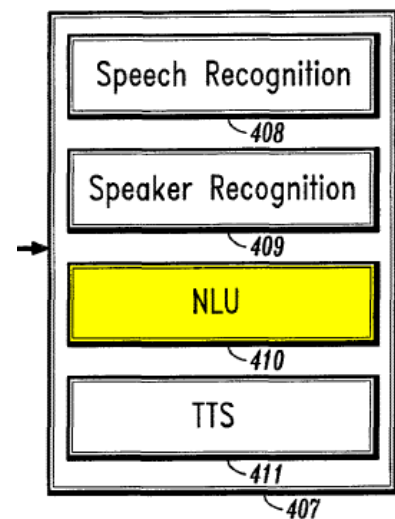
when the score is “above or within an acceptable predetermined range of the threshold value”:

Substantially simultaneous with the spoken utterances provided by the user **12**, the central server **22** may process a user voice sample, for example, from the initial dialog from the potential user and/or from the answer or answers uttered by the potential user, through a text-independent speaker recognition module **52** via link **54** in order to verify (confirm) the user’s identity. The module **52** utilizes the user model previously built (as will be explained in the context of FIG. 5) and which is presented to module **52** from user model **20** via link **51**. It is to be appreciated that such speaker recognition process of module **52** is preferably implemented by default, regardless of the partial score(s) achieved in the question/answer phase, in order to provide an additional measure of security with regard to service/facility access. The voice sample is processed against the user model and another partial score is generated by the score estimator **44**. Based on a comparison of a combination of the partial scores (from the question/answer phase and the background speaker verification provided by module **52**) versus a predetermined threshold value, the central server **22** decides whether or not to permit access to the service/facility to the user **12**. If the combined score is above or within an acceptable predetermined range of the threshold value, the central server **22** may permit access, else the server may decide to deny access completely or merely repeat the process.

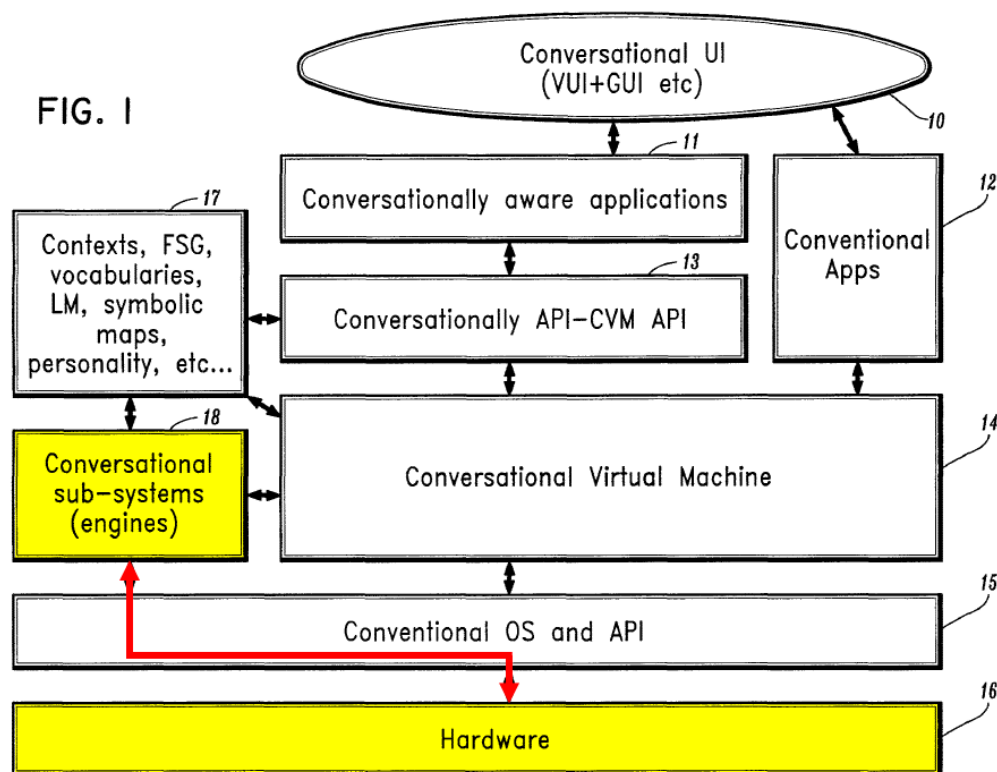
(Kanevsky, 6:66-7:24.) A person of ordinary skill would have understood that a score would be compared to a “**predetermined confidence level**” because the score would reflect the system’s confidence whether the context was acceptable.

- (c) “parsing, at a parser coupled to the processing device, the one or more recognized words or phrases to determine a meaning associated with the utterance and a context associated with the request contained in the utterance, wherein the one or more recognized words or phrases are further associated with the determined context in response to the one or more recognized words or phrases satisfying the predetermined confidence level;” (1[c])

118. Coffman discloses “a parser coupled to the processing device.” As explained for 1[b], Coffman explains that its system includes various “conversational subsystems”; those subsystems include “a natural language understanding and natural language parsing system [NLU] 410” (illustrated at right (Fig. 4 (excerpt))). (Coffman, 29:1-4 (“The conversational system 400 further comprises a plurality of conversational resource subsystems (engines) 407 including, for example, a speech recognition system 408, a speaker recognition system 409, a natural language understanding and natural language parsing system 410 and a text-to-speech synthesis (TTS) system 411.”), Fig. 4.) As with Coffman’s

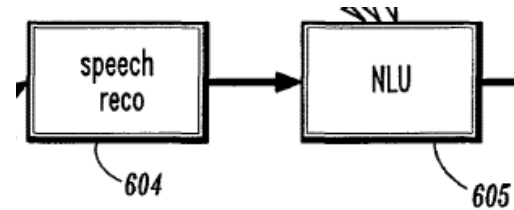


speech recognition engine, a person of ordinary skill in the art would have understood that Coffman's natural language parser, as a software component running on a conventional device such as a desktop computer or PDA, would be **"coupled to the processing device."** (Coffman, 13:1-3 ("The CVM system may be built on top of a conventional OS and APIs 15 and conventional device hardware 16 and located on a server or any client device (PC, PDA, PnC)."), 20:26-29 (describing prioritizing "CPU cycles"), 45:29-32 ("CPU load" and "CPU cycles"), Figs. 1 & 6.) For example, Figure 1 of Coffman depicts conversational sub-systems (engines) 18 coupled to hardware 16:



(Coffman, Fig. 1 (annotations added); *see also id.*, 13:1-3.)

119. A person of ordinary skill in the art would have understood and found it obvious that Coffman “natural language understanding and natural language parsing system 410” would “**pars[e]...the one or more recognized words or phrases**” as part of performing natural language understanding.



Coffman repeatedly describes “NL parsing” and “NLU” and depicts the output of speech recognition (“**recognized words or phrases**”) as input to NLU (see Fig. 6 excerpt at right). (Coffman, 29:1-6, 19:16-19, Fig. 4 (NLU 410), Fig. 6 (“speech reco” 604, NLU 605).) Moreover, the incorporated Kanevsky reference explains that NLU can be implemented using “a combination of classical parsing capabilities with key word spotting and speech recognition.” (Kanevsky, 14:1-7.)

120. Coffman further teaches that parsing the recognized words and phrases as part of natural language understanding is “**to determine a meaning associated with the utterance and a context associated with the request contained in the utterance.**” This is because Coffman uses NLU as part of the process to determine a “context” for the user’s request in order to properly carry out the request. As discussed above for 1**[b][iii]**, Coffman explains that the conversational system includes a “context stack” where the state and history of each active process, such as an application process, is tracked in its own “context.” (Coffman, 29:18-30:3; see

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

also id., 30:4-10.) With respect to 1[c], Coffman further explains that the context stack is searched to determine a context that matches the user's request based on NLU processing. (Coffman, 42:1-5 ("Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active."), 41:2-5 ("Based on the modality (pointer, keyboard, file, speech), the task dispatcher 402 redirects the stream to the appropriate conversational subsystems or conventional subsystem with speech inputs being transcribed/understood. The output of these subsystems is run down the context stack 405 to extract the active query and complete it."), 30:8-10 ("The context of a given discourse is also associated with, e.g., any active grammars, vocabularies and symbolic language which maps the actual query.").) Using an example from Coffman, using NLU, Coffman's system determines that the meaning of the user utterance "I would like to open this" is a request to perform "open" command, which is determined by matching the recognized words and phrases to an "open" command in an application's symbolic language associated with a context. (Coffman, 6:32-7:3.)

121. Under the proposed combination, Coffman's process of identifying an acceptable context by matching the words and phrases also discloses **“the one or more recognized words or phrases are further associated with the determined context in response to the one or more recognized words or phrases satisfying the predetermined confidence level.”** This is because, as previously discussed, Coffman teaches that the words and phrases are used to match a context and its associated formulated request (e.g., in symbolic language or application call/query):

Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active. Until the active command is completed, or until a new command is provided, the selected context is marked active, and pushed to the top of the context stack 405.

(Coffman, 42:1-7; *see also id.*, 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other application is stored to activate if the recognized query is rejected by the user.”).) Therefore, as a result of identifying a context, the words and phrases are associated with that determined context. Moreover, as explained for

1[b][iii], it would have been obvious in view of Kanevsky to perform the claimed “**associating**” under the proposed combination only when speaker identification based on the recognized words and phrases satisfied a predetermined confidence level.

(d) “**formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context;**”
(1[d])

122. 1[d] recites a “**domain agent.**” The ’006 patent provides the following description of “agents”:

Agents are autonomous executables that receive, process and respond to user questions, queries and commands. The agents provide complete, convenient and re-distributable packages or modules of functionality, typically for a specific domain or application. Agents can be complete packages of executable code, scripts, links to information, and other forms of communication data, required to provide a specific package of functionality, usually in a specific domain. In other words, an agent includes everything that is needed to extend the functionality of the invention to a new domain.

(’006, 2:56-65.) Generally speaking, in the software context, an “agent” broadly refers to a program that performs actions on behalf of another entity, such as a user or another software program or computing system. For example, a web browser is sometimes referred to more technically as an “HTTP user agent” (HTTP being short

for Hypertext Transfer Protocol, a communication protocol typically associated with the World Wide Web). Similarly, desktop email software (such as Microsoft Outlook) is sometimes referred to as a “mail user agent” or MUA.

123. A “**domain agent**” in Coffman corresponds to the executable software of a conversationally-aware application (and associated data, such as vocabulary, grammar, language models and NLU symbolic mapping). (Coffman, e.g., 13:26-29 (“The code implementing the applications, API calls and protocol calls includes interpreted and compiled scripts and programs, with library links, conversational logic (as described below) engine call and conversational foundation classes.”).) A conversationally-aware application, consistent with the ’006 patent’s description of “agent,” receives, processes, and responds to user questions, queries and commands. (Coffman, e.g., 13:8-13 (queries transformed into “application calls” and application output returned to user), 37:2-4 (task executes query), 40:9-19 (“execute completed queries” and “output of executed or running processes”).) A conversationally-aware application, consistent with the ’006 patent’s description of “agent,” also usually provides functionality related to a specific domain, such as a particular subject matter (such as email or spreadsheet) or device (such as a PDA or smart phone). (Coffman, 16:35-17:7 (listing “application specific” foundation classes, including “Manage_e-mail,” “Manage_calendar,” and “Manage_addressbook/director”),

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

56:11-14 (e-mail), 58:30-32 (spreadsheet), 56:22-25 (“conversational PDA,” “conversational car PC (PVA, Personal Vehicle Assistant),” and “conversational smart phone”).) Also consistent with the ’006 patent’s description of “agent,” Coffman describes modular development tools for conveniently building complete and re-distributable applications, for example:

Using developmental tools, the developer can easily build his application around the conversational API and CVM. This development environment (referred to herein as Spoken Age) allows programmers to emulate CVM, to debug applications or networked protocols and to rapidly develop conversational user interfaces. Spoken Age includes the CUI and application development for CVM. It also provides the environment for modifying the data files (conversational arguments) of the engines for a given application. In particular this means that at the level of the tools, Spoken Age also includes conventional engine front-ends like SDK Toolkit like the IBM ViaVoice toolkits. This means that toolkits and the algorithms that it provides allows the user to re-build, adapt or extend the data files for a given task.

(Coffman, 58:12-20; *see also id.*, 58:27-28 (“Once an application is developed on a platform and for a specific channel, programmers can rely on Spoken Age to port it to any other platform supporting CVM.”), 7:29-8:9 (“[D]evelopment tools are provided for developer to build, simulate and debug conversational aware

application for CVM. The development tools offer direct implementation of the API calls, protocol calls, application using these API's and protocols, and linking associated libraries, applications exploiting the services and behaviors offered by CVM.... The code implementing these applications can be declarative or procedural. This comprises interpreted and compiled scripts and programs, with library links, conversational logic, engine calls, and conversational foundation classes. Conversational foundation classes are the elementary components or conversational gestures that characterize any dialog, independently of the modality or combination of modalities.”.) Coffman explains that these applications can be distributed for installation by conversational systems. (Coffman, 52:9-11 & 52:18-21 (describing agents to upgrade or install applications).)

124. A conversationally-aware application executable is a “**domain agent**” because, as explained, it provides functionality for a particular area or realm, such as a particular subject matter (such as email or spreadsheet) or a device (such as a PDA or smart phone). (Coffman, 16:35-17:7 (listing “application specific” foundation classes, including “Manage_e-mail,” “Manage_calendar,” and “Manage_addressbook/director”), 56:11-14 (e-mail), 58:30-32 (spreadsheet), 56:22-25 (“conversational PDA,” “conversational car PC (PVA, Personal Vehicle Assistant),” and “conversational smart phone”), 36:29-31 (“More specifically, the

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

present invention provides the capability to have a natural dialog with NLU, NLG and mixed initiative across multiple applications, with multiple domains.")

125. The '006 patent's description of "agents" further states that "[a]gents are autonomous executables." ('006, 2:56.) Coffman further teaches that a conversationally-aware application ("**domain agent**") is an "autonomous executable" because it automatically registers with a conversational system to be able to receive, process and respond to requests, and moreover, operates automatically in response to the system selecting it to handle a request. Coffman explains, for example, that "conversational devices or applications [] register their conversational capabilities." (Coffman, 7:12-14.) Coffman describes "conversational discovery protocols **301** [that] are utilized by spontaneously networked conversational clients **230, 230 a** of the devices to automatically discover local or network conversationally aware systems and dynamically and spontaneously network-connect such conversationally aware systems." (Coffman, 25:18-24; *see also id.*, 23:16-20 ("[T]he distributed conversational protocols **300** also allow the applications and devices to exchange other information such as applets, ActiveX components, and other executable code that allows the devices or associated applications to coordinate a conversation between such devices in, e.g., a master/slave or peer-to-peer conversational network configuration and networks

comprising silent partners”).) And Coffman explains that “[t]he first task to have its dialog completed by producing a non-ambiguous query is executed” (Coffman, 37:2-4), making clear that an application is automatically selected and carries out a request as a result of processing by the conversational system. (*See also id.*, 42:1-7 (“As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active.”).)

126. The autonomous nature of a conversationally-aware application in Coffman’s conversational system is confirmed by International Patent Application Publication No. WO 00/21075 (Maes [EX1007]), which Coffman incorporates by reference and which claims priority to the same provisional patent applications as Coffman. (Coffman, 22:25-29 (“...patent application IBM Docket No. YO999-113P, filed concurrently herewith, entitled ‘System and Method for Providing Network Coordinated Conversational Services,’ which is commonly assigned and incorporated herein by reference...”); Coffman 1:2-3 & Maes, 1:4-5 (both claiming priority to U.S. provisional application serial nos. 60/102,957 and 60/117,595).) Coffman specifically points to Maes for aspects of its conversational computing system. (Coffman, 22:25-29, 26:21-25, 45:11-22, 61:5-6.) Maes describes techniques for conversationally-aware applications to automatically communicate

and coordinate with each other, explaining that “[a] system according to one embodiment of the present invention comprises a plurality of networked servers, devices and/or applications that are made ‘conversationally aware’ of each other by communicating messages using conversational network protocols (or methods) that allow each conversationally aware network device to automatically share conversational resources automatically and in a coordinated and synchronized manner so as to provide a seamless conversational interface through an interface of one of the network devices.” (Maes, 3:4-10.)

127. I note that the preamble further recites “**context-specific domain agents.**” 1[d] does not recite that the “**domain agent**” is “**context-specific.**” But assuming the preamble imposes a limitation on the claim, and that the “**domain agent**” recited in 1[d] must be “**context-specific,**” a conversationally-aware application in Coffman meets this requirement. As explained for 1[b][iii], Coffman describes a “context stack” where the state and history of each active process, such as an application process, is tracked in its own “context.” (Coffman, 29:18-30:3, 30:4-10.) Therefore, a conversationally-aware application is “**context-specific**” by virtue of its association with a context and lack of an association with other contexts.

128. As explained next, the prior art discloses and renders obvious “**formulating, at the parser, the request contained in the utterance in**

accordance with a grammar used by a domain agent associated with the determined context.”

129. Coffman, as discussed, describes using natural language understanding to translate a recognized utterance into a symbolic mapping and query/call to be carried out by an application. (Coffman, e.g., 13:8-13 (“The conversational application API 13 conveys all the information for the CVM 14 to transform queries into application calls and conversely converts output into speech, appropriately sorted before being provided to the user.”), 30:8-10 (“Furthermore, the context of a given discourse is associated with, e.g., any active grammars, vocabularies and symbolic language (attribute value n-uple) which maps the actual query.”).) For example, Coffman teaches that a user’s utterance “I would like to open this” would be processed into “a set of attributes value pairs: Command: Open, Object: Window” and result in an “open” command being performed. (Coffman, 6:32-7:3.) And Coffman, as noted, explains that an application (when registering with a conversational system), provides its active vocabulary, grammar, and language models to the CVM for controlling the application. (Coffman, e.g., 24:31-25:9, 57:22-27, 36:31-32 (“each application will provide the CVM 401 its own parsing and translation arguments”).)

130. A person of ordinary skill in the art would have understood and found

it obvious that, in a system such as Coffman's involving speech recognition using language model(s) and natural language understanding, a grammar associated with an application would be used to formulate the request as a symbolic mapping (such as a set of attribute-value pairs such as "Command: Open, Object: Window") representing the system's understanding and to, in turn, formulate an application query/call in accordance with that understanding. Both the symbolic mapping and the application query/call would separately correspond to the claimed **"request contained in the utterance" "formulat[ed]...in accordance with a grammar used by a domain agent associated with the determined context."**

131. For example, as noted, Coffman describes parsing in connection with natural language processing. (Coffman, 53:25-26 ("User inputs can be represented by their transcriptions or their mappings into a symbolic language after parsing and NLU."), 29:1-4 ("a natural language understanding and natural language parsing system 410").) A person of ordinary skill in the art would have understood and found it obvious in this context that parsing involves using a grammar to identify the parts of speech of the words and phrases in the recognized speech in order to semantically interpret the speech. For example, Gibbon describes "Natural Language Processing" as "[t]he parsing of written language into semantic concepts," where "parsing" is "[t]he labelling of the parts of speech or grammatical elements of

sentences, e.g., subject, predicate, past tense, noun, verb, either by a human analyst or by means of a parser (a programme based on a parsing algorithm).” (Gibbon [EX1008], pp.444, 451.) Coffman’s example of translating the spoken request “I would like to open this” into “a set of attributes value pairs: Command: Open, Object: Window” illustrates parsing of speech in accordance with a grammar. (Coffman, 6:32-7:3.)

132. **Zadrozny** [EX1006] confirms that it would have been obvious to formulate a request “**in accordance with a grammar used by**” a conversationally-aware application (“**a domain agent**”) associated with a context that was determined to be the best match for the user’s utterance. **Zadrozny**, entitled “NL Understanding with a Grammar of Constructions,” is referenced in Kanevsky as an example of a semantic analyzer. (Kanevsky, 14:22-25.) Zadrozny describes a “grammar of constructions” associated with an application (such as a calendar application) for performing natural language understanding to translate a recognized utterance into “a form that can be executed by the application”:

We present an approach to natural language understanding based on a computable grammar of constructions. A construction consists of a set of features of form and a description of meaning in a context. A grammar is a set of constructions. This kind of grammar is the key element of MINCAL, an implemented natural language speech-enabled

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

interface to an on-line calendar system. . . .

With respect to parsing, grammars of constructions can be parsed like “standard” grammars, except that the set of features is richer. Given a string (representing a sentence, a fragment of a discourse or a paragraph), the parser assigns it a construction.... Here the message of an utterance must be understood in the context of an intended action. This is done in two steps. First, the system determines the intended action and its parameters, using domain knowledge (meetings+time+places). Second, once all the parameters have been extracted from the dialog, the system executes the action. To do this, the program uses application-specific knowledge to translate the action and its parameters into a form that can be executed by the application (Xdiary).

(Zadrozny, pp.1-2.) As explained above in the excerpt above, Zadrozny describes a technique using a “grammar of constructions” to formulate an utterance as an intended action and parameters and then translate the action and parameters into a form for the application to execute. Zadrozny illustrates the technique with the example utterance “I want you to arrange a conference in my office at 5.” (Zadrozny, p.3.) Using a grammar of constructions, the utterance is “analyzed as *sent(assert, svoc)*, an assertive sentence consisting of a subject, verb, an object and a complement” and formulated as an action (“schedule”) and series of parameters:

```
[ [ action_name schedule]
  [ event_name
```

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

```
[    a conference]
[ event_time
  [ [ minute 0]
    [ hour
      [ 5 am_or_pm]
    ]
  ]
[ event_place
  [ my office]
```

(Zadrozny, pp.3-4.) As Zadrozny explains, this action and parameters are then translated into a form for the application to execute (i.e., an application call/query).

133. Zadrozny's process is just like Coffman's descriptions of formulating a user's utterance as a set of attribute-value pairs (e.g., "Command: Open, Object: Window") and then converting them into an application call/query. (*See also* Coffman, 30:8-10 ("symbolic language (attribute value n-uple) which maps the actual query"), 13:8-13 (CVM system transforms queries into "application calls").)

134. **Rationale and Motivation to Combine (Coffman, Kanevsky and Kneser with Zadrozny)**. It would have been obvious to a person of ordinary skill in the art to combine Zadrozny's teachings regarding processing speech using a grammar of constructions with the conversational computing system taught by the combination of Coffman, Kanevsky and Kneser. The combination would have predictably resulted in the conversational system in which a user request was formulated in accordance with a grammar used by a conversationally-aware application associated with the context determined to be the best match for the user's

request.

135. Zadrozny is an analogous reference in the same field of speech recognition and processing as the '006 patent. ('006, 1:20-22; Zadrozny, e.g., p.1 (“We present an approach to natural language understanding based on a *computable grammar of constructions*.”) (emphasis in original).) Zadrozny would have also been reasonably pertinent to a number of problems facing the patent inventors, including techniques for understanding natural language speech input and how to determine and formulate actions for an application to undertake in response to the input.

136. A person of ordinary skill in the art would have been motivated to combine. For starters, Kanevsky (which Coffman incorporates by reference) expressly cites Zadrozny as an example of a “semantic analyzer.” (Kanevsky, 14:22-25.) And as noted, authors of Zadrozny are identified as being associated with IBM, the assignee/applicant for both Coffman and Kanevsky. A person of ordinary skill in the art therefore would have naturally turned to Zadrozny for detail about how to implement the systems and techniques described in Coffman and Zadrozny. As with Kanevsky, in fact, a person of ordinary skill in the art would have found it odd to even question Zadrozny’s combinability with Coffman and Kanevsky, or whether Zadrozny can properly be considered to understand Coffman’s conversational

system.

137. A person of ordinary skill in the art would have appreciated that the disclosures of Coffman and Zadrozny confirm their natural fit. As noted, Coffman does not describe the natural language understanding process in detail, mostly describing at a high level that applications provide grammars, vocabularies, language models and symbolic mappings, that a context is matched by “fit[ting] the utterance at the level of the NLU symbolic language” and that “CVM 14 to transform queries into application calls.” Because Kanevsky expressly cites Zadrozny as an example of a semantic analyzer, a person of ordinary skill in the art would have turned to it for its teachings regarding the use of grammars in natural language understanding to translate a recognized utterance into an application call. Moreover, Zadrozny’s process is just like Coffman’s descriptions of formulating a user’s utterance as a set of attribute-value pairs (e.g., “Command: Open, Object: Window”) and then converting them into an application call/query. (*See also* Coffman, 30:8-10 (“symbolic language (attribute value n-uple) which maps the actual query”), 13:8-13 (CVM system transforms queries into “application calls”).)

138. A person of ordinary skill in the art would have had a reasonable expectation of success. Zadrozny, for instance, describes an actual working implementation of a conversational system using a grammar of constructions.

(Zadrozny, e.g., pp.3 (describing MINCAL), 5 (“‘complete’ working system”).) A person of ordinary skill in the art would have found the disclosures of Coffman, Kanevsky, Kneser, and Zadrozny sufficiently detailed to implement the techniques on which I rely, without undue experimentation.

(e) **“processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance; and” (1[e])**

139. As explained above, Coffman teaches that the conversationally-aware application associated with the determined context (**“the domain agent associated with the determined context”**) is invoked to process the user’s request. (Coffman, e.g., 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other application is stored to activate if the recognized query is rejected by the user.”), 13:8-13 & 58:9-11 (“transform queries into application calls”), 40:9-19 (“execute completed queries [and] return output of executed or running processes”), 30:4-10 (“requests made to the corresponding process/task/thread as well as the latest output”).) Under the proposed combination, as explained for 1[d], the conversationally-aware application processes **“the formulated request”** (the symbolic mapping and/or application call/query) because the application carries out processing in accordance with that request. For example, in Zadrozny’s example of

scheduling a conference, Zadrozny teaches that the calendar application would carry out processing to schedule a conference in the system in accordance with the specified action and parameters.

140. A person of ordinary skill in the art would have further understood and found it obvious that the “**processing**” is performed “**to generate a response to the utterance.**” For example, Coffman describes executing completed queries and “return[ing] output of executed or running processes” and presenting them to user using, for example, text-to-speech, as Coffman illustrates in Figure 6:

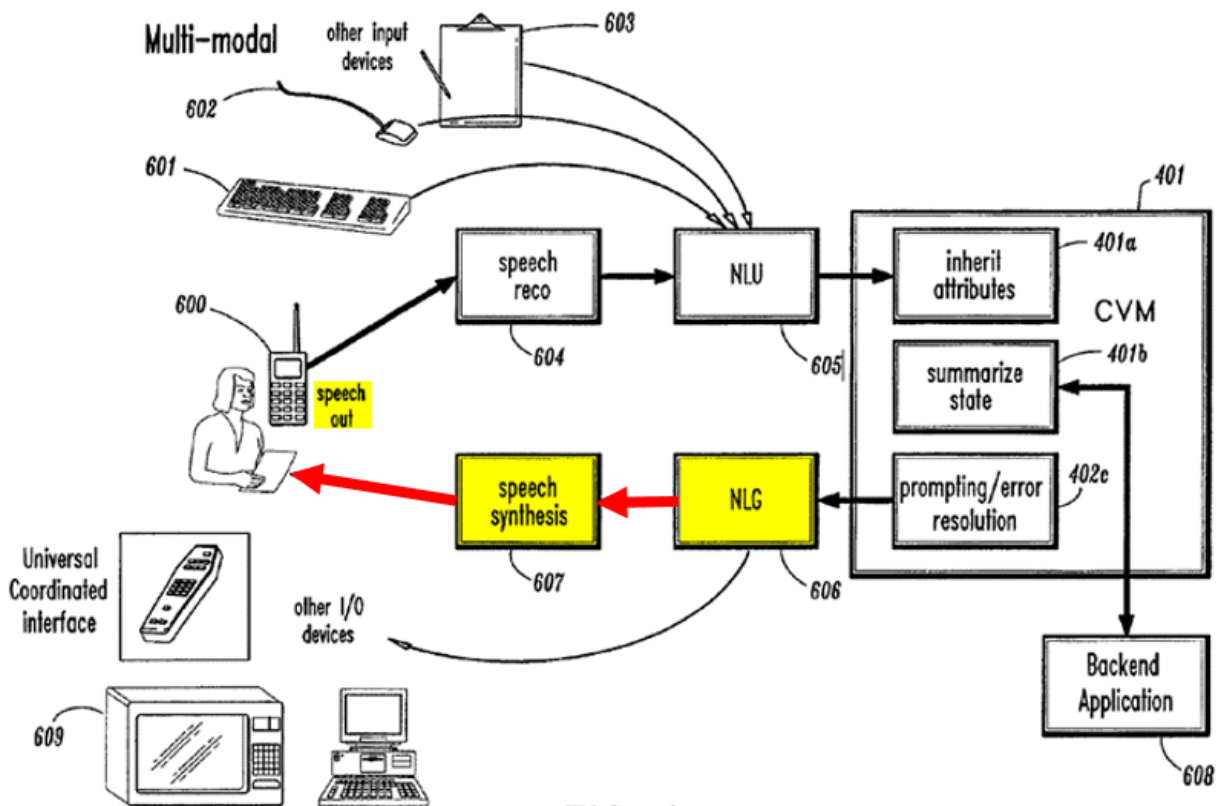


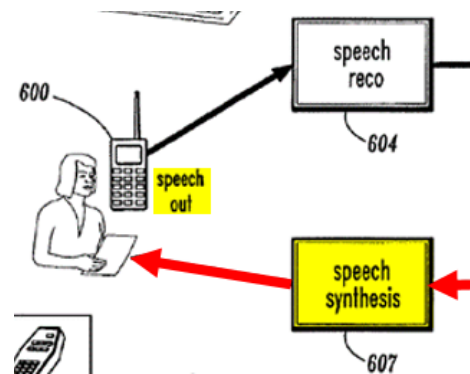
FIG. 6

(Coffman, Fig. 6 (annotations added), 40:9-19; *see also id.*, 13:8-13 & 58:9-11

(CVM converts application “output into speech...provided to the user”), 39:3-5 (“With regard to conversational output dispatches and interface, the CVM 401 produces output to files/streams/resources, display (single or multi-windows, GUI, color, images, movies), audio. Individually, each output stream behaves conventionally.”).)

(f) “presenting the generated response to the utterance via the speech unit.” (1[f])

141. My analysis of 1[e] mostly addresses 1[f]. As explained, Coffman discloses presenting to the user results generated by the conversationally-aware application (“**presenting the generated response to the utterance**”) as a result of processing the symbolic mapping and/or application call/query. (Coffman, Fig. 6, 40:9-19, 13:8-13, 58:9-11, 39:3-5.) Coffman further makes clear that the response can be presented “**via the speech unit,**” explaining that the system can use “text-to-speech” to output the response. (Coffman, Fig. 6 (excerpt at right, showing speech synthesis 607 and “speech out” at telephone 600), 13:8-13 & 58:9-11 (CVM converts application “output into speech...provided to the user”).)



2. Claim 2: “The method according to claim 1, wherein parsing the one or more recognized words or phrases to determine the context associated with the request includes:”

142. As shown below, it would have been obvious under the proposed combination for the step of **“parsing the one or more recognized words or phrases to determine the context associated with the request”** to include the following steps.

(a) “matching the one or more recognized words or phrases to a set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities;” (2[a])

143. A person of ordinary skill in the art would have understood and found it obvious that in Coffman’s system, one or more words in the recognized speech (**“the one or more recognized words or phrases”**) would be **“matched”** to a **“set of keywords”**—words in, for example, an active vocabulary, grammar or NLU symbolic language provided by a conversationally-aware application—to determine the context corresponding to the user’s query. For instance, using Coffman’s example, the word “open” in the request “I would like to open this” matches an “open” command in an application’s symbolic language associated with a context. (Coffman, 6:30-7:3.) More generally, Coffman describes “match[ing]” recognized words and phrases against an NLU symbolic language and “mapping” user input into a symbolic language. (Coffman, 42:1-7; *see also id.*, 53:22-26 (“User inputs

can be represented by their transcriptions or their mappings into a symbolic language after parsing and NLU.”.) This is consistent with the incorporated Kanevsky reference, which as noted, teaches performing natural language understanding “preferably” with a “combination of classical parsing capabilities with key word spotting.” (Kanevsky, 14:1-4.)

144. With respect to the “**set of keywords**” being “**associated with the dynamic set of prior probabilities or fuzzy possibilities**” under the proposed combination, the keywords would match words recognized as a result of applying adaptive language model probabilities (“**the dynamic set of prior probabilities**”)—for example, the keyword “open” in a symbolic language matches the recognized word “open.” As a result, the matched keywords would also be “**associated with the dynamic set of prior probabilities or fuzzy possibilities.**”

(b) “**scoring multiple possible contexts for the one or more recognized words or phrases matched to the set of keywords using a profile associated with the determined user identity, data content associated with the domain agent, or a context stack that includes one or more recent contexts; and**” (2[b])

145. Coffman discloses determining acceptability of “**multiple possible contexts for the one or more recognized words or phrases matched to the set of keywords using...a context stack that includes one or more recent contexts.**”

Coffman describes searching a context stack to find a context that is “an acceptable

match” by “fit[ting] the utterance at the level of the NLU symbolic language.” (Coffman, 42:1-5 (“That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active.”).) If the match with a context in the context stack is “too improbable,” then the next context in the context stack is considered for matching, thereby “**using...a context stack.**” (Coffman, 40:24-27 (“Rejection or likelihood of a new query is managed by each activated task based on the conversational arguments. Queries that are rejected or too improbable cause the dialog controller 404 to peruse down the context stack 405 to look for the next available context.”).) Coffman also makes clear that its context stack “**includes one or more recent contexts.**” (Coffman, 29:20-30 (context stack “accumulates the context...of each active process with an activated discourse....[context] keeps track of the past history of the discourse [and] its current state]....”), 30:13-16 (“The dialog controller 404 manages the context stack 405 by creating a new stack entry in the context stack 405 for each new task/process/thread that is spawned either local or remotely....”), 41:8-9 (“Each new (active) task/process/thread creates a new stack entry in the context stack 405, with or without activated discourse.”).) And

therefore, by searching the context stack for an acceptable context, Coffman discloses **“using...a context stack that includes one or more recent contexts.”**

146. A person of ordinary skill in the art would have further understood and found it obvious that Coffman’s technique of determining whether a context was an “acceptable match” or “too improbable” would involve **“scoring”** the context, such as determining the likelihood or goodness of the match in order to compare it against some level of acceptability. Like the concept of a confidence level, the concept of “scoring” has been well known in the field of speech recognition and processing since at least the 1970s. The obviousness of **“scoring”** a context is confirmed by Kanevsky, which expressly describes multiple different analogous situations (speaker recognition and semantic understanding) in which a score is generated (and compared against a predetermined threshold value) to determine further action to take. For example, Kanevsky describes a process for recognizing the speaker that involves both scoring a user’s spoken input against a stored answer and scoring a user’s voice sample against a stored user model. (Kanevsky, 6:34-42 (“The result of the semantic analyzer **40** is sent to a score estimator **44** via link 46 where a partial score associated with the answer received from the user is generated.”), 7:12-20 (“The voice sample is processed against the user model and another partial score is generated by the score estimator **44**.”).) A combined score is then compared “to a

predetermined threshold value and if the score is one of substantially equivalent to and above the threshold value,” the speaker is permitted access to the system. (Kanevsky, 3:40-48; *see also id.*, 7:12-20 (“Based on a comparison of a combination of the partial scores (from the question/answer phase and the background speaker verification provided by module **52**) versus a predetermined threshold value, the central server **22** decides whether or not to permit access to the service/facility to the user **12**.”).)

147. A person of ordinary skill in the art would have found it obvious to apply these scoring/matching techniques to the context matching described in Coffman. I previously discussed the rationale and motivation to combine Coffman and Kanevsky, which applies here as well. And here, specifically, as noted Coffman describes rejecting a context where a match is “too improbable,” which confirms that there is some level of “probability” required. Kanevsky confirms that using a scoring technique would have been a straightforward and obvious choice—as the use of known “scoring” techniques to improve a similar method in the same way, as the application of known “scoring” techniques to improve methods for determining a context matching a user’s query, with predictable results.

- (c) **“selecting one of the scored multiple possible contexts having a highest score that satisfies the predetermined confidence level to be the determined context.” (2[c])**

148. My analysis of 2[b] largely already addressed this limitation. As explained, a person of ordinary skill in the art would have understood and found it obvious in view of Coffman and Kanevsky that the process of searching a context stack to select an acceptable context would involve scoring each searched context at least until a context having a score high enough to satisfy a predetermined threshold value was identified. (Coffman, e.g., 40:24-27, 42:1-5; *see also id.*, 41:8-11.) The context having such a high score would then be **“selected...to be the determined context,”** and the application associated with that context would process the user’s request.

3. **Claim 3: “The method according to claim 2, wherein parsing the one or more recognized words or phrases to determine the context associated with the request further includes:”**

149. As shown below, it would have been obvious under the proposed combination for the step of **“parsing the one or more recognized words or phrases to determine the context associated with the request”** to further include the following steps.

- (a) **“requesting the user to verify the one or more recognized words or phrases in response to determining that none of the scored multiple possible contexts satisfy the predetermined confidence level; and” (3[a])**

150. As explained for claim 2 above, it would have been obvious in view of Coffman and Kanevsky to determine whether a context had a score satisfying a predetermined confidence level. It would have been further obvious to **“request[] the user to verify the one or more recognized words or phrases in response to determining that none of the scored multiple possible contexts satisfy the predetermined confidence level”** as recited by 3[a]. This is because Coffman explains that a user’s command/query that is deemed incomplete or ambiguous—i.e., context scores do not satisfy a predetermined confidence level—**“will generate a request (a prompt) for the missing or ambiguous information and update the context (requested missing fields).”** (Coffman, 41:16-21; *see also id.*, 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed...”), 40:9-19 (“provide feedback to the user for incomplete or ambiguous queries/command”).) In other words, Coffman teaches that the conversational system will prompt the user to verify their input as the system understood it (**“the one or more recognized words or phrases”**) if the system determines that input was incomplete or ambiguous. Such a correction, completion or disambiguation of

the user's dialogue is a "verification" of the recognized words or phrases.

(b) "determining the context based on information that the user provides in one or more subsequent natural language speech utterances." (3[b])

151. As explained above for 3[a], Coffman describes prompting a user to provide missing or ambiguous information, which results in updating an associated context. Coffman further teaches that non-ambiguous queries are executed by the system. (Coffman, e.g., 37:2-4 ("The first task to have its dialog completed by producing a non-ambiguous query is executed").) A person of ordinary skill in the art would have therefore understood and found it obvious for the system of Coffman to **"determin[e] the context based on [the additional] information the user provides."** Moreover, because Coffman describes spoken prompts and speech input, a person of ordinary skill in the art would have understood and found it obvious for the user to provide the additional information via **"in one or more subsequent natural language speech utterances."** (Coffman, 14:27 ("spoken prompt"), 5:18 ("speech input").)

C. Ground 2: Claim 4 Is Obvious Over Ground 1 Prior Art (Coffman, Kanevsky, Kneser and Zadrozny) in Further View of Maes '101 and Franceschi.

1. Claim 4: "A method for processing natural language speech utterances with context-specific domain agents, comprising:"

152. Like claim 1, the preamble of claim 4 recites **"[a] method for**

processing natural language speech utterances with context-specific domain agents, comprising.” As shown below, the claimed method is disclosed and rendered obvious by Ground 1 prior art in further view of Maes ’101 and Franceschi.

- (a) **“receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;” (4[a])**

153. 4[a] is verbatim identical to 1[a]. Therefore, for the reasons explained above in my analysis of 1[a], 4[a] is disclosed by and obvious over Ground 1 prior art.

- (b) **“recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables, wherein recognizing the one or more words or phrases contained in the utterance includes:” (4[b])**

154. 4[b] is verbatim identical to 1[b]. Therefore, for the reasons explained above in my analysis of 1[b], 4[b] is disclosed by and obvious over Ground 1 prior art.

155. As explained for the following limitations, the prior art also discloses and renders obvious that **“recognizing the one or more words or phrases contained in the utterance includes”** the following steps.

- (i) **“requesting a user that spoke the utterance to spell the one or more words or phrases in response to the one or more recognized words or phrases failing to satisfy a predetermined confidence level; and” (4[b][i])**

156. Coffman explains that a user’s command/query that is deemed

incomplete or ambiguous “will generate a request (a prompt) for the missing or ambiguous information and update the context (requested missing fields).” (Coffman, 41:16-21; *see also id.*, 40:9-19 (“provide feedback to the user for incomplete or ambiguous queries/command”), 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed...”).) In other words, Coffman teaches that the conversational system will “**request[] a user that spoke the utterance**” to verify their input as the system understood it if the system determines that input was incomplete or ambiguous.

157. Coffman does not expressly describe prompting the user to “**spell the one or more words or phrases in response to the one or more recognized words or phrases failing to satisfy a predetermined confidence level.**” But as explained below, applying Coffman’s techniques for prompting a user regarding ambiguous and/or incomplete information to spelling would have been obvious to a person of ordinary skill in the art.

158. For starters, the use of a predetermined confidence level to accept or reject recognition results was a routine and conventional feature of speech recognition systems, and would have formed part of the background knowledge of a person of ordinary skill in the art. (*See, e.g.*, Gibbon [EX1008], pp.391 (defining “confidence measure” as “[a] value computed, for instance, by an automatic speech

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

recogniser indicating the degree of belief that the word, phrase or sentence to which the confidence measure was assigned has been recognised correctly”), 465 (defining “rejection” as “[t]he decision of an automatic speech recognition (ASR) device that the input (or part of the input) cannot be mapped onto one or more words in the vocabulary with sufficient confidence. This results in failure to recognise the input.”).) I personally implemented confidence levels in my speech recognition and processing work starting in 1979. (See, e.g., C. Schmandt, “The Intelligent Ear: A Graphical Interface to Digital Audio” (1981) [EX1017], pp.394-395 (describing “degree of certainty for recognition of each word”).)

159. Moreover, Coffman incorporates by reference International Patent Application Publication No. WO 00/21075 (**Maes** [EX1007]) (Coffman, 22:25-29 (“...patent application IBM Docket No. YO999- 113P, filed concurrently herewith, entitled ‘System and Method for Providing Network Coordinated Conversational Services,’ which is commonly assigned and incorporated herein by reference...”).) I am informed by counsel for Petitioner that, because Coffman incorporates Maes by reference, Maes is effectively part of Coffman as if it were explicitly contained in it. Maes acknowledges “efforts...to develop appropriate confidence measures for [] recognized speech.” (Maes, 17:23-26.) Maes further describes, in the context of determining whether a client should contact a server for additional speech

recognition processing, rejecting speech recognition results “based on confidence measures” and determining whether speech recognition results meet “a given threshold”:

After either local processing (step **202**) or remote processing (step **205**), a determination is made (via the local dialog manager **103** or server dialog manager **108**) as to whether the results of the processing are acceptable (steps **203** and **206**). If it is determined that the processing results are not acceptable (negative determination in step **203** or step **206**), the local client or remote server will automatically forward the features or waveform to a server (via IP, LAN, Bluetooth, IR, RF or connect via phone or IP phone) that is capable of performing such processing (step **204** or step **207**). For instance, the rejection of results for speech recognition and NLU may occur if such results are unknown or mis-recognized or ambiguous (or based on confidence measures associated with the dialog manager of each resources (local or server)). Preferably, the automatic connection from the local or remote system to a server system can be based on the level of the acoustic and LM (language model) scores returned by the local speech decoder using, for example, the techniques taught in U.S. Pat. No. 5,937,383 to Ittycheriah et al., entitled “Apparatus and Methods For Speech Recognition Including Individual or Speaker Class Dependent Decoding History Caches For Fast Word Acceptance or Rejection,” which is commonly assigned and incorporated herein by reference (e.g., the remote server is contacted when these scores are determined to fall below a given

threshold). It is to be understood that any suitable measure or approach to estimate the confidence or the recognized utterance or query (during or after the dialog) can be used to determine if the results obtained by a conversational system are acceptable or not (in which case another system is considered).

(Maes, 16:20-17:7; *see also id.*, 17:23-18:16.) As shown in the excerpt above, Maes describes rejecting speech recognition results when a confidence measure is below a given threshold.

160. But rather than outright rejecting the user's utterance when speech recognition results fall below a given confidence threshold and, for example, asking the user to repeat themselves, it would have been obvious for the system according to the proposed combination to prompt the user to spell one or more words. Requesting the user to provide a spelling would, for example, advantageously further the goal of an adaptive language model to be able to better recognize a user's utterances. Voice spelling would also provide a user a convenient way to provide input in situations where other forms of input were limited or less desirable, such as in a vehicle or a device with limited input capabilities, or when the user is less-abled or disabled. The obviousness of prompting a user to spell one or more words is confirmed by another patent to Stephane Maes, U.S. Patent No. 6,073,101 (**Maes '101** [EX1010]), which explains that "unrecognized utterances can be defined by the

user during enrollment or later enrollment correction by prompting the user to spell the word whenever speech recognition is unsuccessful.” (Maes ’101, 6:52-55.)

Maes ’101 discloses that, by spelling words, the system’s performance can be improved and adapted to the user, by adding the spelled word to the speech recognizer’s vocabulary, possibly associated only with particular users. (Maes ’101, 6:55-66.) The step of **“requesting a user that spoke the utterance to spell the one or more words or phrases in response to the one or more recognized words or phrases failing to satisfy a predetermined confidence level,”** therefore would have been obvious in further view of Maes ’101.

161. **“Phonetic alphabet spelling” Issue:** While 4[b][i] simply recites **“requesting a user...to spell,”** 4[b][iii] refers to **“a phonetic alphabet spelling”** provided by the user. It would have been obvious for the user to provide a **“phonetic alphabet spelling”** of the one or more words or phrases. This is confirmed by U.S. Patent No. 6,321,196 (**Franceschi** [EX1011]). Franceschi, entitled “Phonetic Spelling for Speech Recognition,” explains the motivations and benefits of phonetic alphabet spelling, and further, that such functionality was routine and conventional, noting that commercially-available speech recognition systems already included it:

[I]t frequently occurs that the best approach for an unrecognized, difficult, or new word is for the speaker to spell it....

However, many of the letters sound very similar and may be confused

for each other, even for human listeners. It is therefore known in the art to spell a word phonetically which is meant to indicate using a commonly understood word for each letter in the word to be spelled. For example, one may phonetically spell the word “key” by saying “kilo echo yankee.” One may also use a phonetic alphabet when selecting items from a list by using a word from the phonetic alphabet rather than saying the corresponding letter name.... Various such alphabets have been developed and used over the years with human listeners, primarily by the military, for clearly communicating over sometimes noisy or unreliable radio or telephone links. The developers of speech recognition devices and programs have likewise incorporated a phonetic spelling feature in their products for word spelling.

(Franceschi, 1:38-52, 2:8-13.) Franceschi purports to describe improved techniques for enabling a user to provide phonetic alphabet spelling in speech recognition systems. (Franceschi, e.g., 2:46-50.)

162. **Rationale and Motivation to Combine (Ground 1 Prior Art with Maes ’101 and Franceschi)**. It would have been obvious to a person of ordinary skill in the art to further combine with Maes ’101 and Franceschi. The combination would have predictably resulted in Coffman’s conversational system with the ability to prompt the user to provide a phonetic alphabet spelling if speech recognition results failed to meet a certain confidence level.

163. Maes ’101 and Franceschi are both analogous references in the same

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

field of speech recognition and processing as the '006 patent. ('006, 1:20-22; Maes '101, 1:11-18 (“The present invention generally relates to speech and speaker recognition and, more particularly, to text independent speaker recognition, which may thus include consistency checks of the recognition process, particularly suitable for continuous security and access control and which is transparent to one or more users and provides enhanced resolution of ambiguities between recognizable commands.”); Franceschi, 1:6-8 (“The invention generally relates to speech recognition devices and methods and in particular to the phonetic spelling of words.”).) Maes '101 and Franceschi would have also been reasonably pertinent to a number of problems facing the patent inventors, including techniques for receiving speech input from a user and improving the reliability of speech recognition results by resolving ambiguity.

164. A person of ordinary skill in the art would have been motivated to combine. To start, as noted, Coffman and Maes '101 (as well as the Maes reference incorporated into Coffman) all name a common co-inventor, Stephane Herman Maes of Danbury, Connecticut, and identify the same assignee/applicant, IBM. (Coffman, face page; Maes '101, face page; Maes, face page.) And Franceschi likewise identifies IBM as the assignee. (Franceschi, face page.) Given the connections between these references, therefore, a person of ordinary skill in the art would have

found it odd to even question Maes '101's and Franceschi's combinability with Coffman, or whether they can properly be considered together to understand Coffman's conversational system.

165. Moreover, as noted above, Coffman explains that a user's command/query that is deemed incomplete or ambiguous—i.e., context scores do not satisfy a predetermined confidence level—“will generate a request (a prompt) for the missing or ambiguous information and update the context (requested missing fields).” (Coffman, 41:16-21; *see also id.*, 37:2-4 (“The first task to have its dialog completed by producing a non-ambiguous query is executed...”), 40:9-19 (“provide feedback to the user for incomplete or ambiguous queries/command”).) In other words, Coffman teaches that the conversational system will prompt the user to verify their input as the system understood it if the system determines that input was incomplete or ambiguous. Franceschi teaches that “it frequently occurs that the best approach for an unrecognized, difficult, or new word is for the speaker to spell it.” (Franceschi, 1:38-40.) A person of ordinary skill in the art would have therefore appreciated that this “frequently best approach”—prompting a user to spell a word—would help resolve incomplete or ambiguous input, and that the ability to provide a phonetic alphabet spelling as taught by Franceschi would help avoid ambiguity when processing the user's spelling input. For example, in the English language, because

the “e set” of letters—b, c, d, e, g, etc.—all sound similar, it can be challenging to reliably differentiate one from another. A phonetic alphabet, whether the “alpha, bravo, charlie” military style or the more free-form style Franceschi proposes, is more reliable for differentiating letters, resulting in an improved user experience.

166. A person of ordinary skill in the art would have had a reasonable expectation of success. Incorporating phonetic alphabet spelling into Coffman’s conversational system would have involved conventional techniques and technologies. For example, as noted above, Franceschi notes that existing speech recognition and processing systems already implemented phonetic alphabet spelling functionality. (Franceschi, 2:8-13 (“The developers of speech recognition devices and programs have likewise incorporated a phonetic spelling feature in their products for word spelling.”).) A person of ordinary skill in the art would have found the disclosures of Coffman, Kanevsky, Kneser, Zadrozny, Maes ’101, and Franceschi sufficiently detailed to implement the techniques on which I rely, without undue experimentation.

- (ii) **“updating the information in the one or more dictionary and phrase tables based on a phonetic alphabet spelling associated with the one or more words or phrases, wherein the user provides the phonetic alphabet spelling in one or more subsequent natural language speech utterances;” (4[b][ii])**

167. As explained for 1[b], **“the one or more dictionary and phrase tables”** under the proposed combination of Coffman and Kanevsky correspond to one or more trigram language models provided by conversationally-aware applications to a conversational system and that the system uses for speech recognition.

168. It would have been obvious to a person of ordinary skill in the art to update information in the one or more trigram language models (**“the one or more dictionary and phrase tables”**) **“based on a phonetic alphabet spelling associated with the one or more words or phrases.”** As noted above for 4[b][i], Maes ’101 discloses that, by spelling words, the system’s performance can be improved and adapted to the user, by adding the spelled word to the speech recognizer’s vocabulary, possibly associated only with particular users. (Maes ’101, 6:55-66.) Adding words to a speech recognizer’s vocabulary would also further Kneser’s goal of improving a speech recognizer’s language model, “[s]ince the quality of the adaptation of the language model values has a considerable influence

on the reliability of the speech recognition.” (Kneser, 1:25-29.) It would have been obvious, therefore, to update trigram language model(s) used for speech recognition to incorporate information about the additional vocabulary provided by the user (“**updat[e] the information in the one or more dictionary and phrase tables**”).

169. With respect to “**the user provid[ing] the phonetic alphabet spelling in one or more subsequent natural language speech utterances,**” under the proposed combination and as taught by Maes ’101, the spelling would take place by voice after the user is prompted to provide it (“**in one or more subsequent natural language speech utterances**”). (Maes ’101, 6:52-66.)

- (c) “**parsing, at a parser coupled to the processing device, the phonetic alphabet spelling associated with the one or more words or phrases to determine a meaning associated with the utterance and a context associated with the request contained in the utterance;**” (4[c])

170. 4[c] is substantially similar to 1[f], and my analysis of 1[f] applies here. 4[c] further recites that the parsing step includes “**parsing...the phonetic alphabet spelling associated with the one or more words or phrases.**” A person of ordinary skill in the art would have understood and found it obvious that the user’s phonetic alphabet spelling would be parsed under the proposed combination with Coffman. Parsing the spelling would have enabled the Coffman’s system to determine the spelled word so that the system could include the word as part of determining the

meaning of the user's utterance and an associated context. As a basic example, if the user had provided a phonetic alphabet spelling of "Joe" as part of their utterance "email from Joe," it would have been obvious for the system to parse the phonetic alphabet spelling "Juliet Oscar Echo" (or e.g., "J as in John, O as in Ocean, M as in Mary") to determine that the user was spelling the word "Joe."

- (d) **"formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context;"**
(4[d])

171. 4[d] is verbatim identical to 1[d]. Therefore, for the reasons explained above in my analysis of 1[d], 4[d] is disclosed by and obvious over the proposed combination.

- (e) **"processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance; and"** (4[e])

172. 4[e] is verbatim identical to 1[e]. Therefore, for the reasons explained above in my analysis of 1[e], 4[e] is disclosed by and obvious over the proposed combination.

- (f) **"presenting the generated response to the utterance via the speech unit."** (4[f])

173. 4[f] is verbatim identical to 1[f]. Therefore, for the reasons explained above in my analysis of 1[f], 4[f] is disclosed by and obvious over the proposed

combination.

D. Ground 3: Claim 5 Is Obvious Over Ground 1 Prior Art (Coffman, Kanevsky, Kneser and Zadrozny) in Further View of Chai.

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

174. Like claim 1, the preamble of claim 5 recites “[a] **method for processing natural language speech utterances with context-specific domain agents, comprising.**” As shown below, the claimed method is disclosed and rendered obvious by Ground 1 prior art in further view of Chai.

(a) “receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request;” (5[a])

175. 5[a] is verbatim identical to 1[a]. Therefore, for the reasons explained above in my analysis of 1[a], 5[a] is disclosed by and obvious over Ground 1 prior art.

(b) “recognizing, at a speech recognition engine coupled to the processing device, one or more words or phrases contained in the utterance using information in one or more dictionary and phrase tables;” (5[b])

176. 5[b] is verbatim identical to 1[b], except that 5[b] does not further recite “wherein recognizing the one or more words or phrases contained in the utterance includes.” Therefore, for the reasons explained above in my analysis of 1[b], 5[b] is disclosed by and obvious over Ground 1 prior art.

- (c) **“parsing, at a parser coupled to the processing device, information relating to the utterance to determine a meaning associated with the utterance and a context associated with the request contained in the utterance, wherein the parsed information includes the one or more recognized words or phrases;” (5[c])**

177. For purposes of my analysis, 5[c] is substantially identical to 1[c]. 1[c] recites parsing “the one or more recognized words or phrases,” while 5[c] recites parsing “information relating to the utterance...wherein the parsed information includes the one or more recognized words and phrases.” This is not a meaningful difference because the **“information relating to the utterance”** recited in 5[c] includes “the one or more recognized words or phrases.” Therefore, for the reasons explained above in my analysis of 1[c], 5[c] is disclosed by and obvious over Ground 1 prior art.

- (d) **“formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context, wherein formulating the request in accordance with the grammar used by the domain agent includes:” (5[d])**

178. 5[d], like 1[d], recites **“formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context.”** This portion of 5[d] is therefore disclosed by and obvious over the proposed combination.

179. 5[d] further recites “**wherein formulating the request in accordance with the grammar used by the domain agent includes**” the following steps. As explained below, these steps would have been obvious in further view of Chai.

- (i) **“determining one or more required values and one or more optional values associated with formulating the request in the grammar used by the domain agent;” (5[d][i])**

180. As discussed above for 1[d], a person of ordinary skill in the art would have understood that, in a system such as Coffman’s involving speech recognition using language model(s) and natural language understanding, a grammar associated with an application would be used to understand the recognized speech so that a symbolic mapping and application query/call could be formulated in accordance with the understood meaning. For example, an NL parser in Coffman would use a grammar convert the statement, “I would like to open this” into “a set of attribute value pairs: Command: open, Object: Windows or task selected by the last mouse click. (Coffman, 6:32-7:3.) This understanding is confirmed by Zadrozny, which as noted, discloses a “grammar of constructions” that is likewise used to determine an intended action and its parameters for an application to execute from a user’s natural language input:

With respect to parsing, grammars of constructions can be parsed like “standard” grammars, except that the set of features is richer. Given a

string (representing a sentence, a fragment of a discourse or a paragraph), the parser assigns it a construction.... Here the message of an utterance must be understood in the context of an intended action. This is done in two steps. First, the system determines the intended action and its parameters, using domain knowledge (meetings+time+places). Second, once all the parameters have been extracted from the dialog, the system executes the action. To do this, the program uses application-specific knowledge to translate the action and its parameters into a form that can be executed by the application (Xdiary).

(Zadrozny, pp.1-2; *see also id.*, p.1 (“We present an approach to natural language understanding based on a computable grammar of constructions. A construction consists of a set of features of form and a description of meaning in a context. A grammar is a set of constructions. This kind of grammar is the key element of MINCAL, an implemented natural language speech-enabled interface to an on-line calendar system.”).) Therefore, under the proposed combination, where the application would be a conversationally-aware application of Coffman (“**the domain agent**”), the determined action and parameters for criteria using Zadrozny’s “grammar of constructions” would correspond to “**the request in the grammar used by the domain agent.**”

181. For example, Zadrozny describes using a grammar of constructions to

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

translate the statement “I want you to arrange a conference in my office at 5” into an action (“action_name = schedule”) with various criteria and parameters (e.g., “event_name [a conference]”; “event_place [my office]”):

For example, the sentence *I want you to arrange a conference in my office at 5* is analyzed as *sent(assert, svoc)*, an assertive sentence consisting of a subject, a verb, an object and a complement.... The result of parsing is then interpreted by the domain interpreter to produce:

```
***Slots:
[ [ action_name schedule]
  [ event_name
    [ a conference]
  [ event_time
    [ [ minute 0]
      [ hour
        [ 5 am_or_pm]
    [ event_place
      [ my office]
```

(Zadrozny, pp.3-4.)

182. As shown in the excerpt above, Zadrozny identifies the various fields for actions and criteria/parameters as “slots” (“***Slots:”).⁷ Zadrozny’s technique

⁷ The “slots” technique, as in Zadrozny, was well known in the art and usually called “slot and filler” in natural language processing literature. I first used “slot and filler” in my work in 1979. (See, e.g., C. Schmandt et al., “A Robust Parser and Dialog Generator for a Conversational Office System” (1986) [EX1019], p.003 (section 3 describing “slot-and-filler approach”).)

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

of mapping the words and phrases of the parsed input to various slots related to a grammar of constructions is consistent with Coffman's descriptions of determining a context and query by filling in symbolic fields for a natural language query. Coffman describes for example, "[f]or NLU queries, the symbolic fields are filled for each stream," "fit[ting] the utterance at the level of the NLU symbolic language," and "symbolic language which maps the actual query." (Coffman, 38:15-16, 42:4-5, 30:8-10.)

183. Zadrozny does not appear to expressly disclose that an action based on a grammar of constructions ("**the request in the grammar used by the domain agent**") would have "**one or more required values and one or more optional values**" but this would have been obvious over U.S. Patent No. 6,829,603 (**Chai** [EX1012]), which as I noted earlier names Wlodek Zadrozny as a co-inventor.

184. Like Coffman and Zadrozny, **Chai** similarly describes techniques for processing a natural language request from a user to determine an application action to execute. Like the symbolic field filling described in Coffman and the slot filling disclosed in Zadrozny, Chai similarly describes filling "slots" in "action templates" based on natural language input in order to determine the application action to execute. Chai explains:

One or more presentation manager modules present information from

the computer system to one or more users over one or more network interfaces and accept queries from one or more of the users using one or more known input/output modalities (e.g. Speech, typed in text, pointing devices, etc.). A natural language parser parses one or more natural language phrases received over one or more of the network interfaces by one or more of the presentation managers into one or more logical forms (parsed user input), each logical form having a grammatical and structural organization. A dialog manager module maintains and directs interactive sessions between each of the users and the computer system. The dialog manager receives logical forms from one or more of the presentation managers and sends these to a taxonomical mapping process which matches the items of interest to the user against the content organization in the content database to match business categories and sends modified logical forms back to the dialog manager. In a preferred embodiment, these modified logical forms are matched against a set of predefined action templates, the best matching template is selected, corresponding action requests are sent to an action manager, action results are received from the action manager, and presentation requests are sent to one or more of the presentation managers for presenting the system response to the user. The action manager module receives action requests from the dialog manager, executes the action (e.g. Retrieving the latest price of a stock or all blue pants with cuffs), and sends the action results to the dialog manager.

(Chai, 4:51-5:11.)

185. For example, in an online shopping example, Chai describes processing

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

two user requests, “I want to buy one 30 inch wide White Color Cooktop with Electric PowerSource and with Electric:Radiant Glass CookTop Surface and of KenMore company” followed by “I want to buy one Black CookTop of previous type.” (Chai, 12:28-32.) Chai explains that the user’s input is used to fill various slots of matched action templates, with some slots being “**required**”—for example, a value for the “item name” slot is required:

Before taking discourse context into consideration, the following slots **807** in matching templates **806** with “buy” transaction are filled.

First Template (Used for Pants and Shirts):

TRANSACTION: buy

TEMPLATE SLOTS:

Item name: CookTop

sku:

quantity: 1

size:

color: Black

company:

REQUIRED SLOTS: item name, quantity, company

(Chai, 12:32-44; *see also id.*, 12:45-56 (second template), 12:21-22 (“choosing the template with the most number of required slots filled”).) On the other hand, referring to the example action template above, because a value for the “color” slot is not identified as being required, a person of ordinary skill in the art would have understood it to be optional. Chai therefore discloses both “**one or more required values**” (e.g., the value for the “item name” slot in Chai’s shopping example) and “**one or more optional values**” (e.g., the value for the “color” slot in Chai’s

example). I implemented these techniques that Chai describes in my own work, such as my Put That There system, starting in 1979. (*See, e.g.*, C. Schmandt et al., “The Intelligent Voice-Interactive Interface” (1982), pp.364-365 (describing matching input words against command templates); C. Schmandt et al., “A Robust Parser and Dialog Generator for a Conversational Office System” (1986) [EX1019], pp.003 (section 3 describing “slot-and-filler approach”), 007 (“although there are slots for both name and time, for example, some commands may require only one or the other of them”).)

186. **Rationale and Motivation to Combine (Ground 1 Prior Art with Chai)**. It would have been obvious to a person of ordinary skill in the art to combine Chai’s teachings regarding required and optional slot values for action templates to Zadrozny’s technique for formulating a request using a grammar of constructions. Under the proposed combination, this would have predictably resulted in Coffman system’s step of determining the call/query for the conversationally-aware application (“**the domain agent**”) to include the step of determining required and optional values for the application call/query (“**determining one or more required values and one or more optional values associated with formulating the request in the grammar used by the domain agent**”).

187. Chai is an analogous reference in the same field of speech recognition

and processing, including using applications with natural language input, as the '006 patent. ('006, 1:20-22; Chai, 1:8-10 (“This invention relates to database searching and queries, and more particularly to natural language based interactive database searching and queries in network environment.”).) Chai would have also been reasonably pertinent to a number of problems facing the patent inventors, including how to formulate application queries using natural language.

188. A person of ordinary skill in the art would have been motivated to combine. For starters, in yet another link between prior art references, Chai lists Wlodek Zadrozny, lead author of Zadrozny [EX1006], as a co-inventor and also identifies IBM as the assignee. (Chai, face page.) A person of ordinary skill in the art would have found it odd to even question their combinability, or whether Chai can properly be considered to understand Zadrozny’s techniques for formulating application queries or Coffman’s conversational system.

189. Moreover, while Chai focuses on examples related to online shopping, Chai also makes clear that its techniques are applicable in other situations similar to those discussed in Coffman, such as retrieving stock quotes. (Chai, 5:10 (“Retrieving the latest price of a stock”), 8:16-65 (describing “stock trading application”); Coffman, 42:50-52 (“You can now drive home and while driving quietly listen to your e-mail or check the news or stock quotes.”).) A person of

ordinary skill in the art would have therefore appreciated the broad applicability of Chai's techniques to formulating queries for conversationally-aware applications in Coffman's system. A person of ordinary skill in the art would have also appreciated that the ability to formulate queries with required and optional values would have benefitted Coffman's system by providing a more flexible system for responding to user utterances. For example, the ability to provide additional, but not strictly necessary, information as part of a request would enable a user to more quickly and directly locate desired information rather than receive less precise information or engage in a lengthier colloquy with the conversational system to refine the response to the user's request. Optional parameters also simply reflect the nature of many types of information queries. For example, a shorter query might imply default values for unspecified parameters—for instance, a query about the weather, "what's the forecast," could be interpreted to imply "tomorrow" even though the desired day is not expressly stated.

190. A person of ordinary skill in the art would have had a reasonable expectation of success. Application queries involving required and optional values were well known and conventional in computer programming. Implementing Chai's techniques for formulating application queries would have involved the straightforward application of conventional techniques and technologies. A person

of ordinary skill in the art would have found the disclosures of Coffman, Kanevsky, Kneser, Zadrozny, and Chai sufficiently detailed to implement the techniques on which I rely, without undue experimentation.

- (ii) **“extracting one or more criteria and one or more parameters from one or more keywords contained in the one or more recognized words or phrases, wherein the parser extracts the one or more criteria and the one or more parameters using procedures sensitive to the determined context;” (5[d][ii])**

191. With respect to “**criteria**” and “**parameters**,” the ’006 patent generally describes “**criteria**” as attributes of a request, and “**parameters**” as the values of the attributes (although they are also described somewhat interchangeably). For example, for a request in a weather context, a criterion could be location, and the parameter would be the specific value for the location (e.g., “Alexandria”). For example, “[f]or a weather context,” according to the ’006 patent, “examples of criteria include, location, date and time. Other criteria can include command criteria (i.e., yes/no, on/off, pause, stop), and spelling.” (’006, 18:30-32; *see also id.*, 25:15-18 (same), 11:65-67 (“The system agent **150** manages the criteria handlers **152**, which handle specific parameters or values (criteria) used to determine context for questions and commands.”).)

192. It would have been obvious under the proposed combination to

“extract[] one or more criteria and one or more parameters from one or more keywords contained in the one or more recognized words or phrases.” As explained above, Coffman describes determining a context by “fit[ting] the utterance at the level of the NLU symbolic language” (Coffman, 42:4-5) and filling symbolic fields for a query (Coffman, 38:15-16 (“For NLU queries, the symbolic fields are filled for each stream”), 30:8-10 (“symbolic language which maps the actual query”), 41:16-21). For example, as noted, Coffman describes mapping the recognized natural language utterance “I would like to open this” into a “set of attribute value pairs: Command: open, Object: Windows or task selected by the last mouse click.” (Coffman, 6:32-7:3.) Coffman does not provide an example of how exactly an utterance is “fit...at the level of the NLU symbolic language” or how “symbolic fields are filled,” but it would have been obvious under the combination with Zadrozny and Chai to extract criteria and parameters from words contained in the recognized utterance. For example, as explained, Chai describes analyzing a user’s input (“I want to buy one Black CookTop of previous type”) to identify various slots (“**one or more criteria**”) (e.g., color) and their values (“**one or more parameters**”) (e.g., black) for an action template.

193. It would have also been obvious for the claimed **“extracting”** to **“us[e] procedures sensitive to the determined context.”** As explained above, Coffman

describes using context to determine a conversationally-aware application and call/query. Therefore, by matching a user's words and phrases to fields in the symbolic language of an application call/query, Coffman's techniques, under the proposed combination, would have predictably resulted in **“extracting one or more criteria and one or more parameters ... using procedures sensitive to the determined context.”**

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

194. 5[d][iii] would have been obvious in view of Chai. As discussed above, Chai discloses filling “slots” in action templates with values based on words and phrases input by a user. In one example, slots are filled based on the user's statement, “I want to buy one Black CookTop of previous type.” (Chai, 12:28-32.) The user's statement, however, does not fill all required slots in the action template(s). For example, a value for the required “company” slot is missing. (Chai, 12:28-56.) Chai teaches that missing slots can be filled in by **“inferring”** values from context:

The dialog manager performs discourse analysis (as in e.g. Allen 1995), which means looking at the discourse history of the current user session, the history of the user interactions in previous sessions, and the

preferences of the user to determine if any of the missing slot 807 values can be inferred from context. e.g. the company name for buying a stock might be inferred from the context of the natural language conversation with the machine. In the above example, the value of the attribute “size” can be inferred from the long term history of interactions with the current user and the value of the attribute “color” may be inferred from the conversational history of current session (e.g. the user might say “show me blue shirts” and “I would like to buy two of these”, where in the second sentence, the user really means “I would like to buy two of these blue shirts”). After filling slot values based on context (if possible), in step 820, the dialog manager chooses the best template 806 from among the matched templates based on some best choice criteria.

(Chai, 12:1-20.)

195. Chai therefore discloses **“inferring one or more further criteria and one or more further parameters”** by reviewing context to infer that a value (**“parameter”**) corresponds to a particular slot (**“criteria”**) whose value is missing from an action template—for instance, using an example from Chai above, inferring that “blue” corresponds to a “color” slot and using that information to fill in an action template with a “color” slot.

196. It would have been obvious to a person of ordinary skill in the art to apply Chai’s techniques to Coffman. As noted above, Coffman likewise discloses a “context” for an active process that includes similar information to Chai, including

“past history of the discourse, its current state, and the specific characteristics and full query arguments” and “any data needed for input understanding.” (Coffman, 29:18-30:3.) A person of ordinary skill in the art would have therefore readily appreciated that Chai’s techniques to “infer” criteria and parameters could be applied to Coffman’s system, and recognized the similar benefits that would flow from it, including the ability to complete a user’s query without prompting the user to supply information that the system could determine on its own. This would result in a more efficient and natural dialogue with the user, making the system easier to use.

197. A person of ordinary skill in the art also would have understood that the “inferring” **“us[ed] a dynamic set of prior probabilities or fuzzy possibilities.”** As explained above for claim 1, the “dynamic set of prior probabilities or fuzzy probabilities” under the proposed combination corresponds to the set of trigram probabilities in language model(s) used for speech recognition, such as an adaptive language model. Therefore, it would have been obvious that recognized speech using trigram probabilities (**“using the dynamic set of prior probabilities or fuzzy possibilities”**) from an utterance could provide context information used to fill in missing slot values, predictably resulting in the **“inferring”** step taking place by **“using a dynamic set of prior probabilities or fuzzy possibilities.”**

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

198. 5[d][iv], despite its length, simply corresponds to the step of filling in required and optional query fields (or action “slots”) based on a grammar of constructions as taught by the combination of Coffman, Zadrozny and Chai. “[T]he **one or more extracted criteria, the one or more extracted parameters, the one or more inferred criteria, and the one or more inferred parameters,**” as discussed for 5[d][ii] and 5[d][iii], correspond to recognized words and phrases from the user that are used to fill required and optional slots and their values. The recognized information under the proposed combination would have also been **“transformed...into one or more tokens.”** This is because, consistent with the ordinary understanding of **“token”** as generally referring to an individuated or non-reducible text element in data being parsed, the recognized information would be used to construct a request having a format consistent with the grammar of constructions and its slots as taught by Zadrozny, which would in turn be formatted into a call/query for the conversationally-aware application to process. (*See, e.g.,*

Microsoft Computer Dictionary (5th ed. 2002) [EX1009], p.522 (defining “token” as “2. Any nonreducible textual element in data that is being parsed—for example, the use in a program of a variable name, a reserved word, or an operator.”).) And both of these formats would have been requests formulated “**in the grammar used by the domain agent**”—the initial formulation because it would use the grammar of constructions to identify and fill in slots, and the second formulation (for an application call/query) because it is based on and derived from that initial formulation. By using the tokens to identify and fill in slots, moreover, the tokens would “**hav[e] a format compatible with the grammar used by the domain agent.**”

199. A person of ordinary skill in the art would have also understood and found it obvious for “**the one or more tokens [to] include all the required values and one or more of the optional values**” for the formulated requests. As noted, Coffman describes executing “completed” queries, while providing feedback for “incomplete or ambiguous queries/commands.” (Coffman, 40:9-19.) Chai teaches that a “complete” query can involve a query where all required slots are filled and some or all of the optional fields are filled. (Chai, e.g., 13:15-19 (“Both the templates **806** have all the required slots **807** filled after the discourse context is taken into consideration but since the second template has more parameters filled,

the second template is chosen as the best matching template in the preferred implementation.”.)

- (e) **“processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance; and” (5[e])**

200. 5[e] is verbatim identical to 1[e]. Therefore, for the reasons explained above in my analysis of 1[e], 5[e] is disclosed by and obvious over the proposed combination.

- (f) **“presenting the generated response to the utterance via the speech unit.” (5[f])**

201. 5[f] is verbatim identical to 1[f]. Therefore, for the reasons explained above in my analysis of 1[f], 5[f] is disclosed by and obvious over the proposed combination.

E. Ground 4: Claims 6 and 9 Are Obvious Over Ground 1 Prior Art (Coffman, Kanevsky, Kneser and Zadrozny) in Further View of Redfern.

- 1. **Claim 6: “A method for processing natural language speech utterances with context-specific domain agents, comprising:”**

202. Like claim 1, the preamble of claim 6 recites “[a] **method for processing natural language speech utterances with context-specific domain agents, comprising.**” As shown below, the claimed method is disclosed and rendered obvious by Ground 1 prior art in further view of U.S. Patent No. 6,078,914

(Redfern). Redfern describes a meta-search application where a user can provide a single natural language query to the application, and the application will formulate queries to multiple search engines and return what the application deems the best search results by evaluating the results using the natural language query. (Redfern, e.g., Abstract.) As explained below, under the proposed combination with Redfern, a meta-search application configured in accordance with Coffman to be conversationally-aware would be a **“context-specific domain agent.”**

(a) “receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request, wherein the request includes at least one command and at least one question;” (6[a])

203. As I explained in my analysis of 1[a], Ground 1 prior art discloses and renders obvious **“receiving, at a speech unit coupled to a processing device, a natural language speech utterance that contains a request.”** As explained below, it would have further been obvious in view of Redfern for the **“request”** to **“include[] at least one command and at least one question.”**

204. With respect to a request that **“includes at least one command and at least one question,”** the '006 patent states that a question and a command are similar but that a question typically results in a response while a command results in an action. ('006, 5:59-62 (“The processing of these commands is performed in a nearly identical manner to a question. One difference being that the result of the command

is generally an action rather than a response.”), 10:67-11:4 (“Commands typically result in actions taken by the system **90** itself (i.e., pause or stop), or to a remote device or data source (i.e., download data or program, or control a remote device), through the network interface to the Internet or other data interface.”).) The ’006 patent further states that an utterance can include both aspects of a question and a command. (’006, 3:46-52 (“Some utterances can include both aspects of a command and a query or question. For example, a user may say, ‘record my favorite TV program’. A query is required to determine the name, the channel, and time for the users favorite TV program. A command must be executed to set a video recorder to capture this program. ”).)

205. It would have been obvious under the proposed combination for the request in the user’s utterance to “**include at least one command and at least one question.**” With respect to the “**at least one question,**” Redfern provides an example of a user question, “Where do Monarch butterflies spend the winter?,” which is translated into the request “rank1(Monarch) phrase(butterflies spend) phrase (winter)” to be processed by the meta-search application:

An example of the operation of an embodiment of the present invention is given below. In the example, the user has entered “Where do Monarch butterflies spend the winter?” as the Natural Language Search Data 28. The processed search data from the Natural Language Query

Processor 24 is “rank1(Monarch) phrase(butterflies spend) phrase (winter)” and this is passed to meta search engine 32.

In this example, a set of search engines 36 has been previously selected and includes the Lycos, AltaVista and Excite engines. Meta search engine 32 simplifies the processed search data 28 for each search engine in the set to obtain simplified search data appropriate to each engine. Specifically, for the Lycos engine, the search data which is dispatched is, "Monarch+butterflies+spend+winter". For the AltaVista search engine, the search data is, "Monarch+AND+(butterflies+OR+spend)+AND+(winter) Monarch ranked first". Finally, for the Excite search engine, the search data is, "Monarch+AND+(butterflies+OR+spend)+AND+(winter)". This search data is appropriately combined with the URL for each respective search engine and is transmitted to the search engine.

(Redfern, 15:42-65.) It would have been obvious, under the proposed combination, for the user's question to be provided as part of a spoken utterance.

206. The “**at least one command**” under the proposed combination could have taken a number of forms. For example, the command could relate to selecting the set of search engines/information sources to query. Redfern explains that “[i]t is contemplated that the selection of an appropriate set of search engines can either be performed explicitly by the user, or implicitly by the search system 20, based upon recognized keywords in the processed search data 28 or other information such as

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

the user's identity, location, etc.” (Redfern, 9:22-27.) As Redfern notes, the command to specify the set of search engines/information sources could be explicit or implicit. For instance, it would have been obvious for a user to explicitly specify the set of search engines/information sources for a search with an utterance such as “Search Lycos, AltaVista and Excite for where do Monarch butterflies spend the winter?” As another example, as Redfern notes, the command for the set of search engines/information sources for a search could be implicitly specified by the particular keywords in the user's query, consistent with the '006 patent's statement that an utterance can include aspects of both a question and a command. ('006, 3:46-52.) As another example, a well-known command during web browsing or other network-based activities is to stop/cancel a pending operation. It would have been obvious for a user's utterance to include a command to stop/cancel a pending search operation in order to enter a new one. For example, “cancel that search, find out where do Monarch butterflies spend the winter?” These examples includes a “**command**” because they instruct the system to take some action—such as select and use particular search engines or stop processing. The ability to enter commands of this nature using speech utterances would have been especially useful in situations and devices with limited input/output capabilities beyond voice as contemplated by Coffman.

207. **Rationale and Motivation to Combine (Ground 1 Prior Art and Redfern)**. It would have been obvious to a person of ordinary skill in the art to combine Redfern's teachings regarding a meta-search application with Coffman's teachings regarding conversationally-aware applications. The combination would have predictably resulted in a meta-search application as taught by Redfern with the ability to interface with Coffman's conversational system, thereby advantageously enabling, for example, a user to use the meta-search application by providing spoken natural language input and with a variety of devices with different input and output capabilities.

208. Redfern is an analogous reference in the same field of speech recognition and processing, including using applications with natural language input, as the '006 patent. ('006, 1:20-22; Redfern, 1:6-13 ("[T]he present invention relates to a meta-search system and method for accepting a natural language query which is processed to retrieve information from one or more information sources via at least one search engine and to extract relevant portions of those information sources for output to the originator of the query").) Redfern would have also been reasonably pertinent to a number of problems facing the patent inventors, including how to interface and control an application using natural language input and how to query multiple information sources using natural language input.

209. A person of ordinary skill in the art would have been motivated to combine. Coffman itself describes conversational search functionality as well as integration with websites. (Coffman, 48:25-49:16 (search capability), 58:32-59:3 (“Also, a Java, CGI and XML/HTML-based web site written with Spoken Age can be immediately converted into an IVR providing services through the phone or a restricted speech mark-up language to be accessed with a small embedded conversational browser.”).) Redfern provides an express motivation for a meta-search, explaining that “[t]o enhance the chances that the desired information will in fact be located, a user will often perform the same search on multiple search engines thus exacerbating the number of matches which must be reviewed by the user. The use of more than one search engine can also require the user to redraft his search data to accommodate different search data requirements and/or capabilities of the different search engines. For example, some search engines may only allow keyword-based searches while others may permit searching based upon phrases.” (Redfern, 1:66-2:8.) Redfern explains that “[i]t is desired to have a meta-search engine which will accept natural language search data to search for information from one or more information sources which need not be homogeneous or well defined, the meta-search engine would identify portions of the matching information which it determines to be relevant to the search data and would display at least those

determined portions to the user.” (Redfern, 2:38-44.) Moreover, Redfern itself already describes the application’s ability to receive natural language search data. (Redfern, 4:28-31.) A person of ordinary skill in the art would have readily appreciated that adapting and incorporating Redfern’s meta-search application for Coffman’s conversational system would enhance a user’s ability to input natural language search queries and instructions.

210. A person of ordinary skill would not have perceived any technical hurdles to the combination, and in fact would have considered Redfern and Coffman to be naturally combinable using conventional techniques and technologies. For example, as noted, Redfern itself already describes the application’s ability to receive natural language search data. (Redfern, 4:28-31.) Moreover, Coffman explains that a conversational system can be either client- or server-based and can register both client- and server-based applications. (Coffman, e.g., 12:5-7 (conversational system “may be implemented on a client device or a server.”), 13:1-3 (server or any client device), 22:32-23:3 (“...network conversational applications 200, 200a and network-connected devices (local client and other networked devices such as a server) to exchange information register [sic] their current conversational state, arguments (data files 209) and context with each other.”), 55:13-19.) Redfern similarly does not limit the meta-search application to a remote or internet-based

site, describing it simply as “system **20**” and explaining that it could be used with search engines “accessible via a LAN... or even one or more systems integral with system **20**.” (Redfern, 10:25-38; *see also id.*, 4:8-9 (“FIG. 1 shows a meta-search system **20** in accordance with an embodiment of the present invention”).) A person of ordinary skill in the art would have recognized that any particular architecture of the conversational system would be a matter of design choice.

211. A person of ordinary skill in the art would have therefore had a reasonable expectation of success, and would have found the disclosures of Coffman, Kanevsky, Kneser, Zadrozny, and Redfern sufficiently detailed to implement the techniques on which I rely, without undue experimentation.

(b) “parsing, at a parser coupled to the processing device, information relating to the utterance to determine a meaning associated with the utterance and a context associated with the request contained in the utterance;” (6[b])

212. For purposes of my analysis, 6[b] is substantially identical to 1[c]. 1[c] recites parsing “the one or more recognized words or phrases,” while 6[b] recites parsing “information relating to the utterance.” This is not a meaningful difference because “information relating to the utterance” would plainly include the “the one or more recognized words or phrases.” This is confirmed by 5[c]’s similar recitation of parsing “information relating to the utterance...wherein the parsed information

includes the one or more recognized words and phrases.” Therefore, for the reasons explained above in my analysis of 1[c], 6[b] is disclosed by and obvious over the proposed combination.

- (c) **“formulating, at the parser, the request contained in the utterance in accordance with a grammar used by a domain agent associated with the determined context;”**
(6[c])

213. 6[c] is verbatim identical to 1[d]. I therefore incorporate my analysis of 1[d] here. I note that, under the proposed further combination with Redfern, the recited **“domain agent”** would take the form of a meta-search application configured in accordance with Coffman to be conversationally-aware. The meta-search application receives a natural language query and evaluates it to produce queries to submit to multiple information sources, such as the well-known Lycos, AltaVista and Excite web-based search engines. (Redfern, e.g., 2:52-3:9, 4:8-34, 8:66-9:12, 15:43-65.) The application then scores the results and extracts relevant information to present to the user. (Redfern, e.g., 2:52-3:9, 15:19-41, 16:47-60.)

214. With respect to **“formulating...the request,”** as explained for 1[d], the formulated request corresponds to an application call/query to a conversationally-aware application (**“domain agent”**). Therefore, under the proposed combination with Redfern, **“the formulated request”** would correspond to an application call/query to the meta-search application (**“domain agent”**) that included search data

resulting from processing and formatting the user's natural language query. (Redfern, e.g., 4:35-8:65 (describing processing and formatting).) For example, the natural language query "Where do Monarch butterflies spend winter?" is processed and formatted as "rank1(Monarch) phrase(butterflies spend) phrase(winter)" to pass to the meta-search application. (Redfern, 15:44-49.)

(d) **"processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance, wherein processing the formulated request with the domain agent includes:" (6[d])**

215. 6[d], like 1[e], recites **"processing the formulated request with the domain agent associated with the determined context to generate a response to the utterance."** This portion of 6[d] is therefore disclosed by and obvious over the proposed combination.

216. 6[d] further recites **"wherein processing the formulated request with the domain agent includes"** the following steps. As explained below, these steps would have been obvious in further view of Redfern.

(e) **"directing the at least one command to one or more local or remote processing devices to execute the at least one command; and" (6[e])**

217. As explained above, the **"command"** under the proposed combination could have taken the form of a user command to select (explicitly or implicitly) the

search engines/information sources to query, or alternatively, to cancel a pending search.

218. With respect to a command to select (explicitly or implicitly) the search engines/information sources to query, it also would have been obvious for the conversationally-aware meta-search application under the proposed combination (“**domain agent**”) to process the application call/query (“**process the formulated request**”) by “**directing the at least one command**” to “**local or remote processing devices.**” For example, Redfern explains that, in order to select and use particular search engines, the application consults “a database of search engine capabilities, requirements and addresses (URL’s or other appropriate address information) ... to determine the appropriate parameters for each search engine in the selected set of search engines.” (Redfern, 9:34-38.) By consulting such a database in this manner, the application (“**domain agent**”) “**direct[s] the at least one command**” to the database system (“**to local or remote processing devices**”).

219. With respect to a command to cancel a pending search, it would have been obvious for the conversationally-aware meta-search application under the proposed combination (“**domain agent**”) to process the application call/query (“**process the formulated request**”) by “**directing the at least one command**” to “**local or remote processing devices.**” For example, a person of ordinary skill in

the art would have understood that cancelling a pending search operation would have entailed instructing local and remote processing devices to stop processing related to the search and close network connections.

- (f) **“querying one or more local or network information sources to resolve information requested in the at least one question, wherein querying the one or more local or network information sources further includes:”**
(6[f])

220. It would have been obvious under the proposed combination for the conversationally-aware meta-search application (“**domain agent**”) to process the application call/query (“**process the formulated request**”) by “**querying one or more local or network information sources to resolve information requested in the at least one question**” (e.g., by sending search requests to multiple web-based search engines or other information sources). As discussed above, Redfern discloses a specific example where multiple search engines (“**local or network information sources**”) are queried to resolve the user’s question, “Where do Monarch butterflies spend the winter?” (Redfern, 15:42-65; *see also id.*, 10:25-33 (“Meta search engine **32** receives processed search data **28** and further processes it to place it into forms suitable for the search engine or engines **36** which are defined for the information sources to be searched.”), 2:60-63 (“(iii) creating search data from said extracted terms in an [sic] form appropriate to each of said at least one search engines and

transferring said created search data thereto to initiating a search;”).)

221. As explained below, it would have been obvious under the proposed combination for “**querying the one or more local or network information sources [to] further include[]**” the following steps.

- (g) “**submitting multiple asynchronous queries to the one or more local or network information sources, wherein the multiple queries include one or more duplicate queries submitted to distinct information sources;**”
(6[g])

222. As explained above, Redfern discloses that the meta-search application submits “**multiple ... queries**” to “**the one or more local or network information sources.**” For example, to resolve the user’s question “Where do Monarch butterflies spend the winter?” the application submits a query of “Monarch+butterflies+spend+winter” to the Lycos search engine, a query (and further instruction to list the results having “Monarch” in them first) of “Monarch+AND+(butterflies+OR+spend)+AND+(winter) Monarch ranked first” to the AltaVista search engine, and a query of “Monarch+AND+(butterflies+OR+spend)+AND+(winter)” to the Excite search engine. (Redfern, 15:42-65; *see also id.*, 25:19-21 (describing AltaVista “selection criteria” and “results ranking criteria”).)

223. Redfern further discloses that the submitted queries are

“**asynchronous**” because the meta-search engine submits each query without waiting for an earlier one to be completed or time-out. For example, Redfern describes submitting queries to each search engine and then processing the results as they are returned from each search engine. (Redfern, 15:63-65 (“This search data is appropriately combined with the URL for each respective search engine and is transmitted to the search engine.”), 10:38-44 (“At step **532** in FIG. 5, ‘hits’ **42** (in FIG. 1) from each search engine are received by meta-search engine **32**. These hits are then passed to Search Results Filter **46** when results have been obtained from all of the search engines in the set or when a predetermined time limit has been exceeded without receiving results from one or more search engine.”).) And a person of ordinary skill in the art would have expected steps **528** and **532** of Redfern Figure 5 to be depicted as a loop if the queries and responses were submitted and received *synchronously* rather than asynchronously, but they are not. (Redfern, Fig. 5 (steps **528** and **532** showing that all queries are dispatched in step **528** and all results received at step **532**).) Moreover, it would have been obvious to a person of ordinary skill in the art for the queries to be “asynchronous” in order to more quickly receive and return results to the user. This is because with synchronous requests, each query would be processed before going to the next one. This sort of serial processing would fail to take advantage of the performance benefits provided by having

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

multiple search engines processing queries substantially simultaneously. Redfern itself acknowledges the need for a timely response to the user, describing obtaining results before “a predetermined time limit has been exceeded.” (Redfern, 10:38-44.)

224. With respect to “**the multiple queries include one or more duplicate queries submitted to distinct information sources,**” a person of ordinary skill in the art would have understood and found it obvious that “**one or more duplicate queries**” would be submitted by the meta-search application across the selected set of search engines/information sources. To begin with, nothing in the ’006 patent suggests that queries must be textually verbatim or in an identical format in order to qualify as “**duplicate queries.**” Redfern’s example queries about Monarch butterflies are three “duplicate” queries because they are substantively identical in seeking the same information. This is a similar purpose for duplicate queries as described in the ’006 patent, which states that “[d]uplicate queries are sent to different information sources to ensure that at least one source responds with a useful result in a timely manner.” (’006, 19:26-28.) But even if queries were required to be verbatim in order to qualify as “**duplicate queries,**” this is also disclosed by and obvious over Redfern. For example, as noted above, Redfern identifies an Excite query of “Monarch+AND+(butterflies+OR+spend)+AND+(winter).” This is identical to the AltaVista query,

“Monarch+AND+(butterflies+OR+spend)+AND+(winter).” While the submission to AltaVista also includes the text “Monarch ranked first,” this is an additional instruction to list the results having “Monarch” in them first. (See Redfern, 25:19-21 (describing AltaVista “selection criteria” and “results ranking criteria”).)

225. Lastly, the Lycos, AltaVista and Excite search engines (as well as others described by Redfern such as Lexis and HotBot (Redfern, e.g., 9:4-9, 9:66-10:1)) are clearly **“distinct information sources”**—the whole point of Redfern’s meta-search application is to enable a user to conveniently query multiple distinct information sources at once. (Redfern, 2:47-51 (“It is an object of the present invention to provide a novel meta-search system and method for obtaining information relevant to a natural language query from a plurality of information sources which obviates or mitigates at least one disadvantage of the prior art.”), 2:37-40.)

(h) **“asynchronously receiving multiple results to the multiple asynchronous queries from the one or more local or network information sources; and” (6[h])**

226. Redfern describes **“asynchronously receiving multiple results to the multiple asynchronous queries”** from the search engines (**“information sources”**):

At step 532 in FIG. 5, ‘hits’ 42 (in FIG. 1) from each search engine are received by meta-search engine 32. These hits are then passed to Search Results Filter 46 when results have been obtained from all of the search

engines in the set or when a predetermined time limit has been exceeded without receiving results from one or more search engine.

The hits received by Search Results Filter **46** are generally in the form of an address, such as a URL, at which a relevant information source can be located and the identity of the search engine which returned the hit. Search Results Filter **46** combines the hits from each search engine into a single list and removes redundancies.

(Redfern, 10:38-50.) Moreover, for substantially the same reasons as those discussed above for submitting “asynchronous” queries, “asynchronously receiving” the query results would have been obvious to a person of ordinary skill in the art.

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

227. Redfern describes a process of determining the most relevant search hits to present to the user (i.e., **“determin[ing] one or more best responses to the at least one question”**) that involves **“scoring”** search hits (**“the multiple asynchronously received results”**) by matching words and phrases from the user’s natural language input against text from each search hit. Redfern explains:

The hits received by Search Results Filter **46** are generally in the form of an address, such as a URL, at which a relevant information source

can be located and the identity of the search engine which returned the hit. Search Results Filter **46** combines the hits from each search engine into a single list and removes redundancies. The culled list of hits is placed into the format necessary to retrieve the individual information sources and this formatted list is transferred to Information Retrieval means **50**.

From this formatted list, Information Retrieval means **50** retrieves the complete information sources **54** for each of a preselected maximum number of hits from each search engine **36**. For example, the first **10** hits from each engine, after redundancies have been removed, may be retrieved.

The retrieved information sources are then examined by the Selector means **58**. Selector means **58** performs several functions, including ranking the relevancy of the information sources retrieved and identifying their relevant portions for output to the user.

The process for ranking of the information sources employs the processed search data **28** from Natural Language Query Processor **24**. Specifically, as illustrated at step **680** of FIG. **6**, a scoring regime is established for the retrieved information sources relative to the processed search data **28** and a score table is created to hold determined scores for each information source. A presently preferred scoring regime is given in Appendix B. In this regime, each group in processed search data **28** is treated as a separate candidate and separate totals are maintained for each candidate in the score table. An example of processed search data **28** which reads, “or(phrase(contact),

phrase(personnel), phrase(names)); phrase(people); rank1(Gravis);
rank1(Logitech))” has four candidates.

(Redfern, 10:44-11:10.) The search hit(s) having the highest scoring “segments,” or excerpts, with scores “no less than a preselected level” are selected to present to the user. (Redfern, 13:9-11 (“At step **800** the final segment from each information source is ranked in descending order, by their respective determined scores.”), 14:41-46 (“The final segment determined for this retrieved information source is ranked against the final segments previously determined for the other retrieved information sources and is added to the sorted final segments obtained at step **800**.”), 15:19-31 (“As a final step of Selector means **58**, the highest ranked, ‘cleaned up’ segment is selected for output to the user, as is each cleaned up segment whose score is no less than a preselected level.”).)

228. For example, Appendix K from Redfern illustrates the top two highest scoring responses—i.e., the most relevant received results—to the user’s question “Where do Monarch butterflies spend the winter?” As shown below, the highest scoring response is a hit from the Alta Vista search engine, and the second highest scoring response is a hit from the Excite search engine:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

APPENDIX K

Result #2

Ranking: 3.3608

From: dnr.state.il.us

Found with: Alta Vista

While most insects hibernate, the monarch is the only species of butterfly which actually flies to warmer weather (migrates) in winter. Monarchs from Illinois spend their winters in California and Mexico. In the fall, people have reported seeing entire trees covered with thousands of migrating monarchs! However, only about 1 percent of these monarchs actually survive the journey back to Illinois.

Result #3

Ranking: 2.47543

From: www.tnc.org.

Found with: Excite

The Monarch is the only butterfly that annually migrates both north and south. In late September, for instance, they begin to congregate at Cache River Joint Wetlands Project in Illinois, a Conservancy preserve, and move southward for their annual migration. By October, they've flown hundreds of miles. Millions of Monarch butterflies return to their winter habitat in Mexico via the Devils River Corridor, which flows through the heart of Dolan Ranch Preserve in Texas. Monarchs continue south to the Sierra Madre of Mexico, where they spend the winter. By January, the Mexican fir trees and mountainsides are full of Monarchs, drifting, gliding, fluttering and basking.

(Redfern, 34:13-35.)

229. Redfern's process, under the proposed combination, "**us[es] the dynamic set of prior probabilities or fuzzy possibilities**" to determine the most relevant search hits ("**one or more best responses to the at least one question**") because the process would have used the recognized words and phrases provided from the user, which as discussed for claim 1 were determined using the dynamic set of prior probabilities in the trigram language model ("**the dynamic set of prior probabilities or fuzzy possibilities**"). The relevancy scoring, as described above,

is based on the “processed search data” which is directly derived from the words recognized using these probabilities.

230. As explained below, it would have been obvious under the proposed combination for **“scoring the relevance associated with the multiple asynchronously received results to determine the one or more best responses [to] include[]”** the following steps.

- (j) **“determining one or more tokens required to formulate a response to the at least one question;”**
(6[j])

231. Redfern discloses **“formulat[ing] a response to the at least one question”** by describing a process for identifying and extracting the best “segment,” or portion, of web pages corresponding to the search hits returned by each of the search engines that are responsive to the user’s question. (*See generally* Redfern, 10:45-15:41, Figs. 6-6c.) As explained below, Redfern further discloses that this process involves **“determining one or more tokens,”** i.e., identifying words and phrases in the user’s search data that are used in scoring the search hits.

232. In particular, as noted, a **“token”** in information processing is generally understood as an individuated or non-reducible text element in data being parsed. (*See, e.g.*, Microsoft Computer Dictionary (5th ed. 2002) [EX1009], p.522 (defining “token” as “2. Any nonreducible textual element in data that is being parsed—for

example, the use in a program of a variable name, a reserved word, or an operator.”.)

With respect to Redfern, it explains that one of the first steps in the process to identify and extract the best “segment” is to take the search data corresponding to the user’s question and treat each group in the search data as a separate candidate for scoring. For example, as Redfern explains in the excerpt below, the user’s search data “or(phrase(contact), phrase(personnel), phrase(names)); phrase(people); rank1(Gravis); rank1(Logitech))” has four separate candidates ((1) “or(phrase(contact), phrase(personnel), phrase(names))”; (2) “phrase(people)”; (3) “rank1(Gravis)”; and (4) “rank1(Logitech)”).

The process for ranking of the information sources employs the processed search data **28** from Natural Language Query Processor **24**. Specifically, as illustrated at step **680** of FIG. **6**, a scoring regime is established for the retrieved information sources relative to the processed search data **28** and a score table is created to hold determined scores for each information source. A presently preferred scoring regime is given in Appendix B. In this regime, each group in processed search data **28** is treated as a separate candidate and separate totals are maintained for each candidate in the score table. An example of processed search data **28** which reads, “or(phrase(contact), phrase(personnel), phrase(names)); phrase(people); rank1(Gravis); rank1(Logitech))” has four candidates.

(Redfern, 10:64-11:10.)

233. Redfern's scoring candidates contain "**tokens required to formulate a response to the at least one question**" because they specify individuated words and phrases from the search data that are used to identify a segment and determine whether the segment meets a scoring threshold for being responsive to the user's question. In Redfern's example in the excerpt above, the tokens would include "contact," "personnel," "names," "people," "Gravis," and "Logitech." In Redfern's example question "Where do Monarch butterflies spend the winter?," which is translated into the search data "rank1(Monarch), phrase(butterflies spend), phrase(winter)," the tokens would include "Monarch," "butterflies spend" (and separately "butterflies" and "spend") and "winter." (Redfern, 11:4-30, 15:43-49.)

234. Redfern further explains that, if a score for a segment is too low, such as below a required minimum preselected level, then it is not provided as a response to the user. (Redfern, 15:19-31.) Therefore, a person of ordinary skill would have understood that the tokens were "**required to formulate a response to the at least one question**" because if the token match failed to reach the minimum level, then no segments ("**response to the at least one question**") would be provided to the user.

(k) **“applying one or more scraping criteria to the multiple asynchronously received results to extract one or more values for the asynchronously received results;” (6[k])**

235. As explained for limitations 6[h] and 6[i], the **“multiple asynchronously received results”** correspond to search hits returned from search engines. Redfern explains that search hits generally include a URL to the information source for the hit. An example of a search hit from the Excite search engine is below. As shown, it includes the URL <<http://www.isit.com/butterfly/articles/sendmon.htm>>:

91% Monarch Population Plummets in Mexico [More Like This]
URL: <http://www.isit.com/butterfly/articles/sendmon.htm>
Summary: Urquhart tagged the first monarchs 59 years ago--and graciously agreed to tag the first paper monarchs for this symbolic migration! All migratory monarchs east of the Rocky Mountains spend the winter in just 9 major sanctuaries in Mexico!

(Redfern, 28:37-42.) Another search hit, from the AltaVista search engine, includes the URL <<http://dnr.state.il.us/NREDU/CLASSRM/symbol/insect.htm>>:

Monarch Butterfly
STATE INSECT - MONARCH BUTTERFLY *Danaus plexippus*. The monarch was chosen in 1975 to be Illinois' state insect. The third grade classes at Dennis School . . .
<http://dnr.state.il.us/NREDU/CLASSRM/symbol/insect.htm> - size 2K - 29
Apr 96

(Redfern, 24:38-43.)

236. Redfern **“extract[s] one or more values for the asynchronously received results”** by having the meta-search application automatically download

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

web pages pointed to by search hit URLs and search through the downloaded web pages to find and record values in the pages matching search data tokens (for example, finding and recording “Monarch” in the web pages). Redfern explains, for example:

Search Results Filter **46** combines the hits from each search engine into a single list and removes redundancies. The culled list of hits is placed into the format necessary to retrieve the individual information sources and this formatted list is transferred to Information Retrieval means **50**. From this formatted list, Information Retrieval means 50 retrieves the complete information sources 54 for each of a preselected maximum number of hits from each search engine 36. For example, the first 10 hits from each engine, after redundancies have been removed, may be retrieved.

The retrieved information sources are then examined by the Selector means 58. Selector means **58** performs several functions, including ranking the relevancy of the information sources retrieved and identifying their relevant portions for output to the user....

Next, at step **688**, a first retrieved information source is selected. At step **692**, the information source is examined to determine each match between its contents and the groups in the score table. For each match, an entry is made in the score table for the corresponding candidate including the score assigned the match under the selected scoring regime and the location of the match within the information source.

(Redfern, 10:48-63, 11:31-38.) As Redfern explains in the excerpt above, for each

value found in a web page that matches a token, “an entry is made in the score table for the corresponding candidate including the score assigned the match under the selected scoring regime and the location of the match within the information source.” (Redfern, 11:31-38.)

237. Redfern further discloses that, to extract the one or more values, **“one or more scraping criteria”** are applied to the search hits (**“multiple asynchronously received results”**). The term **“scraping”** in this context generally refers to techniques where a computer program extracts data from human-readable output coming from another program. For example, “web scraping” refers to a process of programmatically retrieving a web page from a website server and parsing the page to extract desired data. Redfern’s meta-search application extracts the one or more values by applying **“one or more scraping criteria”** by identifying and removing redundant search hits, retrieving a preselected maximum number of hits from each search engine, and then searching each retrieved hit for values matching the specific tokens corresponding to the search data. (Redfern, 10:48-63, 11:31-38.)

- (I) **“evaluating the one or more extracted values using the dynamic set of prior probabilities or fuzzy possibilities to resolve ambiguous, incomplete, or conflicting information associated with the one or more extracted values; and” (6[I])**

238. Redfern, under the proposed combination, discloses **“evaluating the**

one or more extracted values using the dynamic set of prior probabilities or fuzzy possibilities” because the words and phrases in the retrieved web pages are matched and scored using tokens resulting from the speech recognition process described in Coffman. As explained, that speech recognition process uses the “dynamic set of prior probabilities” related to, for example, a trigram language model. For example, under the proposed combination, the user’s question “Where do Monarch butterflies spend the winter?” would have been recognized from the user’s speech using prior probabilities of a language model. Words in the user’s question are then used as search data tokens “Monarch,” “butterflies spend” (and separately “butterflies” and “spend”), and “winter.” (Redfern, 11:4-10, 15:43-49.)

239. Redfern’s process of matching and scoring (“**evaluating the one or more extracted values**”) includes “**resolv[ing] ambiguous, incomplete, or conflicting information associated with the one or more extracted values.**” This is because Redfern’s process seeks to identify the best-matching responses from the collection of search hits and excludes results that are incomplete or ambiguous and score below a preselected level. For example, Redfern describes excluding results having fewer than three matches. (Redfern, 11:39-47 (“At step **700**, a determination is made as to whether more than three matches were found within the information source. If three or fewer matches were found, the information source is assigned a

rank of zero....”).) As other examples, the scoring calculation weights against segments that are shorter in length, and also against hits with a high number of matches in just a few tokens and few matches for remaining tokens. (Redfern, 13:1-14.) Segments that are shorter in length or poorer matching would indicate that they contain ambiguous or incomplete information. The scoring process and use of a preselected level therefore operates to resolve ambiguous, incomplete or conflicting information by eliminating search hits whose low scores would reflect those hits having ambiguous, incomplete or conflicting information.

(m) **“selecting one or more best values for the one or more required tokens from the one or more evaluated values, wherein the one or more best responses include the one or more best values for the one or more required tokens; and” (6[m])**

240. Redfern’s meta-search application **“select[s] one or more best values for the one or more required tokens from the one or more evaluated values”** by selecting the best segments including values matching the tokens as a result of the scoring process. Redfern explains:

As a final step of Selector means **58**, the highest ranked, “cleaned up” segment is selected for output to the user, as is each cleaned up segment whose score is no less than a preselected level. In a present embodiment of the invention, up to the ten highest scoring segments whose scores are greater than 0.01 are output to the user as a first set and a second set

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,015,006 B2

of up to the next ten highest scoring segments whose scores are greater than 0.01 are also available for output to the user.

(Redfern, 15:19-28.)

241. For example, for the user's question "Where do Monarch butterflies spend the winter?," which as explained, includes search data tokens "Monarch," "butterflies spend" (and separately "butterflies" and "spend"), and "winter," Redfern identifies the following segment from the second-ranked web page hit from AltaVista as the best response, with underlining added to show values matching tokens:

Result #2

Ranking: 3.3608

From: dnr.state.il.us

Found with: Alta Vista

While most insects hibernate, the monarch is the only species of butterfly which actually flies to warmer weather (migrates) in winter. Monarchs from Illinois spend their winters in California and Mexico. In the fall, people have reported seeing entire trees covered with thousands of migrating monarchs! However, only about 1 percent of these monarchs actually survive the journey back to Illinois.

(Redfern, 34:15-23; *see also id.*, 24:38-43 (showing AltaVista search hit corresponding to the best segment), 16:44-45 (explaining that "Appendix K shows the final segments from these information sources, as output to the user by output means 62.")) As shown above, "**the one or more best responses**"—e.g., the

AltaVista segment excerpted above—**“include the one or more best values for the one or more required tokens”** because that segment includes the highest-scoring values for the search data tokens.

- (n) **“presenting the generated response to the utterance via the speech unit, wherein the response presented via the speech unit includes results associated with the one or more local or remote processing device executing the at least one command and the one or more best responses to the at least one question.” (6[n])**

242. As explained for 1[f], Coffman discloses presenting to the user results generated by the conversationally-aware application (**“presenting the generated response to the utterance”**) **“via the speech unit,”** for example, by using text-to-speech to output results at telephone 600. (Coffman, Fig. 6, 40:9-19, 13:8-13, 58:9-11, 39:3-5.)

243. Under the proposed combination with Redfern, 6[n] simply corresponds to outputting the results of Redfern’s metasearch application based on a user’s request that included a command and question—for example, using text-to-speech to present the highest-scoring segments for the user’s search query “Where do Monarch butterflies spend the winter?” or “Search Lycos, AltaVista and Excite for where do Monarch butterflies spend the winter?”

244. It would have been obvious to a person of ordinary skill in the art to output the results of Redfern’s metasearch application **“via the speech unit,”** such

as the speaker of telephone 600 as described by Coffman. Doing so would be consistent with Coffman's goal of a universal conversational computing system useable across a wide range of user devices, including those with limited input/output capabilities beyond voice. (Coffman, 12:10-17 ("The 'universality' aspect of the CUI 10 implies that the CVM system provides the same UI to a user whether the CVM is implemented in connection with a desktop computer, a PDA with limited display capabilities, or with a phone where no display is provided. In other words, universality implies that the CVM system can appropriately handle the UI of devices with capabilities ranging from speech only to speech to multi-modal, i.e., speech + GUI, to purely GUI."), 9:12-22.)

2. Claim 9: "The method according to claim 6, wherein scoring the relevance associated with the multiple asynchronously received results to determine the one or more best responses further includes:"

245. As explained below, it would have been obvious under the proposed combination for the step of **"scoring the relevance associated with the multiple asynchronously received results to determine the one or more best responses"** to further include the following steps.

- (a) **“determining that the dynamic set of prior probabilities or fuzzy possibilities failed to resolve the ambiguous, incomplete, or conflicting information associated with the one or more extracted values; and”**
(9[a])

246. As explained above for claim 6, Redfern discloses only providing responses whose scores are no less than a preselected level. (Redfern, 15:19-27.) As explained, the scoring process and use of a preselected level operates to resolve ambiguous, incomplete or conflicting information by eliminating search hits whose low scores based on matching extracted values would reflect those hits having ambiguous, incomplete or conflicting information. Under the proposed combination, the matching and scoring of extracted values would have been based on words recognized from the user’s speech using prior probabilities of a language model (i.e., using **“the dynamic set of prior probabilities or fuzzy possibilities”**). A person of ordinary skill in the art would have therefore understood and found it obvious that a determination of no response would represent a determination that the matching and scoring process (use of **“the dynamic set of prior probabilities or fuzzy possibilities”**) **“failed to resolve the ambiguous, incomplete, or conflicting information associated with the one or more extracted values.”**

- (b) **“requesting a user that spoke the utterance to provide additional information relating to the request, wherein the domain agent uses the additional information relating to the request to resolve the ambiguous, incomplete, or conflicting information associated with the one or more extracted values.” (9[b])**

247. Under the proposed combination, in a situation where the meta-search application failed to provide a response to the user’s question, it would have been obvious for the system to prompt the user to refine or restate their question (**“requesting a user that spoke the utterance to provide additional information relating to the request”**) and resubmit their query in order to identify a search hit meeting the meta-search application’s scoring criteria (**“wherein the domain agent uses the additional information relating to the request to resolve the ambiguous, incomplete, or conflicting information associated with the one or more extracted values”**). Identifying one or more search hits meeting the meta-search application’s scoring criteria would thus **“resolv[e] the ambiguous, incomplete, or conflicting information.”**

248. For example, Redfern describes “not obtaining acceptable results without multiple and/or recursive search attempts.” (Redfern, 2:9-11.) Although Redfern’s meta-search techniques are an attempt to reduce the likelihood of such a situation, a person of ordinary skill in the art would have recognized that it would remain a possibility. And as previously noted, Coffman itself describes prompting

the user for missing or ambiguous information. (Coffman, 41:16-21.) Although Coffman's description relates to resolving ambiguous or incomplete queries in order to submit them for processing by an application, it would have been obvious to a person of ordinary skill in the art to apply this known technique to improve Redfern's similar process in order to allow the user to refine or clarify the query.

VI. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS

249. As explained in **Part III.B**, I understand that "secondary considerations" may be used as indicia of nonobviousness, provided that any such consideration has a nexus to the invention claimed. I understand from Petitioner's counsel that Patent Owner in the related district court litigation has not yet identified any evidence related to secondary considerations with a nexus to the claimed invention. Nor am I aware of information, such as commercial success, unexplained results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others, suggesting that the claims addressed in this Declaration are not obvious. To the extent Patent Owner later provides information it claims relates to secondary considerations, I reserve the right to supplement my analysis and opinions to comment on it.

VII. CONCLUSION

250. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in this proceeding. If required, I will appear for cross-examination at the appropriate time. I reserve the right to offer opinions relevant to the invalidity of the challenged claims and/or offer testimony in support of this Declaration.

251. I hereby declare that all statements made herein of my own knowledge are true and that all statements are 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 18 U.S.C. § 1001.

Dated: April 4, 2024

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Chris' followed by a stylized surname.

Christopher M. Schmandt
Winchester, MA

Exhibit A

Christopher Schmandt

4 Longfellow
Winchester, MA 01890
+1-617-230-4257

Education

M.I.T., Master of Science, Visual Studies (Computer Graphics), 1980

M.I.T., Bachelor of Science, Computer Science, 1978

Professional Experience -- MIT

Media Laboratory, Principal Research Scientist, 1985-2018 (retired)
Director, Living Mobile Research Group (formerly Speech + Mobility)

Architecture Machine Group, Research Associate, 1980-1984

Architecture Machine Group, Research Assistant, 1979-1980

Architecture Machine Group, Graphics Programmer, 1977-1979

Departmental Undergraduate Research Opportunities Program Coordinator, 1984-2018

Laboratory Intellectual Property Committee 2001-2016, chair 2002-2009

Departmental Committee on Graduate Studies, 1996-2001, 2007-2018

Sponsored Research Activities

Alerting and Mobile Messaging, Digital Life Consortium, MIT Media Lab, 1997-2018

Acoustical Cues to Discourse Structure, National Science Foundation, Principal Investigator, 1995-1998

Parsing Radio, News in the Future Consortium, MIT Media Lab, 1993-1999

Desktop Audio, SUN Microsystems, Inc., Principal Investigator, 1989-1996

Voice Interaction in Hand Held Computers, Apple Computer, Principal Investigator, 1991-1993

Voice Interfaces for Network Services, AT&T, Principal Investigator, 1989-1991

Back Seat Driver, NEC, Principal Investigator, 1988-1991

Acoustic and Visual Cues for Speech Recognition, DARPA, co-Principal Investigator, 1986-1988

Personal Computers and Telephony, NTT Public Corporation, Principal Investigator, 1984-1989

Home Telecomputing, Atari, Inc., Principal Investigator, 1983

Testifying legal engagements within the last five years, representing party in italics

- *Microsoft v. SpeakWare*, 2019
- *Bumble Trading LLC v. Match Group LLC*, 2019
- *Tile Inc. v. Cellwitch Inc.*, 2019
- *Blackberry Ltd v. Facebook, Inc* 2020
- *EROAD Ltd v. PerDiem Co.*, 2020
- *Snap Inc. v. SRK Technology LLC*, 2020
- *Shopify Inc Ltd v. Express Mobile Inc.*, 2020
- *Express Mobile Inc. v Wix Ltd* 2020
- *Motorola Mobility LLC v. Ironworks patents, LLC.* 2021
- *Express Mobile Inc. v GoDaddy Inc* 2021
- *Bumble Trading LLC v. Kinectus, LLC.* 2021
- *Tile, Inc v. Linquet Technologies, Inc.* 2021
- *Facebook, Inc. v. Palo Alto Research Center, Inc.* 2021
- *Unified Patents, LLC v. Gesture Technology Partners, LLC.* 2021
- *Quantum Metric, Inc v. Content Square Isreal, Ltd.* 2021
- *Meta Platforms Inc. v. AlmondNet, Inc* 2022
- *Apple Inc. v. Zentian Limited* 2022
- *Google LLC v. Mira Advanced Technology Systems Inc* 2022
- *Google LLC v. Robocast* 2023
- *Google LLC v. Buffalo Patents LLC* 2023
- *Amazon v. AlmondNet* 2023
- *Google LLC v. Valtrus* 2023
- *Samsung Electronics Co., Ltd v. Stanton Techiya, LLC* 2024
- *Roku, Inc. v. AlmondNet, Inc.* 2024
- *LinkedIn v. AlmondNet, Inc* 2024

Professional Experience -- Intellectual Property

Co-inventor on U.S. Patents

- 5,177,685 for "Automobile navigation system using real time spoken driving instructions"
- 6,728,348 for "System for storing voice recognizable identifiers using a limited input device such as a telephone key pad"
- 6,937,986 for "Automatic dynamic speech recognition vocabulary based on external sources of information"
- 7,098,776 for "Methods and apparatus for vibrotactile communication"
- 7,392,280 for "Method for summarization of threads in electronic mail"
- 7,738,637 "Interactive voice message retrieval"
- 7,443,283 for "Methods and apparatus for connecting an intimate group by exchanging awareness cues and text, voice instant messages, and two-way voice communications"
- 7,865,560 for "System for summarization of threads in electronic mail"
- 8,121,653 for "Methods and apparatus for autonomously managing communications using an intelligent intermediary"
- 8,135,128 for "Animatronic creatures that act as intermediaries between human users and a telephone system"

Co-inventor on U.S. Patent Applications

- 20020076009 "International dialing using spoken commands"
- 20020087328 "Automatic dynamic speech recognition vocabulary based on external sources of information"

- 20020064257 "System for storing voice recognizable identifiers using a limited input device such as a telephone keypad"
- 20030023688 "Voice-based message sorting and retrieval method"
- 20030081738 "Method and apparatus for improving access to numerical information in voice messages"
- 20030144846 "Method and system for modifying the behavior of an application based upon the application's grammar"
- 20030158903 "Calendar bar interface for electronic mail interaction"

Publications

Field Study of a Tactile Sound Awareness Device for Deaf and Hard of Hearing Users ISWC 2020. (with D. Jain, B. Chiu, S. Goodman, L. Findlater, J. Froehlich)

SkinMorph: Texture-Tunable On-Skin Interface Through Thin, Programmable Gel ISWC 2018. (with Cindy Kao, M. Banforth, D. Kim)

Technical Interventions to Detect, Communicate, and Deter Sexual Assault. ISWC 2017. (with Manisha Mohan)

Exploring Interactions and Perceptions of Kinetic Wearables. DIS 2017. (with Cindy Hsin-Liu Kao, D. Ajilo, O. Anilonyte, A. Dementyev, I. Choi and S. Follmer)

Leveraging User-made Predictions to Help Understand Personal Behavior. MobileHCI 2017. (with Miriam Greis, Tilman Dangler and Albrecht Schmidt)

Rovables: On-Body Robots as Mobile Wearables. UIST 2016. (with Cindy Hsin-Liu Kao, A. Dementyev, I. Choi, D. Ajilo, M. Xu, and S. Follmer)

DuoSkin: Rapidly Prototyping On-Skin User Interfaces Using Skin-Friendly Materials. ISWC 2016. (with Cindy Hsin-Liu Kao, Christian Holz, Asta Roseway, and Andres Calvo)

Immersive Terrestrial Scuba Diving Using Virtual Reality (with Dhruv Jain, Misha Sra, Jingru Go, Rodrigo Margues, Raymond Wu and Justin Chiu) Proceedings, UIST 2016

Expanding social mobile games beyond the device screen (with Misha Sra) Journal of Personal and Ubiquitous Computing, 2015

NailO: Fingernails as an input surface (with Cindy Hsin-Liu Kao, Artem Dementyev, Joseph Paradiso) CHI 2015

Mugshots: A mug display for front and back stage social interaction in the workplace (with Cindy Hsin-Liu Kao) TEI (Tangible and Embedded Interfaces) 2015

Mime: compact, low power 3D gesture sensing for interaction with head mounted displays (with Andrea Colaco, Ahmend Kirmani, Hye Soo Yang, Nan-Wei Gong, and Vivek Goyal) Proceedings of UIST 2013.

Spotz: A location-based approach to self-awareness (with Misha Sra) Proceedings of Persuasive 2013.

Setting the stage for interaction: A tablet application to augment group discussion in a seminar class (with Drew Harry and Eric Gordon) Proceedings of CSCW 2012.

Indoor Location Sensing using Geo-Magnetism (with Jaewoo Chung, Matt Donahoe, Ig-Jae Kim, Pedram Razavai and Micaela Wiseman) Proceedings of International Conference on Mobile Systems, Applications, and Services (Mobisys) 2011.

My second bike: a TV-enabled social and interactive riding experience (with Jaewoo Chung, Kuang Xu, Andrea Colaco, and Victor Li) Proceedings of IEEE Communications and Networking Conference, Jan 2010.

Going my way?: User-aware route planner (with Jaewoo Chung) Proceedings CHI 2009.

Globetoddler: Designing for remote interaction between preschoolers and their traveling parents (with Paulina Modlitba) CHI 2008 Extended Abstracts

Are we there yet? - a temporally aware media player (with Matt Adcock and Jaewoo Chung), Australian User Interface Conference (AUIC) 2008

Physical embodiments for mobile communication agents (with Stefan Marti), UIST 2005

Giving the caller the finger: collaborative responsibility for cellphone interruptions (with Stefan Marti) Extended Abstracts, CHI 2005

Active Messenger: email filtering and delivery in a heterogeneous network (with Stefan Marti) Human-Computer Interaction Journal (HCI) Volume 20 (2005)

WatchMe: communication and awareness between members of a closely-knit group (with Natalia Marmasse) Proceedings of Ubicomp 2004

An audio-based personal memory aid (with S. Vemuri, W. Bender, S. Tellex and B. Lassey) Proceedings of Ubicomp 2004

Improving speech playback using time-compression and speech recognition (with Sunil Vemuri, Philip DeCamp, and Walter Bender) Proceedings of CHI 2004

Impromptu: managing networked audio applications for mobile users (with Kwan Lee, Jang Kim, and Mark Ackerman), Proceedings of MobiSys 2004

TalkBack: a conversational answering machine (with Vidya Lakshminpathy and Natalia Marmasse) Proceedings of UIST 2003

"ListenIn" to domestic environments from remote locations (with Gerardo Vallejo) Proceedings of the 2003 International Conference on Auditory Display (ICAD)

Safe & Sound: a wireless leash (with Natalia Marmasse) Extended Abstracts, Proceedings of CHI 2003

Mediated voice communication via mobile IP (with Jang Kim, Kwan Lee, Gerardo Vallejo, and Mark Ackerman), Proceedings of UIST 2002.

The Audio Notebook: Paper and pen interaction with structured speech (with Lisa Stifelman and Barry Arons), Proceedings of CHI 2001

Synthetic News Radio (with Keith Emnett) IBM Systems Journal, Vol. 39 Nos. 3-4, pp. 646-659, 2000.

Everywhere messaging (with Natalia Marmasse, Stefan Marti, Nitin Sawhney, and Sean Wheeler) IBM Systems Journal, Vol. 39 Nos. 3-4, pp. 660-677, 2000.

Location-aware information delivery with comMotion (with Natalia Marmasse), Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing, pp. 157-171, Springer, 2000.

Nomadic Radio: Scalable and contextual notification for wearable audio messaging (with Nitin Sawhney), Proceedings of CHI 1999.

Speaking and listening on the run: Design for wearable audio computing (with Nitin Sawhney), Proceedings of International Symposium on Wearable Computing, 1998.

Audio Hallway: A virtual acoustic environment for browsing, Proceedings of UIST 1998.

Dynamic Soundscape: Mapping time to space for audio browsing (with Minoru Kobayashi), Proceedings of CHI 1997.

CLUES: Dynamic personalized message filtering (with Matt Marx), Proceedings of CSCW 1996.

Using acoustic structure in a hand-held audio playback device (with Deb Roy), IBM Systems Journal, Vol 35, Nos. 3 and 4, 1996.

Mailcall: Message presentation and navigation in a nonvisual environment (with Matt Marx), Proceedings of CHI 1996

AudioStreamer: Exploiting simultaneity for listening (with Atty Mullins), short paper, CHI 1995.

Multimedia momadic services on today's hardware, IEEE Network, September/October 1994.

Putting people first: Specifying proper names in speech interfaces (with Matt Marx), proceedings of UIST 1994.

Chatter: A conversational learning speech interface (with E. Ly) AAAI Workshop on Intelligent Multi-Media Multi-Modal Systems, 1994.

Voice Communication with Computers: Conversational Systems. New York: Van Nostrand Reinhold. 1994.

Capturing, structuring, and representing ubiquitous audio (with D. Hindus and C. Horner), ACM Transactions on Information Systems, Vol. 11, No. 4, October 1993.

Speech Recognition Architectures for Multimedia Environments, (with E. Ly and B. Arons), Proceedings of the 1993 AVIOS Conference, September 1993.

Phoneshell: the Telephone as Computer Terminal, Proceedings of the ACM Multimedia Conference, August 1993.

Voicenotes: A Speech Interface for a Hand-Held Voice Notetaker (with L. Stifelman, B. Arons, and E. Hulteen), Proceedings of INTERCHI'93, April 1993.

From Desktop Audio to Mobile Access: Opportunities for Voice in Computing, book chapter in Advances in Human-Computer Interaction, Vol. 4, H.R. Hartson and D. Hix editors. 1992.

Ubiquitous Audio: Capturing Spontaneous Collaboration (with D. Hindus), Proceedings of CSCW'92, November 1992.

Integrating Audio and Telephony in a Distributed Workstation Environment (with S. Angebrannt, R. Hyde, D. Luong, and N. Siravara), Proceedings of the Summer 1991 USENIX Conference, June 1991.

Augmenting a Window System with Speech Input (with M. Ackerman and D. Hindus), Computer, IEEE Computer Society, Vol. 23, No. 8, August 1990.

Observations on Using Speech Input for Window Navigation (with D. Hindus, M. Ackerman, and S. Manandhar), Proceedings, Human-Computer Interaction, Interact '90, IFIP, August 1990.

Phonetool: Integrating Telephones and Workstations (with S. Casner), Proceedings, GLOBECOM '89, IEEE Communications Society, November 1989.

Desktop Audio (with B. Arons), UNIX Review, October 1989.

Synthetic Speech for Real Time Direction-Giving (with J. Davis), IEEE Transactions on Consumer Electronics, IEEE, September 1989.

The Back Seat Driver: Real Time Spoken Driving Instructions (with J. Davis), Proceedings, IEEE Vehicle Navigation and Information Systems Conference, IEEE, Toronto, Canada, September 1989.

An Audio and Telephone Server for Multi-media Workstations (with M. McKenna), Proceedings, Second IEEE Conference on Workstations, IEEE, Palo Alto, CA., 1988.

Employing Voice Back Channels to Facilitate Audio Document Retrieval, Proceedings, ACM Conference on Office Information Systems (COIS), Santa Clara, CA, 1988.

Conversational Telecommunications Environments, Proceedings, Second International Conference on Human-Computer Interaction, 1987.

Understanding Speech Without Recognizing Words, Proceedings, American Voice Input/Output Society Conference, AVIOS, 1987.

A Robust Parser and Dialog Generator for a Conversational Office System (with B. Arons and C. Simmons), Proceedings, American Voice Input/Output Society Conference, AVIOS, Palo Alto, CA, 1987.

Integrated Messages and Network Services for a Personal Workstation, IEEE Workshop on Telematics and Message Handling Systems, IEEE, 1986.

Voice Interaction in an Integrated Office and Telecommunications Environment, Proceedings, American Voice Input/Output Society Conference, AVIOS, San Francisco, CA, 1985.

Voice Communication with Computers, book chapter in Advances in Human-Computer Interaction, H. R. Hartson ed., 1985.

Speech Synthesis Gives Voiced Access to an Electronic Mail System, Speech Technology, Vol. 2, No. 3, Aug/Sept 1984.

A Conversational Telephone Messaging System (with B. Arons), IEEE Transactions on Consumer Electronics, IEEE, Vol CE-30, August 1984.

Phone Slave: A Graphical Telecommunications Interface (with B. Arons), Proceedings, Society for Information Display International Symposium, SID, San Francisco, CA, June 1984.

Input/Display Registration in a Stereoscopic Workstation, Displays, April 1984.

Remote Access to Voice and Text Messages, Proceedings, American Voice Input/Output Society Conference, AVIOS, Washington D.C., 1984.

Fuzzy Fonts: Analog Models Improve Digital Text Quality, Proceedings, National Computer Graphics Association Conference, Chicago, IL, 1983.

Greyscale Fonts Designed From Video Signal Analysis, Society of Applied Learning Technology, Houston, TX, 1983.

Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station, Proceedings, ACM/SIGGRAPH, Detroit, MI, 1983.

A Programmable Virtual Vocabulary Speech Processing Peripheral (with W. Bender), Proceedings, American Voice Input/Output Society Conference on Voice Data Entry Systems Applications, AVIOS, 1983.

The Intelligent Voice Interactive Interface (with E. A. Hulteen), Proceedings, Human Factors in Computer Systems, National Bureau of Standards/ACM, Gaithersburg, MD, 1982.

Interactive Three-Dimensional Computer Space, Proceedings, SPIE Conference on Processing and Display of Three-Dimensional Data, SPIE, San Diego, CA, 1982, Vol. 367.

Speech Communications, a Systems' Approach, Proceedings, American Voice Input/Output Society Conference on Entry Systems Applications, 1982.

Voice Interaction: Putting Intelligence into the Interface, Proceedings, IEEE International Conference on Cybernetics and Society, IEEE, Seattle, WA, 1982.

The Intelligent Ear: A Graphical Interface to Digital Audio, Proceedings, IEEE International Conference on Cybernetics and Society, IEEE, Atlanta, GA, 1981.

Soft Typography, Information Processing 1980, IFIPS, S. Lavington ed., North-Holland Publishing Co., 1980.
