

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,
Petitioner

v.

ZENTIAN LIMITED,
Patent Owner

Inter Partes Review Case No. IPR2023-00033
U.S. Patent No. 7,587,319

**DECLARATION OF CHRISTOPHER SCHMANDT
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 7,587,319**

TABLE OF CONTENTS

I.	INTRODUCTION AND QUALIFICATIONS	10
A.	EDUCATIONAL BACKGROUND AND PROFESSIONAL EXPERIENCE.....	10
II.	METHODOLOGY; MATERIALS CONSIDERED.....	13
III.	OVERVIEW AND LEGAL STANDARDS	15
A.	PERSON OF ORDINARY SKILL IN THE ART.....	16
B.	OBVIOUSNESS.....	17
C.	ANALOGOUS ART	22
D.	CLAIM CONSTRUCTION	23
IV.	LEVEL OF A PERSON OF ORDINARY SKILL	23
V.	OVERVIEW OF THE TECHNOLOGY	24
A.	SPEECH RECOGNITION.....	24
B.	FEATURE VECTORS	32
C.	ACOUSTIC MODELS	36
D.	HIDDEN MARKOV MODELS	37
E.	PRIOR ART SPEECH RECOGNITION SYSTEMS	45
VI.	OVERVIEW OF THE '319 PATENT	49
VII.	OVERVIEW OF THE PRIOR ART	50
A.	OVERVIEW OF <i>THELEN</i>	50
B.	OVERVIEW OF <i>BAILEY</i>	51
C.	OVERVIEW OF <i>CHEN</i>	52
D.	OVERVIEW OF <i>HUANG</i>	52
E.	OVERVIEW OF <i>JIANG</i>	53
F.	OVERVIEW OF <i>KATZ</i>	53
VIII.	OPINIONS REGARDING GROUND 1.....	54
A.	OBVIOUSNESS OF MODIFYING ELECTRONIC HARDWARE AND SOFTWARE TO BE CIRCUITRY	54
B.	CLAIMS 46, 50, 54, 64 AND 67 ARE OBVIOUS OVER <i>THELEN</i> IN VIEW OF <i>BAILEY</i> IN FURTHER VIEW OF <i>CHEN</i>	55
1.	<i>Claim 46</i>	55
a)	Claim 46(Pre): “A speech recognition circuit comprising”	55
b)	Claim 46(a): “an input buffer receiving processed speech parameters;”	55

c)	Claim 46(b): “a plurality of lexical memories containing in combination complete lexical data for word recognition, each lexical memory containing part of said complete lexical data;”	63
d)	Claim 46(c): “a plurality of processors connected in parallel to said input buffer for processing the speech parameters in parallel,”	70
e)	Claim 46(d): “said processors being arranged in groups of processors, each group of processors being connected to a lexical memory;”	72
f)	Claim 46(e): “a control processor controlling each processor to process said speech parameters using partial lexical data read from a respective said lexical memory; and”	78
g)	Claim 46(f): “a results memory storing the results of the processing of the speech parameters from said processors.”	82
2.	<i>Claim 50</i>	86
a)	Claim 50: “A speech recognition circuit according to claim 46, wherein each processor compares the processed speech parameters with the lexical data in a corresponding lexical memory to identify words as a word recognition event and sends information identifying the identified words to said results memory as the processing results.”	86
3.	<i>Claim 67</i>	88
a)	Claim 67: “A speech recognition circuit according to claim 46, wherein said lexical memories store said lexical data as Hidden Markov Models, and each processor performs the Viterbi search algorithm using a respective part of said lexical data.”	88

IX. OPINIONS REGARDING GROUND 2..... 90

A.	CLAIMS 47-49 ARE OBVIOUS OVER <i>THELEN</i> IN VIEW OF <i>BAILEY</i> AND <i>CHEN</i> IN FURTHER VIEW OF <i>HUANG</i>	90
1.	<i>Claim 47</i>	90
a)	Claim 47: “A speech recognition circuit according to claim 46, wherein each lexical memory stores lexical data as at least one lexical tree data	

	structure, each lexical tree data structure comprises a model of words having common prefix components, and an initial component of each lexical tree data structure is unique.”.....	90
2.	<i>Claim 48</i>	95
a)	Claim 48: “A speech recognition circuit according to claim 47, wherein each lexical memory stores said lexical tree data structure as an n phone model of words, where n is an integer, and said components comprise n phones.”	95
3.	<i>Claim 49</i>	97
a)	Claim 49: “A speech recognition circuit according to claim 48, wherein each lexical memory stores said lexical tree data structure as a mono phone model of words, and said lexical tree processors use said mono phone models to generate context dependant phone models of words dynamically for use in processing the speech parameters.”.....	97
X.	OPINIONS REGARDING GROUND 3	101
A.	CLAIMS 53 AND 58-59 ARE OBVIOUS OVER <i>THELEN</i> IN VIEW OF <i>BAILEY</i> AND <i>CHEN</i> IN FURTHER VIEW OF <i>JIANG</i>	101
1.	<i>Dependent Claim 53</i>	101
a)	Claim 53: “A speech recognition circuit according to claim 46, wherein each processor determines and outputs scores for words in the processing results at word recognition events.”	101
2.	<i>Dependent Claim 58</i>	103
a)	Claim 58: “A speech recognition circuit according to claim 46, wherein said processors determine and output scores in the processing results during the processing of said speech parameters.”	103
XI.	OPINIONS REGARDING GROUND 4	105
A.	CLAIMS 51 IS OBVIOUS OVER <i>THELEN</i> IN VIEW OF <i>BAILEY</i> AND <i>CHEN</i> IN FURTHER VIEW OF <i>KATZ</i>	105
1.	<i>Dependent Claim 51</i>	105
a)	Claim 51: “A speech recognition circuit according to claim 50, a including language model processor providing a language model output for modifying	

the processing results at a word recognition event
by a said processor.” 105

XII. CONCLUSION..... 108

CLAIM LISTING

CLAIM 46:

Claim 46[Pre] A speech recognition circuit comprising:

46(a) an input buffer receiving processed speech parameters;

46(b) a plurality of lexical memories containing in combination complete lexical data for word recognition, each lexical memory containing part of said complete lexical data;

46(c) a plurality of processors connected in parallel to said input buffer for processing the speech parameters in parallel, said processors being arranged in groups of processors, each group of processors being connected to a lexical memory;

46(d) a control processor controlling each processor to process said speech parameters using partial lexical data read from a respective said lexical memory;
and

46(e) a results memory storing the results of the processing of the speech parameters from said processors.

Claim 47:

A speech recognition circuit according to claim 46, wherein each lexical memory stores lexical data as at least one lexical tree data structure, each lexical

tree data structure comprises a model of words having common prefix components, and an initial component of each lexical tree data structure is unique.

Claim 48:

A speech recognition circuit according to claim 47, wherein each lexical memory stores said lexical tree data structure as an n phone model of words, where n is an integer, and said components comprise n phones.

Claim 49:

A speech recognition circuit according to claim 48, wherein each lexical memory stores said lexical tree data structure as a mono phone model of words, and said lexical tree processors use said mono phone models to generate context dependant phone models of words dynamically for use in processing the speech parameters.

Claim 50:

A speech recognition circuit according to claim 46, wherein each processor compares the processed speech parameters with the lexical data in a corresponding lexical memory to identify words as a word recognition event and sends information identifying the identified words to said results memory as the processing results.

Claim 51:

A speech recognition circuit according to claim 50, a including language model processor providing a language model output for modifying the processing results at a word recognition event by a said processor.

Claim 53:

A speech recognition circuit according to claim 51, wherein each processor determines and outputs scores for words in the processing results at word recognition events.

Claim 54:

A speech recognition circuit according to claim 46, wherein said processors process said speech parameters by comparing said speech parameters with said lexical data or data derived from lexical data in respective said lexical memories.

Claim 58:

A speech recognition circuit according to claim 46, wherein said processors determine and output scores in the processing results during the processing of said speech parameters.

Claim 59:

A speech recognition circuit according to claim 58, wherein said processors prune the processing of the speech parameters in dependence upon the determined scores.

Claim 64:

A speech recognition circuit according to claim 46, wherein said input buffer is arranged to receive said speech parameters as feature vectors.

Claim 67:

A speech recognition circuit according to claim 46, wherein said lexical memories store said lexical data as Hidden Markov Models, and each processor performs the Viterbi search algorithm using a respective part of said lexical data.

I, Christopher Schmandt, declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

1. I am over the age of 21 and am competent to make this declaration.

A. Educational Background and Professional Experience

2. 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.

3. 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.

4. For the first fifteen years of my career, my research emphasized speech recognition and speech user interfaces. I built the first conversational computer system utilizing speech recognition and synthesis (“Put That There”) starting in 1980. 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). Along with my students we built the first system for real time spoken driving directions, including speech-accessible maps of Cambridge, Massachusetts in 1987 (Back Seat Driver). 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.

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.

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

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.

II. METHODOLOGY; MATERIALS CONSIDERED

8. I have relied upon my education, knowledge, and experience with speech recognition systems more generally, as well as the other materials as discussed in this declaration in forming my opinions.

9. For this work, I have been asked to review U.S. Patent No. 7,587,319 (“’319 Patent”) (Ex. 1001) including the specification and claims, and the ’319 Patent’s prosecution history (“’319 File History”) (Ex. 1002). In developing my opinions relating to the ’319 Patent, I have considered the materials cited or discussed herein, including those itemized in the Exhibit Table below.

<u>Exhibit</u>	<u>Description</u>
Exhibit 1001	U.S. Patent No. 7,587,319 (“’319 Patent”)
Exhibit 1002	Prosecution History for the 7,587,319 Patent (“’319 File History”)
Exhibit 1004	U.S. Patent No. 6,374,219 to Jiang (“Jiang”)
Exhibit 1005	U.S. Patent No. 5,428,803 to Chen et al. (“Chen”)
Exhibit 1006	Hsiao-Wuen Hon, <i>A survey of hardware architectures designed for speech recognition</i> , Carnegie Mellon University, 1991 (“Hon”)
Exhibit 1007	Ph.D. Thesis of Mosur Ravishankar (“Ravishankar”)
Exhibit 1008	Intentionally Left Blank
Exhibit 1009	U.S. Patent No. 5,983,180 to Robinson (“Robinson”)
Exhibit 1010	U.S. Patent No. 5,036,539 to Wrench Jr., et al. (“Wrench”)
Exhibit 1011	Frederick Jelinek, <i>Statistical Methods for Speech Recognition</i> , The MIT Press, 1997 (“Jelinek”)

Exhibit 1012	Christopher Schmandt, <i>Voice Communication with Computers</i> , Van Nostrand Reinhold, 1994 (“ <i>Schmandt</i> ”)
Exhibit 1013	Lawrence Rabiner and Biing-Hwang Juang, <i>Fundamentals of Speech Recognition</i> Prentice Hall PTR, 1993 (“ <i>Rabiner</i> ”)
Exhibit 1014	Richard Klevans and Robert Rodman, <i>Voice Recognition</i> , Artech House, 1997 (“ <i>Klevans</i> ”)
Exhibit 1015	U.S. Patent No. 7,120,582 to Young et al. (“ <i>Young</i> ”)
Exhibit 1016	John Holmes and Wendy Holmes, <i>Speech Synthesis and Recognition</i> , 2 nd Edition, Taylor & Francis, 2001 (“ <i>Holmes</i> ”)
Exhibit 1017	U.S. Patent No. 4,926,488 to Nadas et al. (“ <i>Nadas</i> ”)
Exhibit 1018	U.S. Patent No. 6,182,037 to Maes (“ <i>Maes</i> ”)
Exhibit 1019	U.S. Patent Application Publication No. 2003/0220796 to Aoyama et al. (“ <i>Aoyama</i> ”)
Exhibit 1020	U.S. Patent No. 6,092,045 to Stubley et al. (“ <i>Stubley</i> ”)
Exhibit 1021	U.S. Patent No. 6,151,574 to Lee et al. (“ <i>Lee</i> ”)
Exhibit 1022	U.S. Patent No. 6,580,814 to Ittycheriah et al. (“ <i>Ittycheriah</i> ”)
Exhibit 1023	U.S. Patent No. 6,452,348 to Toyoda (“ <i>Toyoda</i> ”)
Exhibit 1024	William A. Wulf and C.G. Bell, <i>C.mmp—A multi-mini-processor</i> , Carnegie-Mellon University, 1972 (“ <i>Wulf</i> ”)
Exhibit 1025	Lee D. Erman, Richard D. Fennell, Victor R. Lesser, and D. Raj Reddy, <i>System Organizations for Speech Understanding: Implications of Network and Multiprocessor Computer Architectures for AI</i> , IEEE Transactions on Computers, Vol. C-25, No. 4, April 1976 (“ <i>Erman</i> ”)
Exhibit 1026	U.S. Patent No. 6,393,481 to Deo et al. (“ <i>Deo</i> ”)
Exhibit 1027	U.S. Patent No. 6,615,338 to Tremblay et al. (“ <i>Tremblay</i> ”)
Exhibit 1028	U.S. Patent No. 5,922,076 to Garde (“ <i>Garde</i> ”)
Exhibit 1029	Lawrence R. Rabiner, <i>A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition</i> , Proceedings of the IEEE, Vol. 77, No. 2, February 1989 (“ <i>Rabiner 89</i> ”)
Exhibit 1030	U.S. Patent No. 6,526,380 to Thelen et al. (“ <i>Thelen</i> ”)
Exhibit 1031	U.S. Patent No. 5,459,798 to Bailey et al. (“ <i>Bailey</i> ”)
Exhibit 1033	Declaration of June Ann Munford (“ <i>Munford</i> ”)
Exhibit 1034	U.S. Patent No. 6,076,056 to Huang et al. (“ <i>Huang</i> ”)
Exhibit 1035	U.S. Patent No. 4,831,550 to Katz (“ <i>Katz</i> ”)
Exhibit 1036	U.S. Patent No. 4,567,606 to Vensko et al. (“ <i>Vensko</i> ”)
Exhibit 1037	U.S. Patent No. 4,907,278 to Cecinati et al. (“ <i>Cecinati</i> ”)

Exhibit 1038	B. I. (Raj) Pawate and Peter D. Robinson, “ <i>Implementation of an HMM-Based, Speaker-Independent Speech Recognition System on the TMS320C2x and TMS320C5x</i> ”, 1996 (“ <i>Pawate</i> ”)
Exhibit 1039	U.S. Patent No. 4,386,416 to Giltner et al. (“ <i>Giltner</i> ”)
Exhibit 1040	U.S. Patent No. 4,571,700 to Emry, Jr. et al. (“ <i>Emry</i> ”)
Exhibit 1041	U.S. Patent No. 4,646,261 to Ng (“ <i>Ng</i> ”)
Exhibit 1042	U.S. Patent No. 5,150,399 to Yasuda (“ <i>Yasuda</i> ”)

10. I have considered these materials from the viewpoint of a POSITA as of the priority date of the '319 Patent. For the purposes of this declaration, I have been asked to assume that the priority date of the '319 Patent is February 4, 2002. I note that my opinions provided in this Declaration are made from the perspective of a POSITA as of this priority date of the '319 Patent, unless expressly stated otherwise. To the extent that I use any verb tense in this Declaration that is present tense (e.g., “a POSITA would understand” instead of “a POSITA would have understood”), such verb tense should be understood to be my opinion as of the '319 Patent's priority date (again, unless expressly stated otherwise). I merely use the present verb tense for ease of reading.

III. OVERVIEW AND LEGAL STANDARDS

11. In formulating my opinions, I have been instructed to apply certain legal standards. I am not a lawyer. I do not expect to offer any testimony regarding what the law is. Instead, the following sections summarize the law as I have been instructed to apply it in formulating and rendering my opinions found later in this declaration. I understand that, in an *inter partes* review (“IPR”) proceeding, patent

claims may be deemed unpatentable if it is shown that they were anticipated or rendered obvious in view of the prior art. I understand that prior art in an IPR review is limited to patents or printed publications that predate the priority date of the patent at issue. I understand that questions of claim clarity (definiteness) and enablement cannot be considered as a ground for considering the patentability of a claim in these proceedings.

A. Person of Ordinary Skill in the Art

12. I understand that the '319 Patent, the record of proceedings at the Patent Office (which I understand is called the "File History" or "Prosecution History"), and the teachings of the prior art are evaluated from the perspective of a person of ordinary skill in the art ("POSITA"). I understand that the factors considered in determining the ordinary level of skill in the art include: (i) the levels of education and experience of persons working in the field; (ii) the types of problems encountered in the field; and (iii) the sophistication of the technology. I may also consider, if available, the education level of the inventor, prior art solutions to the problems encountered in the art, and the rapidity with which innovations are made in the relevant art.

13. I understand that a person of ordinary skill in the art is not a specific real individual, but rather a hypothetical individual having the qualities reflected by the factors above. This hypothetical person has knowledge of all prior art in the

relevant field as if it were arranged on a workshop wall and takes from each reference what it would teach to a person having the skills of a POSITA.

B. Obviousness

14. I understand that a claim may be invalid under § 103(a) if the subject matter described by the claim as a whole would have been “obvious” to a hypothetical POSITA in view of a single prior art reference or in view of a combination of references at the time the claimed invention was made. Therefore, I understand that obviousness is determined from the perspective of a hypothetical POSITA. I further understand that a hypothetical POSITA is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent in suit and all analogous prior art. I understand that obviousness in an IPR review proceeding is evaluated using a preponderance of the evidence standard, which means that the claims must be more likely obvious than nonobvious.

15. I also understand that an analysis of whether a claimed invention would have been obvious should be considered in light of the scope and content of the prior art, the differences (if any) between the prior art and the claimed invention, and the level of ordinary skill in the pertinent art involved. I understand as well that a prior art reference should be viewed as a whole. I understand that in considering whether an invention for a claimed combination would have been obvious, I may assess whether there are apparent reasons to combine known elements in the prior art in the

manner claimed in view of interrelated teachings of multiple prior art references, the effects of demands known to the design community or present in the marketplace, and/or the background knowledge possessed by a POSITA. I also understand that other principles may be relied on in evaluating whether a claimed invention would have been obvious, and that these principles include the following:

- A combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results;
- When a device or technology is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or in a different one, so that if a POSITA can implement a predictable variation, the variation is likely obvious;
- If a technique has been used to improve one device, and a POSITA would have recognized that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill;
- An explicit or implicit teaching, suggestion, or motivation to combine two prior art references to form the claimed combination may demonstrate obviousness, but proof of obviousness does not depend on or require showing a teaching, suggestion, or motivation to combine;

- Market demand, rather than scientific literature, can drive design trends and may show obviousness;
- In determining whether the subject matter of a patent claim would have been obvious, neither the particular motivation nor the avowed purpose of the named inventor controls;
- One of the ways in which a patent's subject can be proved obvious is by noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims;
- Any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed;
- "Common sense" teaches that familiar items may have obvious uses beyond their primary purposes, and in many cases a POSITA will be able to fit the teachings of multiple patents together like pieces of a puzzle;
- A POSITA is also a person of ordinary creativity, and is not an automaton;
- A patent claim can be proved obvious by showing that the claimed combination of elements was "obvious to try," particularly when there

is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions such that a POSITA would have had good reason to pursue the known options within his or her technical grasp; and

- One should not use hindsight in evaluating whether a claimed invention would have been obvious.

16. I also understand that an obviousness determination includes the consideration of various factors such as: (1) the scope and content of the prior art, (2) the differences between the prior art and the asserted claims, (3) the level of ordinary skill in the pertinent art, and (4) the existence of secondary considerations such as commercial success, long-felt but unresolved needs, failure of others, and so forth.

17. I am informed that it is improper to combine references where the references teach away from their combination. I am informed that a reference may be said to teach away when a person of ordinary skill in the relevant art, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the patent applicant. In general, a reference will teach away if it suggests that the line of development flowing from the reference's disclosure is unlikely to be productive of the result sought by the patentee. I am informed that a reference teaches away, for

example, if (1) the combination would produce a seemingly inoperative device, or (2) the references leave the impression that the product would not have the property sought by the patentee. I also am informed, however, that a reference does not teach away if it merely expresses a general preference for an alternative invention but does not criticize, discredit, or otherwise discourage investigation into the invention claimed.

18. I am informed that even if a *prima facie* case of obviousness is established, the final determination of obviousness must also consider “secondary considerations” if presented. In most instances, the patentee raises these secondary considerations of non-obviousness. In that context, the patentee argues an invention would not have been obvious in view of these considerations, which include: (a) commercial success of a product due to the merits of the claimed invention; (b) a long-felt, but unsatisfied need for the invention; (c) failure of others to find the solution provided by the claimed invention; (d) deliberate copying of the invention by others; (e) unexpected results achieved by the invention; (f) praise of the invention by others skilled in the art; (g) lack of independent simultaneous invention within a comparatively short space of time; (h) teaching away from the invention in the prior art.

19. I am further informed that secondary considerations evidence is only relevant if the offering party establishes a connection, or nexus, between the

evidence and the claimed invention. The nexus cannot be based on prior art features.

The establishment of a nexus is a question of fact. While I understand that the Patent Owner here has not offered any secondary considerations at this time, I will supplement my opinions in the event that the Patent Owner raises secondary considerations during the course of this proceeding.

20. Finally, I understand that obviousness in an IPR must be proven by a preponderance of the evidence.

C. Analogous Art

21. I have been informed that for a prior art reference to be proper for use in an obviousness analysis, the reference must be “analogous art” to the claimed invention. I have been informed that a reference is analogous art to the claimed invention if: (1) the reference is from the same field of endeavor as the claimed invention (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor (even if it is not in the same field of endeavor as the claimed invention). In order for a reference to be “reasonably pertinent” to the problem, it must logically have commended itself to an inventor’s attention in considering his problem. In determining whether a reference is reasonably pertinent, one should consider the problem faced by the inventor, as reflected either explicitly or implicitly, in the specification. I believe that all of the references that my opinions in this IPR are based upon are well within the range of

references a person having ordinary skill in the art would consult to address the type of problems described in the Challenged Claims.

D. Claim Construction

22. I understand that the claim terms in this IPR will be construed according to their ordinary and customary meaning as understood in light of the claim language, the patent's description, and the prosecution history viewed from the perspective of a POSITA. I further understand that where a patent defines claim language, the definition in the patent controls, even if there are other definitions that might be understood by those working in the art.

IV. LEVEL OF A PERSON OF ORDINARY SKILL

23. Based on my review and analysis of the '319 Patent, the prior art cited herein, and the ordinary skill factors described in this section, a POSITA in the field of the '319 Patent at the time of the earliest possible priority date (February 4, 2002) would have been knowledgeable regarding the field of speech recognition technology. In my experience working in this field, most workers of ordinary skill in the art as of the earliest possible priority date of February 4, 2002 would have had a master's degree in computer engineering, computer science, electrical engineering, or a related field, with at least two years of experience in the field of speech recognition, or a bachelor's degree in the same fields with at least four years of experience in the field of speech recognition. *See, e.g., the '319 Patent (Ex. 1001) at*

1:18-58 (describing the “Field of the Invention” and providing a “Description of the Related Art”). A person with less relevant practical experience but with additional education can also qualify as a POSITA in the field of the ’319 Patent provided the additional education focused on speech recognition technology. When I refer to the understanding of a POSITA, I am referring to the understanding of such a person as of February 4, 2002.

24. As of February 4, 2002, I had more than ordinary skill in the art. I am, however, familiar with the skills and knowledge possessed by those I would have considered to be of ordinary skill in the art as of that date. For example, I knew people working in the field while I built systems employing speech recognition in handheld mobile devices (ComMotion 1999, Nomadic Radio 2000, and Impromptu 2001, for example).

25. My opinions provided in this declaration would not change in view of minor modifications to this level of ordinary skill.

V. OVERVIEW OF THE TECHNOLOGY

26. I was asked to briefly summarize the background of the technology from the standpoint of a POSITA prior to February 4, 2002.

A. Speech Recognition

27. Speech recognition using computer systems was known and widely used well before the priority date of the ’319 Patent. Some of the earliest speech

recognition systems were developed in the 1950s and 1960s. *Rabiner* (Ex. 1013), 37-38. By the 1980s, speech recognition systems using Hidden Markov Models were well known and “widely applied in virtually every speech-recognition research laboratory in the world.” *Rabiner*, 39. In general, speech recognition refers to “the technologies used to allow computers to recognize the words in human speech.” *Schmandt* (Ex. 1012), 45. Indeed, speech was known to be used in “a variety of computing environments based on its role in human communication. Speech can be used as a method of *interacting* with a computer to place requests or receive warnings and notices. Voice can also be used as the underlying *data* itself[.]” *Schmandt*, 11. Further, at least as early as 1994 (publication date of *Schmandt*), “[d]esktop workstations can already support both these speech functions.” *Id.*

28. One of the initial processes in a speech recognition system is “parameter extraction,” which entails “methods of extracting information from speech waveforms. Parameter or feature extraction consists of preprocessing an electrical signal to transform it into a usable digital form, applying algorithms to extract only speaker-related information from the signal[.]” *Klevans* (Ex. 1014), 12. Such extraction involves “digital signal processing (DSP) methods that are common to all computer speech systems.” *Klevans*, 12-13. *Klevans* teaches that “the sound wave created by an individual’s speech is transduced into an analog electrical signal via a microphone. The electrical signal is sampled and quantized, resulting in a

digital representation of the analog signal.” *Klevans*, 13. Following such analog to digital processing, “conditioning” may be performed on the signal, which includes bandpass filtering to remove additive noise and spectral flattening to compensate for “the effect of the vocal tract on the excitation signal created by the vibrating vocal folds.” *Id.*

29. In general, it was known that speech recognizer systems included three basic components: (1) “speech **representation** that is computationally efficient for pattern matching” such as “Hidden Markov Models” that “are a statistical representation based on the various ways words or phonemes may be pronounced.” *Schmandt*, 45-46. I discuss Hidden Markov Models in more detail at Section V.D below. (2) A “set of **templates** or **models**, which are descriptions of each word to be recognized, in the representation of speech used by the recognizer. The templates describe the words in the recognizer’s vocabulary, i.e., those words that the recognizer can identify. They are reference models against which an input can be compared to determine what was spoken.” *Schmandt*, 46. (3) A “**pattern matching** algorithm to determine which template is most similar to a specimen of speech input. This element of the speech recognizer must determine word boundaries, locate the most similar template, and decide whether the difference between the input and the selected template is minor enough to accept the word.” *Id.* In general, a speech recognizer captures incoming speech and converts it “to the chosen internal

representation (see Figure 7.1). The pattern matcher then selects the template that most closely matches the input or rejects the utterance if no template is a close enough match.” *Id.* The functional elements of an exemplary speech recognizer are depicted below in Figure 7.1.

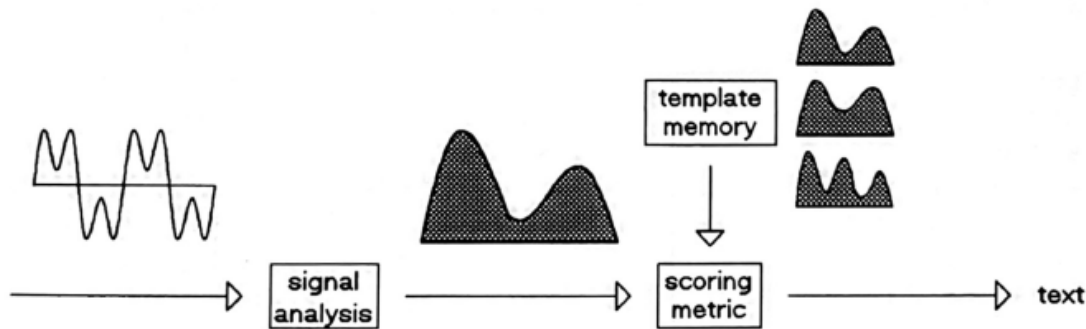


Figure 7.1. The functional elements of a speech recognizer. The user’s speech is digitized and converted to the recognizer’s internal representation. The captured speech is then compared with the words stored in the recognizer’s template memory. The pattern matching algorithm then determines which template, if any, is the closest match.

Schmandt, 46 (“Figure 7.1”).

30. When discussing speech recognition, one important topic is phonemes, which are speech sounds that form the words in a language.

The **phonemic** layer is concerned with the string of phonemes of which the words are composed. Phonemes are the speech sounds that form the words of any language. Phonemes include all the sounds associated with vowels and consonants. A grunt, growl, hiss, or gargling sound is not a phoneme in English, so it cannot be part of a word; such sounds are not referred to as speech. At the phoneme layer, while talking we are either continuously producing speech sounds or are silent. We are not silent at word boundaries; the phonemes all run together.

Schmandt, 19. *Schmandt* further discloses that “[p]honemes are important small units, several of which make up most syllables.” *Schmandt*, 23. Further, *Schmandt* discloses that a “**phoneme** is a unit of speech, the set of which defines all the sounds from which words can be constructed in a particular language. There is at least one pair of words in a language for which replacing one phoneme with another will change what is spoken into a different word. Of course, not every combination of phonemes results in a word; many combinations are nonsense.” *Schmandt*, 23-24.

31. It was also known to utilize spectrograms for “characterizing the speech signal and representing the information associated with the sounds” “via a spectral representation.” *Rabiner*, 50. *Rabiner* teaches that “[p]erhaps the most popular representation of this type is the sound spectrogram in which a three-dimensional representation of the speech intensity, in different frequency bands, over time is portrayed.” *Id.* *Rabiner* teaches that the “spectral intensity at each point in time is indicated by the intensity (darkness) of the plot at a particular analysis frequency.” *Id.*

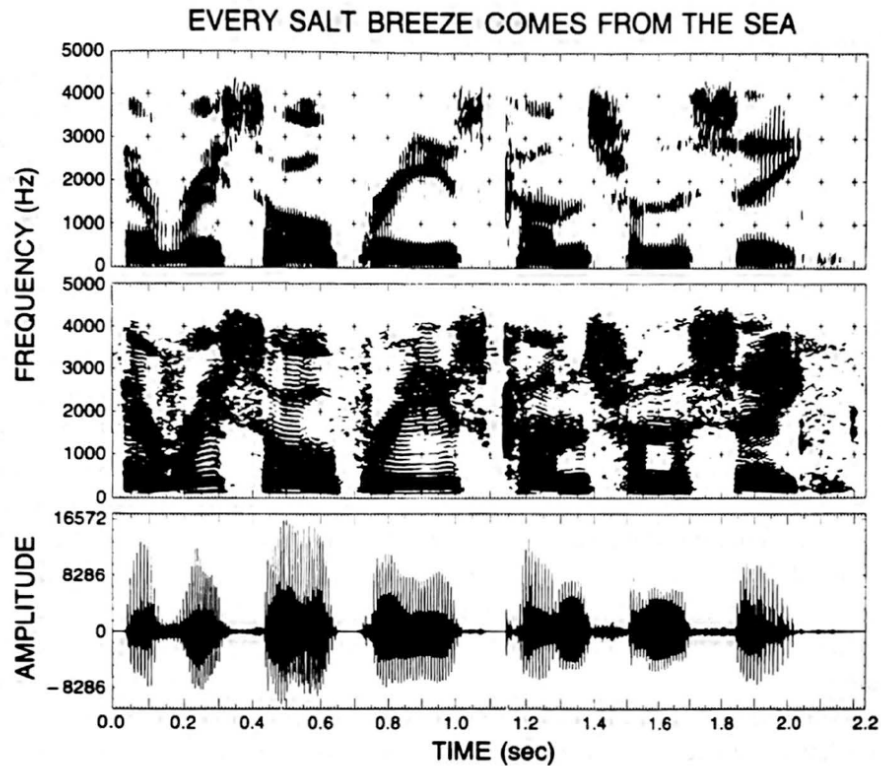


Figure 2.8 Wideband and narrowband spectrograms and speech amplitude for the utterance “Every salt breeze comes from the sea.”

Rabiner, 50 (“Figure 2.8”).

32. One speech recognition technique known well before the priority date of the '319 Patent was the statistical pattern-recognition approach to speech recognition. *Rabiner* (published in 1993) teaches the statistical pattern-recognition technique includes four main steps: feature measurement, pattern training, pattern classification, and decision logic. *Rabiner*, 82. *Rabiner* teaches the first step is “[f]eature measurement, in which a sequence of measurements is made on the input signal to define the ‘test pattern.’ For speech signals the feature measurements are usually the output of some type of spectral analysis technique, such as a filter bank

analyzer, a linear predictive coding [LPC] analysis, or a discrete Fourier transform (DFT) analysis.” *Id.* The second step, pattern training, involves creating reference patterns or models that characterize “the statistics of the features of the reference pattern.” *Id.* During the pattern classification step, an “unknown test pattern is compared with each (sound) class reference pattern and a measure of similarity (distance) between the test pattern and each reference pattern is computed.” *Id.* *Rabiner* further teaches that to “compare speech patterns (which consist of a sequence of spectral vectors), we require” “a local distance measure, in which local distance is defined as the spectral ‘distance’ between two well-defined spectral vectors[.]” *Id.* The fourth step is decision logic, “in which the reference pattern similarity scores are used to decide which reference pattern (or possibly which sequence of reference patterns) best matches the unknown test pattern.” *Id.*

33. It was also known to use lexical trees to organize vocabulary before the priority date of the ’319 Patent. For example, *Young* teaches using a “lexical tree 500 that is initialized before processing begins. The lexical tree 500 represents the active vocabulary 230 based on the phonetic relationships between words in the vocabulary.” *Young* (Ex. 1015), 11:37-41. *Young* further teaches that the “lexical tree includes a root node 505 that represents new words entering the lexical tree. From the root node 505, the tree expands to a group 510 of nodes that correspond to phonemes with which words start.” *Young*, 11:41-44. *Young* teaches that “[e]ach

node in the group 510 represents a phoneme that appears at the beginning of one or more words” using an example where “a node 610 corresponds to all words in the vocabulary that start with the phoneme ‘H’. Together, the nodes in the group 510 include representations of the starting phoneme of every word in the vocabulary.”

Young, 11:47-53. Figs. 5 and 6 of *Young* are depicted below.

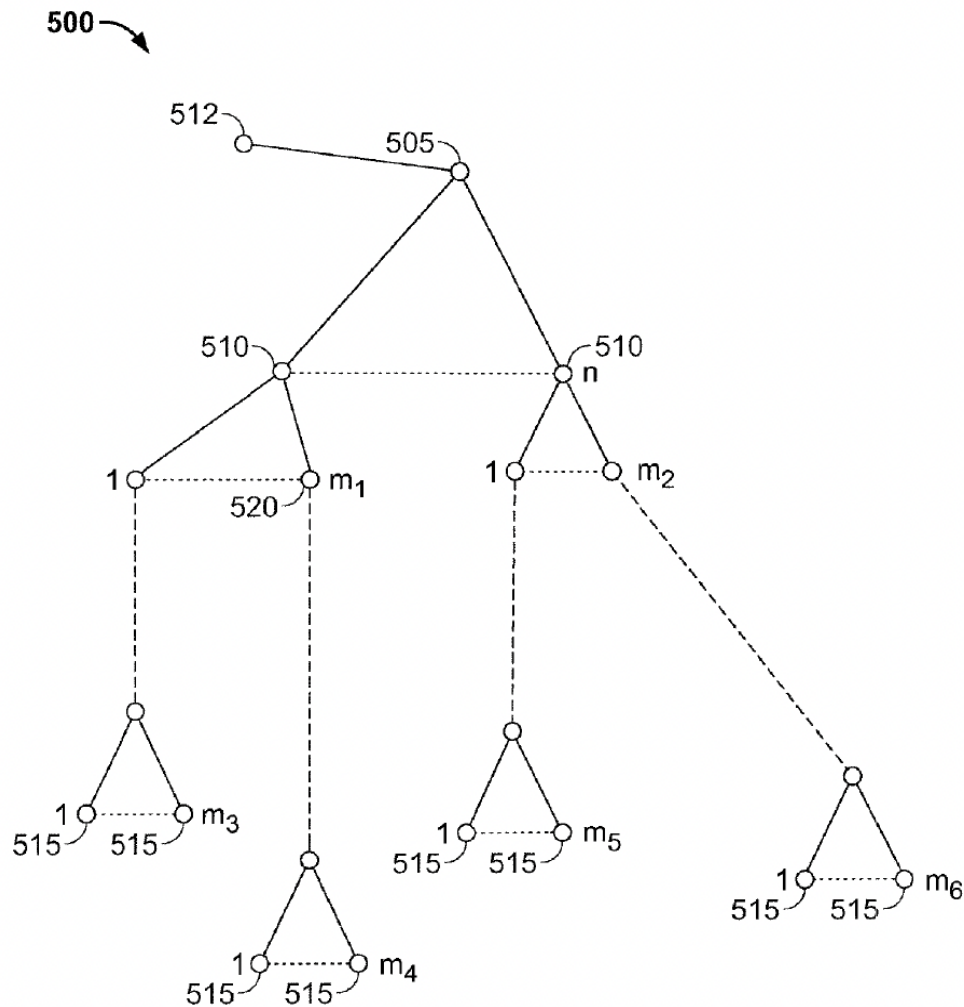


FIG. 5
(Prior Art)

Young, Fig. 5.

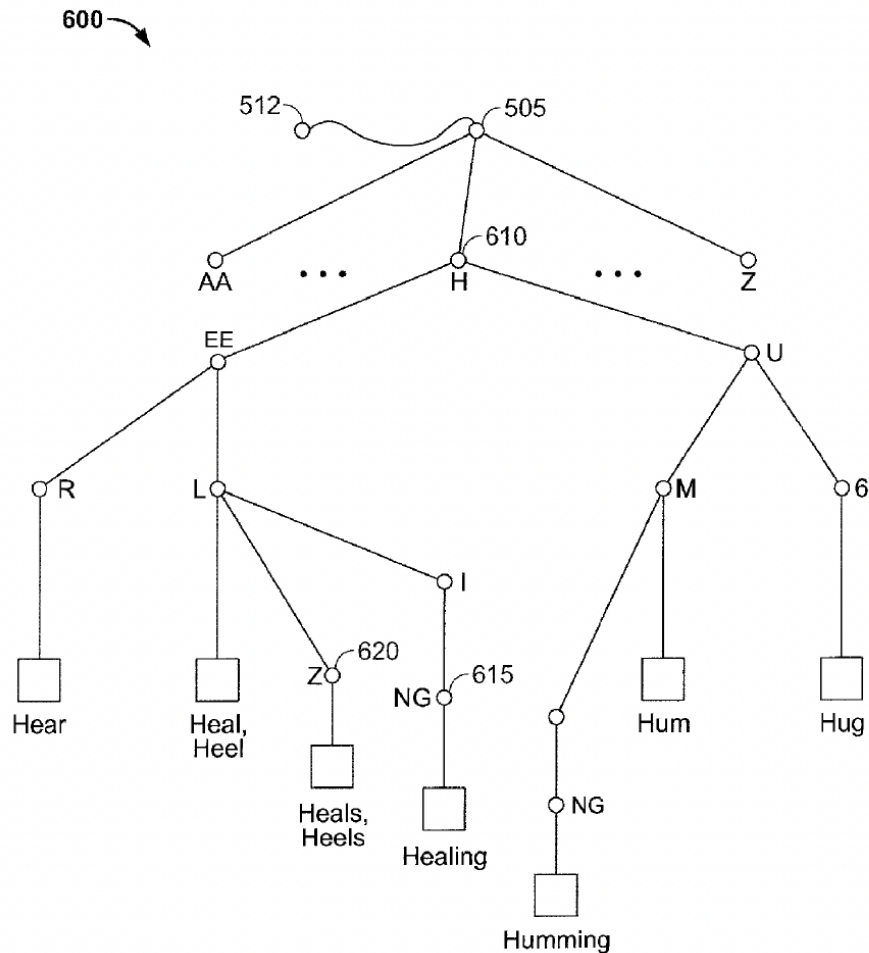


FIG. 6
(Prior Art)

Young, Fig. 6.

B. Feature Vectors

34. Using feature vectors in speech recognition systems was well known before the priority date of the '319 Patent. For example, *Holmes* teaches that in “calculating a distance between two words it is usual to derive a short-term distance that is local to corresponding parts of the words, and to integrate this distance over the entire word duration. Parameters representing the acoustic signal must be derived

over some span of time, during which the properties are assumed not to change much.” *Holmes* (Ex. 1016), 13. *Holmes* further teaches that in “one such span of time the measurements can be stored as a set of numbers, or **feature vector**, which may be regarded as representing a point in multi-dimensional space. The properties of a whole word can then be described as a succession of feature vectors (often referred to as **frames**), each representing a time slice[.]” *Id.*

35. In general, feature vectors are representations of speech where each dimension of the vector corresponds to “some prescribed acoustic feature.” *Nadas* (Ex. 1017), 1:15-23 (reproduced below). For example, *Nadas* teaches an n-dimensional space where dimensions correspond to acoustic features.

In speech processing, speech is generally represented by an n-dimensional space in which each dimension corresponds to some prescribed acoustic feature. For example, each component may represent a amplitude of energy in a respective frequency band. For a given time interval of speech, each component will have a respective amplitude. **Taken together, the n amplitudes for the given time interval represent an n-component vector in the n-dimensional space.**

Id. (emphasis added). Additionally, *Nadas* teaches that “[w]hen an unknown speech input is uttered, for each time interval, a value is measured or computed for each of the n components, where each component is referred to as a feature. The values of all of the features are consolidated to form an n-component feature vector for a time interval.” *Nadas*, 1:32-37. *Nadas* teaches that “the components of the feature vectors

(and prototype vectors)” may “correspond to well-known (1) Cepstral coefficients, (2) linear predictive coding coefficients, or (3) frequency band-related characteristics.” *Nadas*, 10:15-19. *Nadas* further teaches that a “speech type of event, such as a word or a phoneme, is characterized by a sequence of feature vectors in the time period over which the speech event was produced.” *Nadas*, 1:50-52. Further, *Jelinek* teaches that the “current continuous speech recognizers use as acoustic processor outputs vectors of cepstral coefficients [3].” *Jelinek* (Ex. 1011), 61.

36. *Maes* teaches using a well-understood acoustic front-end to extract feature vectors. *Maes* (Ex. 1018), 5:6-10 (“[A]n acoustic front-end 110 which is, itself, well-understood in the art, is used to sample utterances in an overlapping fashion and to extract feature vectors 120 therefrom, preferably as MEL cepstra, delta and delta-delta coefficients along with normalized log-energies.”). In another example, *Aoyama* teaches a feature extraction unit that analyzes audio input and extracts information to generate a feature vector. In this example, *Aoyama* teaches performing a Mel Frequency¹ Cepstrum Coefficient (MFCC) analysis and extracting

¹ I note that the Mel scale is a frequency representation which approximates the frequency response of the human auditory system, i.e., how humans hear frequency.

a feature vector utilizing a linear predictive coefficient (LPC) or Cepstrum coefficient. *Aoyama* (Ex. 1019), [0138]-[0139], [0150]-[0152], [0156]-[0160].

37. *Stubley* teaches converting speech samples into feature vectors is advantageous because feature vectors are more compact and provide better speech recognition performance than the speech samples themselves.

The purpose of the signal processing component is to convert the utterance, which is a series of samples of speech, into a more compact series of feature vectors. Besides being more compact, the feature vectors provide better speech recognition performance since they tend to remove many speaker-dependent effects and improve the robustness of the system to noise.

Stubley (Ex. 1020) at 11:30-36.

38. *Lee* teaches input speech is analyzed and represented by feature vectors that correspond to phonemes that are characterized by hidden Markov models (HMMs).

[I]nput speech is analyzed in signal frames, represented by feature vectors corresponding to phonemes making up individual words. The phonemes are characterized by hidden Markov models (HMMs), and a Viterbi algorithm is used to identify a sequence of HMMs which best matches, in a maximum likelihood sense, a respective concatenation of phonemes corresponding to an unknown, spoken utterance.

Lee (Ex. 1021) at 1:24-31 (emphasis added).

C. Acoustic Models

39. It was also known before the priority date of the '319 Patent to use acoustic models in speech recognition systems. For example, *Jelinek* teaches “acoustic modeling” includes “a statistical *acoustic model* of the speaker’s interaction with the acoustic processor.” *Jelinek*, 21. *Jelinek* teaches one type of acoustic model is a “*phonetic acoustic model*” that “is based on an intuitive linguistic concept.” *Jelinek*, 30. For example, *Jelinek* teaches creating a phonetic dictionary for vocabulary and creating a correspondence between words and sequences of “symbols from a predetermined phonetic alphabet[.]” *Id.*

40. In another example, *Schmandt* teaches using a set of templates or models as a representation of speech used by the speech recognizer.

A set of **templates** or **models**, which are descriptions of each word to be recognized, in the representation of speech used by the recognizer. The templates describe the words in the recognizer’s vocabulary, i.e., those words that the recognizer can identify. They are reference models against which an input can be compared to determine what was spoken.

Schmandt at 46.

41. A further example of using acoustic models is taught by *Stubley*. *Stubley* teaches “comparing a series of observations representing unknown speech, to stored models representing known speech[.]” *Stubley*, 1:5-10, 1:60-62. *Stubley* teaches that “the models comprise representations of elements of speech.” *Stubley* at

2:51-52. Further, *Stubbley* teaches in one example “the models comprise groups of representations of phonemes. This enables the models to be stored and used more efficiently.” *Stubbley*, 2:40-42.

42. In another example, *Ittycheriah* teaches storing a number of reference models in a memory accessible to a processor. *Ittycheriah* (Ex. 1022) at 3:34-35 (“A reference model store 112 is connected to the model construction module 110 for storing the L reference models[.]”), *Ittycheriah*, 4:30-32 (“Further, the reference model store 112 and the rank/distance store 120 may be electronic computer read/write memory or any other suitable memory.”).

43. In a further example, *Rabiner 89* teaches characterizing signals in terms of signal models. *Rabiner 89* teaches a “signal model can provide the basis for a theoretical description of a signal processing system which can be used to process the signal so as to provide a desired output.” *Rabiner 89*, 1 (Ex. 1029). Further, *Rabiner 89* teaches that such signal models include “statistical models in which one tries to characterize only the statistical properties of the signal[.]” with one example being a “hidden Markov model (HMM)” *Id.* Thus, one particular type of acoustic model is a Hidden Markov Model, which I discuss in detail below.

D. Hidden Markov Models

44. Before the priority date of the '319 Patent, Hidden Markov Models were used as a technique to “make pattern matching more powerful for dealing with

variations in speech patterns.” *Schmandt* at 54. Indeed, *Jelinek* teaches that the “usual acoustic model employed in speech recognizers” is “the *hidden Markov model*[.]” *Jelinek*, 21. In general, a Hidden Markov Model (HMM) is “a two-stage probabilistic process that can be used as a powerful representation for speech. A Hidden Markov Model is a well-behaved mathematical construct[.]” *Schmandt*, 57. Further, it was known that HMMs consisted of states. “An HMM consists of a number of internal states; the model passes from an initial state to the final state as a step-by-step process generating an observable output at each step (state transition). For example, the states may correspond to phonemes in a word with the observable output corresponding to the presence or absence of a number of acoustic features.” *Schmandt* at 57-58. Additionally, “[a]t each step, the model can either move to a new state or stay in the current one. The model is ‘hidden’ in that we cannot observe the state directly but only its output. From the series of observable outputs, we can attempt to guess when the model was in each state.” *Schmandt*, 58. “Alternatively, we can say whether some series of outputs was likely to have been generated by a particular HMM.” *Id.*

45. *Schmandt* teaches a “pedagogic example” of a “box-and-bowl setup” to explain Hidden Markov Models.

For example, consider the arrangement shown in Figure 7.10, in which a box and a bowl are each full of black balls and white balls. The box has many more black balls than white while white balls dominate the bowl. Starting with

the box, we remove one ball from it at random and pass it to an observer in another room who cannot see what is being done. Then a normal six-sided die is tossed. If the result is less than four, then the source of the next ball is the bowl. If the die is greater than three, then the next ball will be selected from the box. The cycle is then repeated. Whenever we select from the box, a die throw of less than four shifts attention to the bowl. Once we start selecting from the bowl, we continue selecting from it unless the die shows a one, at which point we are finished. This model is likely to spend a majority of its cycles selecting from the bowl since a state transition from the box has a probability of $1/2$ of shifting to the bowl, but a bowl transition has a probability of only $1/6$ of terminating and $5/6$ of continuing with the bowl. We should expect this arrangement to initially present more black balls than white to the observer (while it is in the box state) and then to produce more white balls. We should also expect more white balls overall. But keep in mind that this is a probabilistic process: Not all black balls come from the box and not all selections come from the bowl.

Id. Schmandt further teaches that the “box-and-bowl setup is a Hidden Markov Model; it is characterized by a number of *states* and associated probabilities for each *transition* from any state. The box and bowl are each a state; the rules about throwing the die define the transition properties.” Schmandt, 58-59. Schmandt also teaches that “[p]robability is expressed as a number between zero and one. If an event occurs with probability $1/2$, we would expect it to happen once in two trials. A probability of $1/6$ implies that we can expect the event once out of six trials. A smaller probability indicates that an event is less likely to occur.” Schmandt, 58.

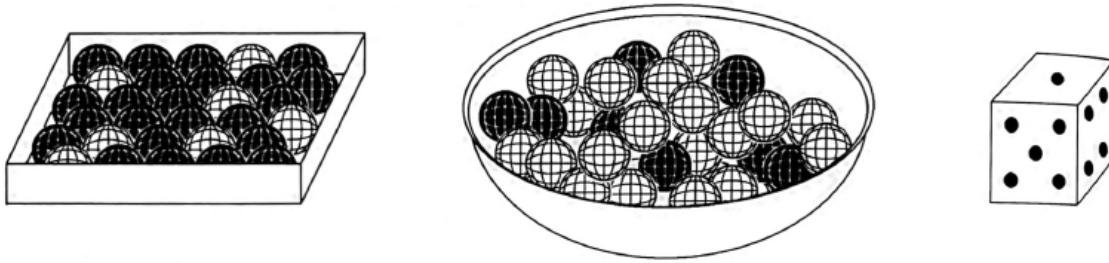


Figure 7.10. A box and a bowl full of colored balls illustrate a Hidden Markov Model.

Schmandt, 59 (“Figure 7.10”).

46. Further, *Schmandt* teaches a “more general HMM” as depicted in Figure 7.11 “A Markov Model” below. *Schmandt* teaches that “in each state $S(n)$ there is some probability $P(n,n)$ of remaining in that state and some probability $P(n, n+1)$ of transitioning to the next state.” *Schmandt*, 59. Additionally, *Schmandt* discloses that “[i]n some models it may be possible to bypass a state or a series of states as indicated by the dashed arc labeled P_{13} in the figure. The sum of the probabilities of all the arcs leaving a state is one; the model accounts for every possible path through its states.” *Schmandt*, 59.

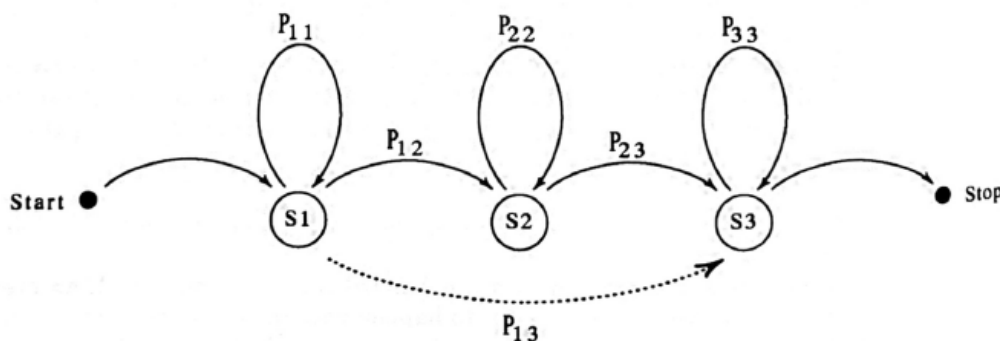


Figure 7.11. A Markov Model.

Schmandt, 59 (“Figure 7.11”).

47. *Schmandt* teaches that the example of the box-and-bowl setup and the more general HMM model apply to speech recognition because an HMM's internal states can represent acoustic segments.

An HMM can be used to represent a word with internal states representing characteristic acoustic segments, possibly phonemes or allophones. The output of a state is a frame or vector of acoustic parameters or features; this output is probabilistic to allow for variability in pronunciation and hence differences in the acoustic representation. The duration of an acoustic segment is a function of the number of steps in which the model is in the state corresponding to the segment. Staying in the same state, i.e. lengthening a phone, depends on the probability associated with the transition from that state to itself (P_{11} in Figure 7.11). Arcs such as P_{13} may be included to indicate that an intermediate state **S2** is optional during pronunciation, such as the second syllable in 'chocolate.'

In terms of the basic recognizer components, the 'templates' consist of a set of HMMs with one HMM associated with every word. The acoustic representation of an input specimen of speech is *not* a collection of states but rather a set of acoustic parameters corresponding to the possible observed sequences from the template HMMs. For the HMM-based recognizer, classification consists of determining which HMM has the highest probability of producing the observed acoustic sequence. The Viterbi algorithm is a commonly used method of solving this classification problem[.]

Schmandt, 59-60 (emphasis added).

48. *Schmandt* further teaches that "Hidden Markov Models are a powerful representation of speech and consequently are used in many speech recognition systems currently under development." *Schmandt*, 60. *Schmandt* also teaches that

HMMs can be used to “represent not only the phones of each word but also the probabilities of transitioning from one word to another.” *Id.*

49. *Rabiner* teaches that an observation vector in an HMM can be “LPC-derived weighted cepstral coefficients and weighted cepstral derivatives[.]” *Rabiner*, 231. For example, *Rabiner* teaches that the observation vector in the HMM can be a “38-component vector consisting of 12 cepstral coefficients, 12 delta cepstrum coefficients, 12 delta-delta cepstral coefficients, delta log energy, and delta-delta log energy.” *Rabiner*, 280; *see also Rabiner* at 301-302 (teaching spectral vectors with 38 components).

50. Another example of using an HMM for speech recognition well before the priority date of the '319 Patent is taught by *Nadas*. *Nadas* (issued in 1990) teaches an acoustic coprocessor 200 for speech recognition and “contemplates a probabilistic embodiment in which each prototype—rather than identifying a vector—corresponds to a probabilistic finite state machine PFSM (or Markov model).” *Nadas* (Ex. 1017), 9:68-10:4.

51. In another example, *Aoyama* teaches using an HMM as the acoustic model for speech recognition and said HMM model is stored in an acoustic model memory unit.

The matching unit 92, referring to the acoustic model memory unit 93, the dictionary memory unit 94 and the grammar memory unit 95 in utilizing the feature vector from the feature extraction unit 91 as occasion

demands, speech recognizes the voice (input speech) entered into the microphone 51 based on such as the Hidden Markov model (HMM) law.

More specifically, **the acoustic model memory unit 93 memorizes acoustic model** (e.g., including the standard pattern to be used in DP (Dynamic Programming) matching, other than HMM) showing the acoustic feature on the sub-words such as phoneme, syllable, and phoneme series in the audio language for identifying the speech. **Here, since the speech recognition is conducted based on the Hidden Markov Model law, the HMM will be used as the acoustic model.**

Aoyama at [0139]-[0140] (emphasis added); *see also Aoyama* at [0150]-[0151].

52. In yet another example, *Stubley* teaches using HMMs as acoustic models.

Advantageously, the models are represented as finite state machines with probability distribution functions attached. Continuously varying signals can be modeled as a series of stationary states, with a probability function at each state, indicating whether the signal moves to the next state or stays at the given state, after a given time interval. An example is the Hidden Markov Model (HMM). This enables the models to contain a representation of patterns with varying rates of change, yet a common underlying pattern.

Stubley, 2:29-37; *see also Stubley*, 7:23-35, 12:29-44 (“**The modelling paradigm used here is to construct models using hidden Markov models (HMMs)[.]**”) (emphasis added).

53. In another example, *Lee* teaches analyzing input speech represented by feature vectors that correspond to phonemes of words. *Lee* further teaches characterizing phonemes by Hidden Markov Models.

[I]nput speech is analyzed in signal frames, represented by feature vectors corresponding to phonemes making up individual words. The phonemes are characterized by hidden Markov models (HMMs), and a Viterbi algorithm is used to identify a sequence of HMMs which best matches, in a maximum likelihood sense, a respective concatenation of phonemes corresponding to an unknown, spoken utterance.

It is well known that each HMM comprises model parameters, e.g., mixture components which are characterized by Gaussian distributions. **In a learning phase in a speech recognition system, the HMMs are adapted to input speech by a user to adjust to the particular speech characteristics of the user, thereby increasing accuracy of the speech recognition.**

Lee, 1:24-38 (emphasis added).

54. In yet another example, *Toyoda* teaches a voice recognition module (sensor signal processor 31) that uses an HMM. *Toyoda* (Ex. 1023), 7:48-50. Furthermore, *Toyoda* teaches referencing an acoustic model memory that stores an acoustic model representing phonemes while a matching unit 42 recognizes voice input.

Using the feature parameters from the feature parameter extractor 41, **the matching unit 42 recognizes the voice input to the microphone 9 based on the continuous HMM model while referencing an acoustic model**

memory 43, a dictionary memory 44, and a grammar memory 45 as necessary.

The acoustic model memory 43 stores an acoustic model that represents an acoustic feature such as phonemes and syllables in a voice to be recognized. Since voice recognition is here carried out using the continuous HMM, an HMM is employed.

Toyoda, 7:61-8:5 (emphasis added).

E. Prior Art Speech Recognition Systems

55. Numerous speech recognition systems were well known before the priority date of the '319 Patent. For example, IBM's Tangora classifies acoustic input that "is then matched to words using discrete Hidden Markov Models of phoneme realization. A linguistic decoder then classifies the output of the acoustic front end producing a word for output." *Schmandt*, 64.

56. Other well-known speech recognition system were CMU's Sphinx and Sphinx II. Sphinx "operates in nearly real time using hardware multiprocessor peripherals to aid in the search process." *Id.* Further, "Sphinx makes extensive use of Hidden Markov Models: each phone is represented by an HMM, each word is a network of phones, and the language is a network of words." *Schmandt*, 65. Further,

*Ravishankar*² describes the Sphinx-II speech recognition circuit as including HMMs as “a set of states connected by transitions” expressly characterized as an “acoustic model,” explaining that HMMs were used as acoustic models in the art prior to the ’319 Patent. *Ravishankar*, 28-29, 38, 40 (“Acoustic modelling in Sphinx-II is based on hidden Markov models...”). *Ravishankar* teaches that “[o]rganizing the HMMs to be searched as a phonetic tree instead of the flat structure of independent linear HMM sequences for each word is probably the most often cited improvement in search techniques[.]” *Ravishankar*, 33.

57. Yet another well-known speech recognition system was MIT’s SUMMIT. SUMMIT is a “phoneme-based connected speech recognizer[.]” *Schmandt*, 65. Transformed speech is segmented by “finding acoustic landmarks in the signal. Regions between the landmarks are grouped into segments on the basis of similarity. A set of phonetic classifiers assigns probabilistic phonetic labels to each of these portions of the speech signal.” *Id.* Further, “[w]ords are represented by

² I have reviewed the Declaration of June Ann Munford (Ex. 1033 (“*Munford*”)). *Munford* attests to *Ravishankar*’s public availability date and that Ex. 1007 (a scan done by June Ann Munford of a physical copy of *Ravishankar* from Carnegie Mellon University) is a true and correct copy of *Ravishankar*. *Munford* (Ex. 1033) at 1-15.

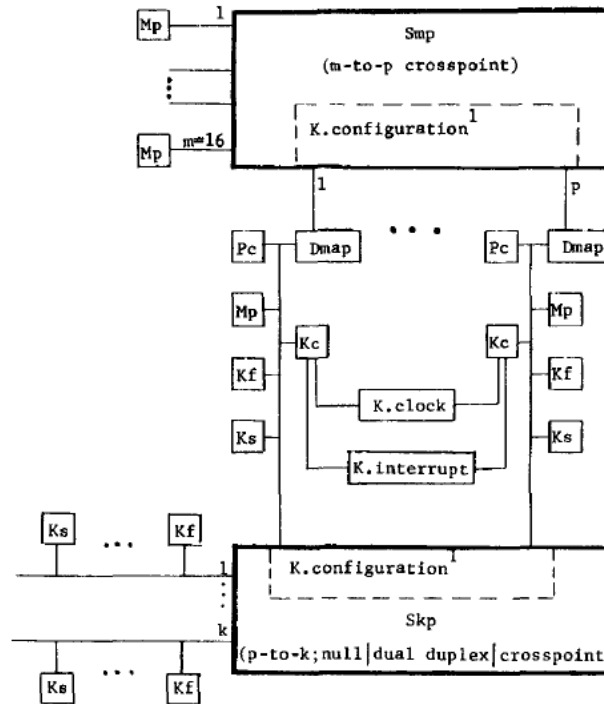
a network of phonemes; several networks may be used to encode alternate pronunciations. A pattern matching algorithm similar to dynamic programming matches the phoneme labels of the input against the word networks, deducing word boundaries in the process.” *Id.*

58. Many speech recognition systems prior to the '319 Patent recognized the advantages of using a computing platform having clusters of processors, employing clustered architectures for power, flexibility, scalability, and cost management. For example, *Hon* describes the MARS system, a speech recognition circuit having hardware organized in three levels: System, Cluster, and Processing Element. *Hon*, 19-20 (Ex. 1006). Each of the processing elements was a “16-bit custom designed microprogrammed VLSI processor” that communicated via a crossbar switch to form a pipeline. *Hon*, 19. Fifteen PEs connected via a crossbar formed a “cluster,” with the overall MARS system having up to 256 clusters. *Id.* *Hon* articulates how this clustered architecture provided advantages over other speech recognition computing platforms, stating “In contrast to the SRI-Berkeley’s search machine, MARS is scalable because it can be expanded by adding more clusters.” *Id.*

59. Other examples of speech recognition circuits employing clusters of processors can be found throughout the prior art. *Wulf* teaches a multi-processor computer system developed at CMU dubbed C.mmp using PDP-11 minicomputer

processors to recognize speech. *Wulf*, 1 (Ex. 1024). As seen in Figure 1 below,

C.mmp's included a plurality of processors Pc clustered to shared memory Mp:



where: Pc/central processor; Mp/primary memory; T/terminals;
 Ks/slow device control (e.g., for Teletype);
 Kf/fast device control (e.g., for disk);
 Kc/control for clock, timer, interprocessor communication

¹ Both switches have static configuration control by manual and program control

Figure 1—Proposed CMU multiminiprocessor computer/C.mmp

Wulf, 3-4, Fig. 1. *Erman* describes both the C.mmp system explained in *Wulf* and the HEARSAY speech recognition system that used a multiprocessor system to parallelize speech recognition processes using 16 processors clustered into two groups of 8 processors each administered by a single scheduler. *Erman*, 1-2, 5 (Ex. 1025). U.S. Patent No. 6,393,481 to Deo et al. teaches clusters of 4 general purpose

processors sharing access to data storage that may be used for speech recognition. *Deo*, 9:43-58 (Ex. 1026). U.S. Patent No. 6,615,338 to Tremblay et al. teaches a clustered architecture in a VLIW processor including three DSPs operating as media functional units sharing memory to add, multiply-add, shift, compare, and perform other functions for a general function unit. *Tremblay*, Abstract, 6:51-7:6, Fig. 3 (Ex. 1027). *Tremblay* expressly envisions its clustered architecture being used for speech processing. *Tremblay*, 11:10-21, Fig. 3. U.S. Patent No. 5,922,076 to Garde teaches a digital signal processor designed to optimize performance for speech recognition using a cluster of four, six, or eight processors interconnected by an external bus to memory. *Garde*, 1:10-14, 1:66-2:6, 4:10-21, Fig. 1 (Ex. 1028). In summary, the '319 Patent did not invent the concept of clustered processors, nor did it originally conceive of the concept of employing clustered architectures for speech recognition. A POSITA at the time of the '319 Patent would have been familiar with systems using clustered processors for speech recognition and the advantages gained by utilizing a clustered processor architecture.

VI. OVERVIEW OF THE '319 PATENT

60. I have reviewed the '319 Patent (Ex. 1001). Descriptions of the '319 Patent that may be pertinent to my analysis are provided herein, and my descriptions of the '319 Patent are intended only as a summary of the '319 Patent and/or intended as being pertinent to the Challenged Claims of the '319 Patent. My opinions

provided herein are not intended as a discussion of my entire understanding of the '319 Patent or any other claims in any related patent to the '319 Patent. The '319 Patent describes a speech recognition circuit generally reciting 5 elements: an input buffer, a plurality of lexical memories, a plurality of grouped processors, a control processor, and a results memory. *Thelen* teaches a speech recognition circuit, providing a plurality of recognition models operating in parallel, each targeted to a “specific part of the huge vocabulary” of the system and controlled by a single controller. *Thelen*, 1:53-2:16, 7:30-49. *Thelen* expressly teaches that each of its recognition models may have its “own processing resources,” such as separate processors or processing units in a parallel operating processor. *Thelen*, 2:30-33, 7:17-23. *Thelen* does is not explicit about an input buffer, the physical memories recited, or groups of processors. *Bailey* provides disclosure of lexical memories and results memory, as well as further detail on an input buffer and parallel processors. Thus, of the 5 recited elements, the system of *Thelen*+*Bailey* lacks only the “groups” of processors, taught by the clusters of *Chen*.

VII. OVERVIEW OF THE PRIOR ART

A. Overview of *Thelen*

61. *Thelen* (Ex. 1030) teaches a speech recognition circuit including a plurality of processors controlled by a controller 350, operable to recognize processed incoming speech parameters using a plurality of recognition models,

wherein each of the “recognition models” is targeted to a “specific part of the huge vocabulary” of the system. *Thelen*, 1:53-2:16, 7:30-49. *Thelen*’s speech recognition is similar to the ’319 Patent’s in that it provides a speech recognition circuit under the control of a controller that uses lexical data for speech recognition divided into parts. *Compare Thelen*, 1:53-2:16, 7:30-40, 7:50-58 with *’319 Patent*, 3:39-43, 10:59-11:5. Thus, in my opinion, *Thelen* is in the same field of endeavor of the ’319 Patent and reasonably pertinent to a problem faced by the inventors of the ’319 Patent.

B. Overview of *Bailey*

62. *Bailey* (Ex. 1031) teaches a speech recognition circuit that leverages a plurality of pattern recognition engine processors 525a-c, each operating in parallel using its own private memory, to process incoming speech from an input buffer and store the results in a results memory. *Bailey*, 10:18-11:37, Fig. 3A. *Bailey*’s architecture is similar to the ’319 Patent’s in that it provides multiple processors connected in parallel to an input buffer and controlled by a control processor. *Compare Bailey*, 9:30-39, 10:59-11:37, Fig. 3A with *’319 Patent*, 4:53-5:4, 5:25-31, Fig. 2. Thus, in my opinion, *Bailey* is in the same field of endeavor of the ’319 Patent and reasonably pertinent to a problem faced by the inventors of the ’319 Patent.

C. Overview of *Chen*

63. *Chen* (Ex. 1005) teaches an arrangement of processors into clusters, wherein each of the processors in the cluster has access to a common shared memory. *Chen*, 5:9-17, 9:5-19, Fig. 4. Like the '319 Patent, *Chen*'s parallel processing computing platform utilizes clustered processors to reduce improve computing power and performance. *Compare Chen* (Ex. 1005), 3:17-27, 5:9-17, 9:5-43, Fig. 4 with '319 Patent, 3:13-24, 5:6-41, Fig. 2. Thus, in my opinion, *Chen* is in the same field of endeavor of the '319 Patent and reasonably pertinent to a problem faced by the inventors of the '319 Patent.

D. Overview of *Huang*

64. *Huang* (Ex. 1034) teaches a speech recognition circuit utilizing an HMM acoustic model and lexical trees. *Huang*, 1:5-39, 2:32-52. Specifically, *Huang* describes a process by which each phoneme in the data set of recognizable words (based on incoming training data) is modeled using monophone models which “are used in the recognition scheme to determine a most likely matching phoneme for an input speech utterance[.]”. Later, during speech recognition, context dependent phone models are dynamically-generated, and the monophone models are used to determine target phonemes of spoken words to be searched in the prefix tree (after further steps regarding. *Huang*, 7:45-8:20, 11:64-12:46. Thus, in my opinion, *Huang*

is in the same field of endeavor of the '319 Patent and reasonably pertinent to a problem faced by the inventors of the '319 Patent.

E. Overview of *Jiang*

65. *Jiang* (Ex. 1004) teaches a speech recognition system having a plurality of acoustic models that performs speech processing, acoustic model scoring, and word identification. *Jiang* at Abstract, 4:8-27, 4:51-5:3, 6:46-7:10, 7:30-8:51. *Jiang* teaches a speech recognition circuit utilizing an HMM acoustic model to compute probabilities and generate scores for searching a lexical tree. *Compare Jiang*, 2:1-15, 6:17-28, 7:29-54, 8:16-64 with '319 Patent, 1:30-43, 6:41-48, 8:64-9:29. Thus, in my opinion, *Jiang* is in the same field of endeavor of the '319 Patent and reasonably pertinent to a problem faced by the inventors of the '319 Patent.

F. Overview of *Katz*

66. *Katz* (Ex. 1035) teaches a speech recognition circuit that “estimat[es] the probability that a word from a vocabulary of words is the next word in a string” using m-grams. *Katz*, 1:8-15, 2:35-44, 9:41-44. *Katz*, like the '319 Patent, teaches a speech recognition circuit utilizing a language model processor. *Compare Katz*, 6:33-41, 2:34-44, Figs. 2-3, with '319 Patent, 5:21-46, 7:64-8:9, 8:52-9:6, Fig. 2. Thus, in my opinion, *Katz* is in the same field of endeavor of the '319 Patent and reasonably pertinent to a problem faced by the inventors of the '319 Patent.

VIII. OPINIONS REGARDING GROUND 1

A. Obviousness of Modifying Electronic Hardware and Software to Be Circuitry

67. In my opinion, it would have been obvious to a POSITA to implement the computerized systems and methods taught by *Thelen, Bailey, Chen, Huang, Jiang*, and/or *Katz* specifically using “circuitry,” as recited in the claims. A POSITA would have recognized that computers are electronic devices comprising at least one or more computer processors, memory, a power source, and electrical connections therebetween and operating on electricity, thus teaching or at least rendering obvious “circuitry” performing the computerized steps taught by each of *Thelen, Bailey, Chen, Huang, Jiang*, and *Katz*. To the extent any particular citation from one of these references describes computerized methods stored as software, a POSITA would have found it obvious to physically implement those computerized methods using electronic circuitry such as a processor core and associated memory. Thus, in my opinion, a POSITA would have found it obvious and been motivated to embody the various electronic components disclosed in the references as “circuitry” without undue experimentation.

B. Claims 46, 50, 54, 64 and 67 Are Obvious Over *Thelen* in View of *Bailey* in Further view of *Chen*

1. Claim 46

a) Claim 46(Pre): “A speech recognition circuit comprising”

68. Though *Thelen* does not expressly use the word “circuit,” a POSITA would have recognized that *Thelen* discloses a circuit for performing speech recognition. 1:5-12, 6:46-66, Abstract, Figs. 1, 3. *Thelen* describes in detail a number of computer-implemented structures and hardware components including a microphone, A/D converter, and PCs comprising one or more processors and program modules that would have comprised “circuitry” as understood by a POSITA. *Thelen*, 6:46-66, 7:17-26, 9:33-62, Fig. 5. Further, a POSITA would have found it obvious to implement the structures and computerized methods described by *Thelen* using “circuitry,” especially since *Thelen* expressly teaches its invention may be implemented using “conventional computer technology.” *Thelen*, 9:33-62.

b) Claim 46(a): “an input buffer receiving processed speech parameters;”

69. *Thelen*’s circuit includes a “Spectral Analysis Subsystem 335” in which a speech input signal from an interface 320 is “spectrally and/or temporally analyzed to calculate a representative vector of features.” *Thelen*, 4:29-33. An A/D converter then samples analogue speech to digital samples for further “processing,” blocking the samples into frames to produce a “feature vector” or “observation vector” (OV).

Thelen, 4:34-36, 6:55-66. The processed speech parameters are then provided in parallel to a plurality of speech recognizers (REC 1-3) 331-333.

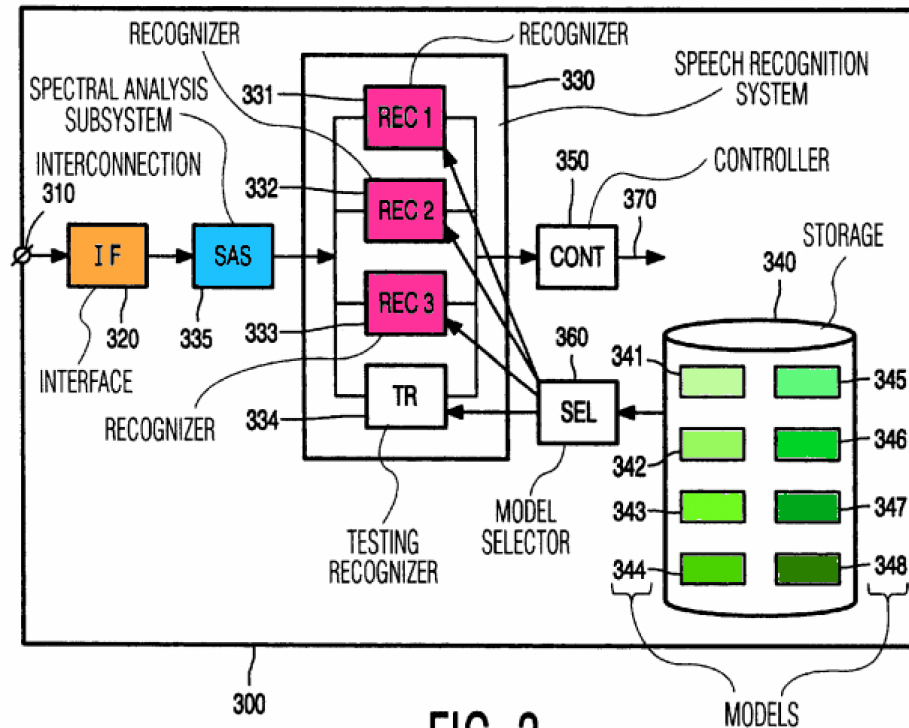


FIG. 3

Thelen, Fig. 3 (annotated), 7:6-7, 7:17-29. The observation vectors OV (mapped as the “processed speech parameters” of claim 46) are transmitted out from the spectral analysis subsystem (335 in FIG. 3 above and 110 in FIG. 1 below):

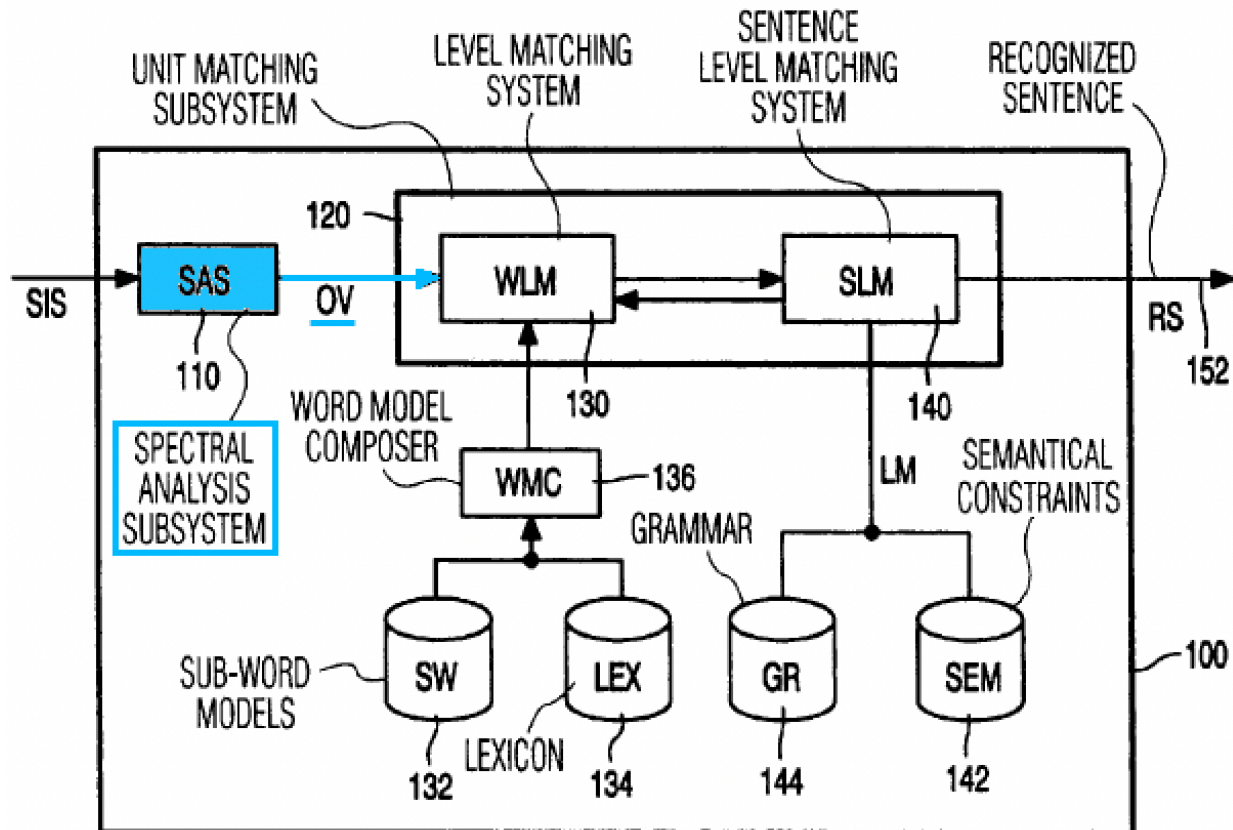


FIG. 1

Thelen, FIG. 1 (annotated).

70. Meanwhile, *Bailey* similarly teaches a speech recognition circuit including signal generator 522 comprising an A/D converter for transforming audio signals captured from a microphone into “digital data which the general purpose computer system may process.”

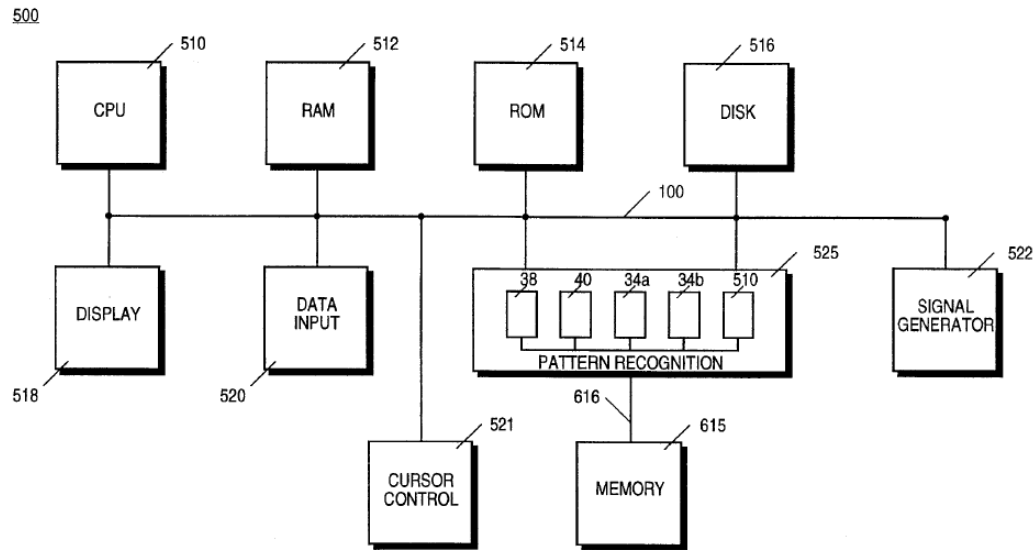


FIG. 2

Bailey, 9:31-39, 9:1-30, Fig. 2. *Bailey* expressly describes this process “isolat[es] unknown sound segments from the sampled digital data[,]” subsequently storing them in RAM 512. *Bailey*, 9:36-41, 11:9-29, Figs. 2, 3A. Thereafter, the processed speech parameters are provided from RAM for recognition by a plurality of pattern recognition (PR) engines 525a-c via comparison to a plurality of known prototype patterns. *Id.* Much like *Thelen*, *Bailey* describes this first step of speech recognition as “a preprocessing or feature extraction and normalization step,” performed at CPU 510. *Bailey*, 12:12-15; *Thelen*, 4:20-33, 5:2-53, 7:50-58. Next, *Bailey* describes that a highly “processing intensive” “search and match step” of is dedicated to a Processing Engine (PRE), which delegates this heavy processing away from the CPU 510. *Bailey*, 12:12-15. Thus, speech is preprocessed by CPU, and the preprocessed speech, which (per *Thelen*) would have included observation vectors

(or “feature vectors”) is sent to the PRE 525 for pattern matching. *Bailey*, 11:15-37; *Thelen*, 6:53-66, FIG. 1. *Bailey* then teaches that pattern recognition is performed on unknown sound patterns stored in RAM 512:

The signal generator 522 also includes an [sic] mechanism for **isolating unknown sound segments from the sampled digital data and storing those unknown sound segments into a memory unit such as RAM 512**. These sound segments are one type of unknown pattern that the present invention will process in order to recognize.

Bailey, 9:35-41, 11:16-17 (“unknown input **pattern** may be **temporarily stored** in memory array 512.”). I note that *Bailey* includes exemplary code with a comment stating that vectors are loaded from external memory 615. *Bailey*, 50:30. A POSITA would have recognized that this is referring to loading vectors for reference patterns (not input patterns) because memory 615 stores reference patterns. *Bailey*, 14:30-33, 13:11-12. Further, a POSITA would have known that both that code and its included commentary are merely exemplary and not limiting to the functions taught by *Bailey*.

71. A POSITA would have understood, based on these teachings that *Bailey* teaches the claimed “input buffer receiving processed speech parameters.” *Bailey*, 3:60-4:1, 42:54-57, 43:7-48. Input buffers were well-known hardware-based data structures prior to the ’319 Patent, providing temporary storage in memory for passing data between computing components. This understanding of an input buffer by a POSITA mirrors the description of *Bailey* of passing data corresponding to

unknown input patterns of sound from its A/D converter to the plurality of pattern recognition (PR) engines 525a-c (for the “search and match” step) via temporary storage in RAM 512. *Bailey*, 9:35-41, 11:9-37, 12:12-15. Specifically, a POSITA would have recognized *Bailey’s* description of an “unknown input pattern” (11:16-17) being stored in RAM 512 to be describing storing preprocessed (“feature-extracted”) speech, including, at the least, *Bailey’s* preprocessed speech parameters in RAM.

72. In an example, *Bailey* explains that pattern recognition of handwriting is performed via CPU 510 preprocessing incoming coordinates and sending a normalized set of features to the PR engine 525, though notably *Bailey* is clear that its description is intended to apply equivalently to speech recognition as well (i.e. preprocessing an unknown *speech* signal to send a normalized set of features to the PR engine 525). *Bailey*, 18:44-48. *Bailey* explains that the “CPU may input up to n separate unknown patterns from block 522 during a single recognition processing cycle, [where] a separate unknown would be sent to each PR engine over bus 100,” by temporarily storing the unknown input pattern in RAM 512. *Bailey*, 11:16-17, 11:23-26. According to *Bailey*, such an unknown input pattern may be “composed of a multitude of points or states” such as, specifically, HMM states determined via feature extraction. *Bailey*, 1:45-51, 4:35-39, 49:51-66 (“...the set of distribution probabilities, for all features, in all states, and for all possible features

values....temporary results...are also stored in data memory 32”), FIG. 1B. Thus, a POSITA would have recognized that *Bailey* teaches extracting feature vectors used for HMM pattern matching from incoming speech, and the unknown input patterns are temporarily stored in RAM 512.

73. RAM was a well-known hardware component for providing such a “buffer” as described by *Bailey* prior to the ’319 Patent, as can be seen in prior art references such as *Pawate*. *Pawate* describes a speech recognition circuit in which frames of speech are transmitted from a Feature Extractor to a “RAM-based” Generalized Set Feature Buffer (GSFBUF). *Pawate*, 11-12, 15. Thereafter, the extracted features are transmitted to two processors that perform, in parallel, the “computationally intensive task” of word hypothesizing (i.e. *Bailey*’s “search and match” step). Here, *Pawate* is employing an input buffer to temporarily store processed speech parameters (feature vectors) prior to pattern matching. Other prior art patents, such as *Yasuda*, *Ng*, *Emry*, and *Giltner* teach input buffers in RAM. *Yasuda* (Ex. 1042), 10:67-11:2, 13:57-58 (“State 204 serves to unload RAM input buffers and transfer them to disk when full.”), 14:4-7, 14:19-20, 14:35-47, 14:60-63 (“State 252 loads the caller’s message in RAM input buffers to disk memory.”); *Ng* (Ex. 1041), 5:42-46 (“The MPU 110 programs a DMA transfer to the processor RAM input buffer[.]”); *Emry* (Ex. 1040), 3:40-42 (“RAM input buffer 29[.]”), 3:60-66, 4:6-11, Fig. 2; *Giltner* (Ex. 1039), 6:46-58 (“RAM input buffer 42.”), 9:40-45

(“The message data stored in the transmit mode RAM input buffer 42 is called up, one character at a time, to a word detector and buffer[.]”), 7:29-33 (“random access memory input buffer 44.”), Fig. 1 (depicting RAM input buffer 42 and RAM input buffer 44). As can be seen, input buffers implemented in the RAM of computing circuits was very well-known and widely-used prior to the ’319 Patent.

74. A POSITA would have had a reasonable expectation of success for combining the input buffer taught by *Bailey* into *Thelen* for receiving the processed speech parameters (already taught by *Thelen*), requiring only RAM (a form of digital “storage,” already described by *Thelen* and ubiquitous prior to the ’319 Patent) and software changes well within the skill of a POSITA at the time. A POSITA would have appreciated that speech recognition circuits would have benefitted from the decoupling of the rate of the “search and match” step from the rate of the preprocessing step, eliminating the bottleneck that can be caused and allowing the system greater speed and/or efficiency, motivating the combination. A POSITA would have expected each of the speech recognition circuits of *Thelen* and *Bailey* to include such RAM-based input buffers, as omitting the buffer would have resulted in systems that were both inferior and harder to make operational. For example, *Thelen* teaches consecutive samples of digitized speech input are grouped into overlapping frames. *Thelen*, 4:30-43, FIGs. 1, 3. A feature vector is then determined for each frame of speech, and the vectors are transmitted to *Thelen*’s plurality of

recognizers, which receive the processed speech parameters. *Id.* A POSITA would have recognized that by incorporating *Bailey's* input buffer, such as that disclosed by, would have enabled the SAS 110,335 of the modified *Thelen* circuit to digitize, sample, and block/group the frames at its maximum operational rate, loading them into the input buffer for the pattern recognition (PR) engines 525a-c to utilize at *their* maximum operational rate, potentially speeding both steps.

- c) **Claim 46(b): “a plurality of lexical memories containing in combination complete lexical data for word recognition, each lexical memory containing part of said complete lexical data;”**

75. Words recognizable by *Thelen's* large vocabulary system are modeled as sequences of sub-word units (such as phones, diphones, or syllables) and stored in lexicon 134, as seen below in Fig. 1.

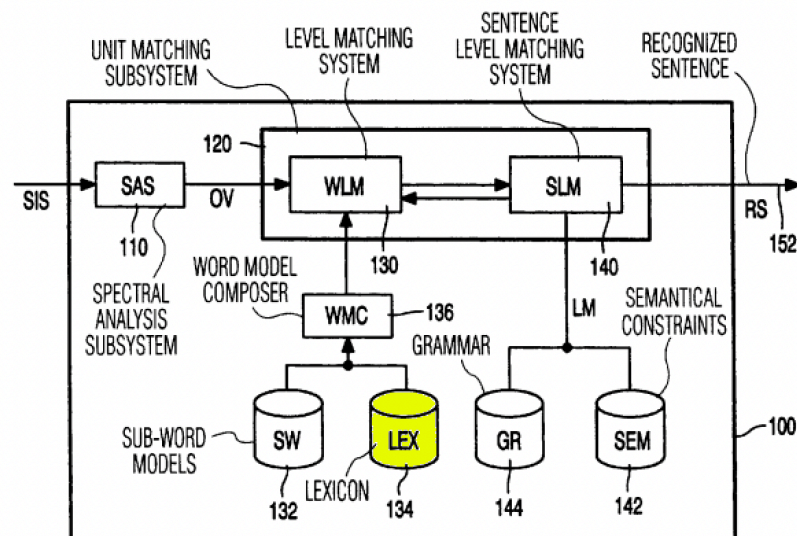


FIG. 1

Thelen, 5:14-22, Fig. 1 (annotated). *Thelen* provides a plurality of “speech recognizers,” each associated with a particular “recognition model” stored in memory and targeted to a “specific part of the huge vocabulary” of the system. *Thelen*, 1:53-65. *Thelen* characterizes this division of its “complete” data for word recognition as being an “improvement which is better capable of dealing with huge vocabularies,” because it limits the possible sequences of sub-word units that may be matched using a Hidden Markov Model (HMM) to only sequences stored in lexicon 134, “reduc[ing] the outcomes” of allowable recognized sequences of sub-word units to only those present within lexicon 134. *Thelen*, 1:53-65, 5:24-53. Here, a POSITA would have recognized that the sum combination of the data stored in all the recognition models, which are all stored in lexicon 134, constitute “complete lexical data” because that sum combination of data contains all of the sequences of phones that represent words recognizable to the system. A POSITA would have further recognized that the data stored within *each* of the recognition models would then represent “part of said complete lexical data.” Yet further, a POSITA would have understood each lexicon 134 of each recognition model comprises “lexical data,” as claimed, because the speech recognition circuit is **constrained** to only identify words stored in the lexicon for the particular recognition model pertaining to its particular subset of the “huge vocabulary” of the overall *Thelen* system.

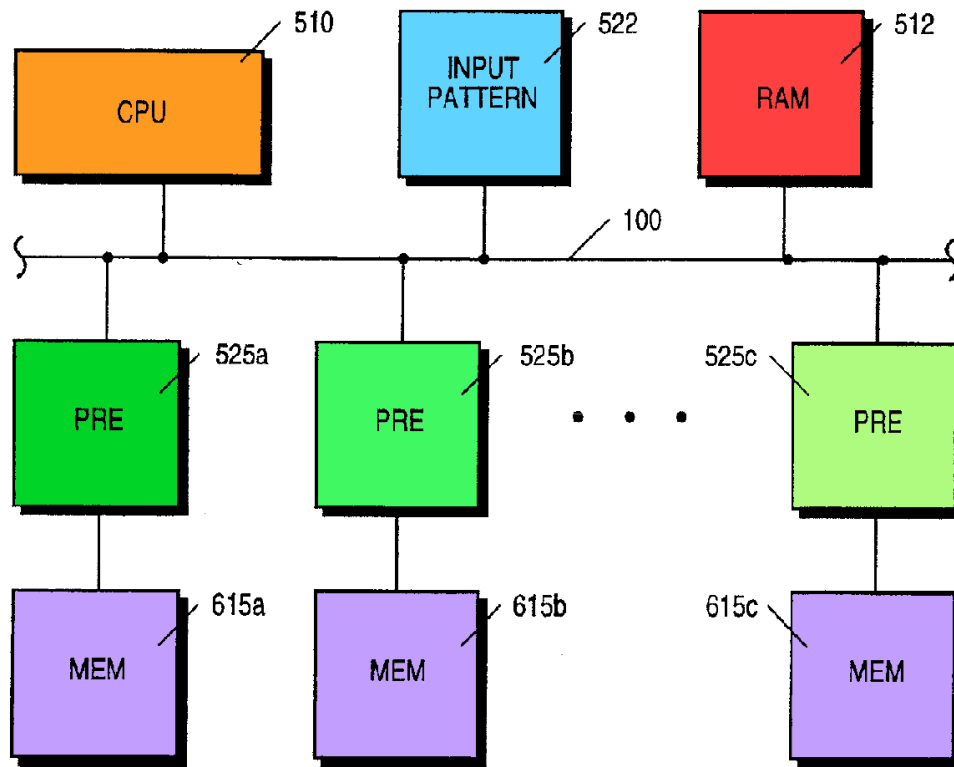
76. Notably, *Thelen* teaches that the recognition models will overlap in *some* of the data contained within each model but will also have portions distinct to the context of each particular model representing jargon specific to that context. *Thelen*, 2:5-16. Example of “contexts” that may be recognized by a particular recognition model include health, entertainment, computers, arts, business, education, government, science, news, and/or travel. *Thelen*, 2:3-5. Specifically, *Thelen* explains “each of those contexts will normally overlap in vocabulary, for instance in the general words of the language. **The contexts will differ in statistics of those common words as well in the jargon specific for those contexts.** By using several of those models to recognize the input, a wider range of utterances can be recognized using properly trained models. A further advantage of using several models is that this allows a better discrimination during the recognition. If one huge vocabulary was used, certain utterances would only be recognized in one specific meaning.” *Thelen*, 2:5-16. This understanding, that each of the recognition models will have at least some words not present in one or more of the other recognition models, is reiterated, by *Thelen*, teaching that overlap between recognition models will usually occur in **part** of the vocabulary. *Thelen*, 7:30-40. This arrangement, wherein the “parts of said complete lexical data” have some overlap as well as potentially some differences, is consistent with embodiments supported by the ’319 Patent that “increase[] flexibility” of the system by storing “more than one copy of

the lexical data” in different lexical memories, such that “more than one lexical tree processor cluster 22 **can have access to the same lexical data.**” ’319 Patent, 10:59-11:5. The ’319 explains that “[t]his increases the flexibility and allows the controller to avoid any bottlenecks in processing which, for example, can occur if the highest pass scores all require processing by one or a number of lexical tree data structures residing in one of the acoustic model memories.” A POSITA would have recognized the ’319 Patent supports and claims, via claim 46(b), speech recognition circuits in which the “parts of said complete lexical data” stored in each lexical memory at least somewhat overlap – that is, there may be at least some (if not complete) overlap between the “parts” of the lexical data, so long as the “parts” of the lexical data provide, in combination, the **complete** lexical data. ’319 Patent, 10:59-11:5, 15:55-16:2 (Claim 46). Thus, *Thelen*’s recognition models teach the claimed “parts of said complete lexical data.” Notably, the claimed “lexical data” of claim 46 is not required to be in any particular format until claim 47, which specifies that the lexical data is formatted as a lexical tree.

77. *Thelen* teaches memory storing the complete lexical data, stating that its recognition models, which are formed of lexicon 134 and sub-word models 132, are “loaded from a storage 340.” *Thelen*, 7:30-34, 5:22-24, FIG. 3. Here, a POSITA would have recognized that since they are *loaded* from storage 340, the models were *stored* in storage memory 340. *Id.* *Thelen* does not expressly discuss a “plurality of

lexical memories” containing the recognition models. *Thelen* does suggest, however, that the separate recognition engines may “**each hav[e its] own processing resources**” so that “the user does not experience a significant delay in the recognition.” *Thelen*, 2:30-33. A POSITA would have recognized that these processing resources may have comprised processors and memory. In a specific embodiment, *Thelen* envisions that each recognizer (including its processing resources) may be associated with an exclusive recognition model in a “fixed one-to-one relationship.” *Thelen*, 7:42-49.

78. *Bailey’s* system is a “high speed speech recognition” circuit “having the processing power and efficiency to realize real-time accurate pattern recognition” using a plurality of pattern recognition engines 525 in parallel, “splitting the processing work.” *Bailey*, Abstract, 1:16-20, 2:62-66, 10:18-44, 10:59-11:8, Fig. 3A. As shown below in Figure 3A, each of *Bailey’s* **pattern recognizers 525a-c** is associated with a respective **private memory 615a-c**:



Bailey, 10:18-11:37, 13:25-51, Fig. 3A (annotated). Here, each PRE 525a-c represents a “Pattern Recognition Engine,” and should not be confused as being “preprocessors.” *Id.* Each PR engine 525a-525c interfaced with its own private memory 615a-615c containing “its own library of prototype patterns” for comparison, in parallel, with frames of unknown speech data. *Bailey*, 10:18-11:44, 11:9-29, 13:25-51, Fig. 3A.

79. In my opinion, a POSITA would have been motivated and found it obvious to combine the plurality of lexical memories taught by *Bailey* into the speech recognition system taught by *Thelen* to store the parts of the lexical data constituting each of the recognition models conveniently and efficiently in a 1-to-1

relationship with the “processing resources” utilizing that data, i.e. in each memory 615a-c corresponding to each PRE 525a-c. A POSITA would have recognized that the modifications necessary would have required only rearrangement of well-known memory components and relocation of lexical data already present in *Thelen*, leading a POSITA to a reasonable expectation of success for the combination. *Bailey* expressly motivates this combination, stating that “the PR engine 525 of the present invention may be coupled with any general purpose computer system to increase that computer system’s accuracy and performance regarding pattern recognition applications. It is appreciated that the present invention is able to provide the above functionality in a real-time response of at most 0.1 seconds between unknown pattern input and pattern identification, thus providing a system that is fully user interactive[,]” suggesting to a POSITA that *Bailey*’s structures and concepts would appropriately and advantageously be applied to the similar speech recognition system of *Thelen*. *Bailey*, 12:3-11. The combination would have furthered *Thelen*’s expressly-stated goal of operating in “real-time.” *Thelen*, 2:30-37, 7:26-29. Modifying *Thelen* via *Bailey* would have required combining prior art elements according to known methods to yield the predictable result of a plurality of speech recognizers operating in parallel using lexical data stored in a plurality of lexical memories, within a POSITA’s skill without undue experimentation.

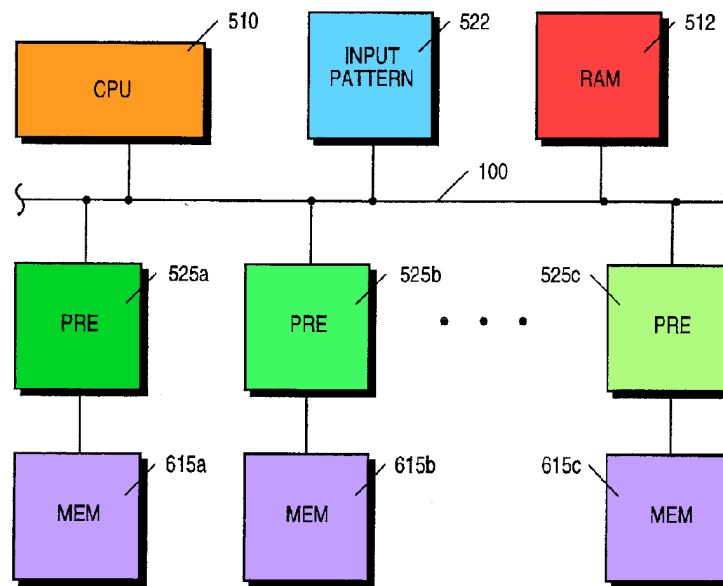
d) Claim 46(c): “a plurality of processors connected in parallel to said input buffer for processing the speech parameters in parallel,”

80. Both *Thelen* and *Bailey* teach a plurality of processors connected in parallel, and *Thelen* modified by *Bailey* teaches a speech recognition circuit comprising a plurality of processors, connected in parallel to an input buffer. *Thelen* alone teaches a plurality of recognizers 331-333 (illustrated in Fig. 3, also referred to as “recognition engines”) operating in parallel includes a separate processor:

The recognizers operate in ‘parallel’, in the sense that they independently recognize the same speech input at almost the same moment. This can be realized by using separate resources for each of the recognizers, such as **a separate processor** or processing unit in a ‘parallel’ operating processor, such as a **VLIW processor**.

Thelen, 7:17-23. Here, a POSITA would have recognized *Thelen* teaches “a plurality of processors connected in parallel,” as claimed, because each of the recognizers is implemented via separate processing resources including separate VLIW (very long instruction word) processors. Since, as discussed above, *Thelen* teaches inputting observation vectors (speech parameters) to the recognizers, which operate on the input speech “in parallel,” independently recognizing speech, a POSITA would have understood or at least found obvious that *Thelen* teaches a plurality of processors “processing the speech parameters in parallel.” *Thelen*, 7:2-7, Figs. 2-3. Similarly, as illustrated below in Fig. 3A, *Bailey* teaches a plurality of **PR engines 525a-c** is

connected in parallel to **RAM 512** (comprising the input buffer, as mapped above for Claim 46(a)) via communication bus 100:



Bailey, 10:18-11:37, 13:25-51, Fig. 3A (annotated). *Bailey* expressly teaches that its plurality of pattern recognition engines 525 are “in parallel,” and a POSITA would have recognized the arrangement of recognition engines 525 illustrated in Fig. 3A as being connected “in parallel” to the input buffer, mapped above as being located within RAM 512. *Bailey*, Abstract, 9:5-39, 11:23-34, Fig. 3A. A POSITA would have appreciated that each of the PREs comprises a processor because it performs processing via its execution block 417. *Bailey*, 13:25-51, 14:15-29. Thus, in the modified circuit of *Thelen+Bailey*, *Thelen*’s plurality of recognizers (with each recognizer including its own processor) is connected in parallel to *Bailey*’s RAM 512 (which includes the claimed input buffer), and where the plurality of *Thelen*

recognizers process the speech parameters (OVs) in parallel. However, neither *Thelen* nor *Bailey* expressly teaches “processors being arranged in groups of processors,” as claimed. Additionally or alternatively, *Bailey*’s plurality of PREs (each of which comprises a programmable multiprocessor) are arranged in parallel and connected to *Bailey*’s RAM 512.

81. As above for limitation 46(b), in my opinion, a POSITA would have combined *Bailey*’s memory architecture with the plurality of parallel “recognition engines” taught by *Thelen* to further *Thelen*’s expressly-stated goal of operating in “real-time,” using well-known memory components and relocation of lexical data already present in *Thelen*, leading a POSITA to a reasonable expectation of success for the combination. *Thelen*, 2:30-37, 7:26-29. The combination would have had a reasonable expectation of success to a POSITA because it constituted combining prior art elements according to known methods without undue experimentation to yield the predictable result of a plurality of speech recognizers operating in parallel using lexical data stored in a plurality of lexical memories for the benefit of increased speed and/or efficiency of the system.

e) Claim 46(d): “said processors being arranged in groups of processors, each group of processors being connected to a lexical memory;”

82. As illustrated below in Fig. 4, *Chen* teaches clusters 100a-d comprising groups of processors 102a-d that are symmetrically connected to respective cluster

shared memories 104a-d via connection nodes 106a-d (labeled “switching nodes” in

Fig. 4):

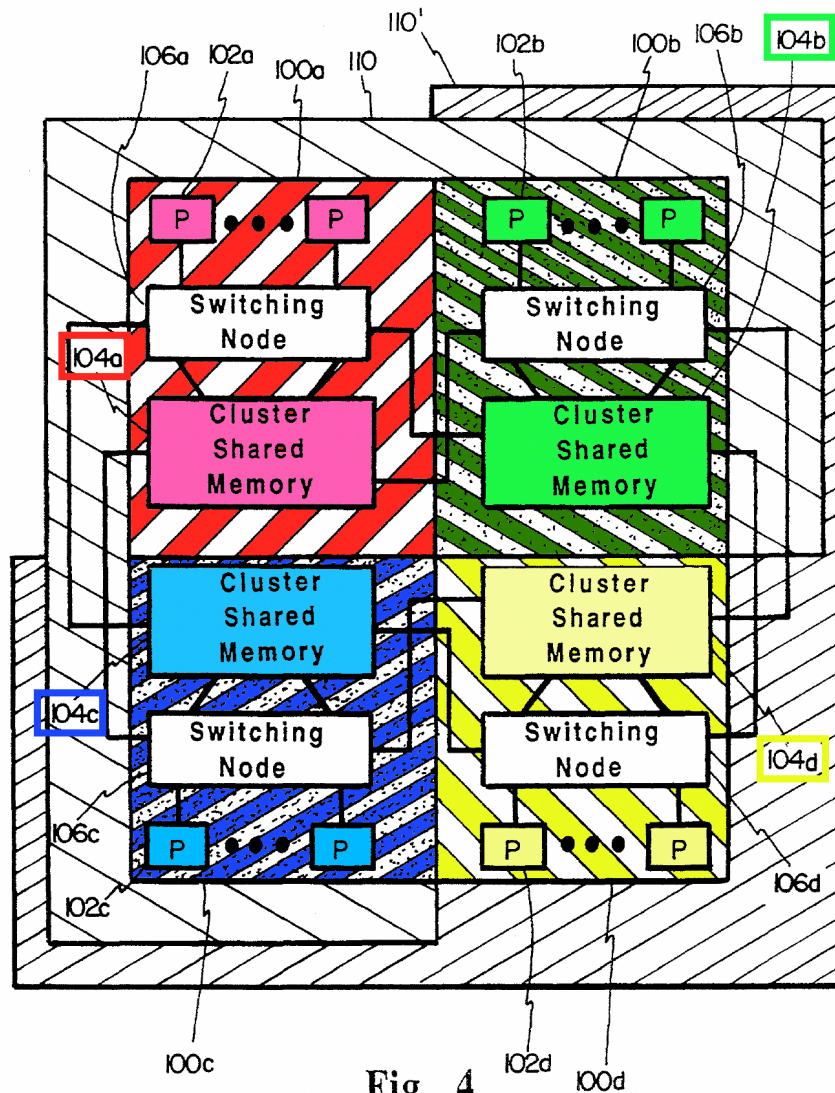


Fig. 4

Chen, Fig. 4 (annotated), 9:5-19. Specifically, *Chen* teaches that its configuration of processors arranged in clusters teach “groups” of processors, where each of the processors attached to a given cluster shared memory (which may be any formed using commercially available memory) constitute a “group” of processors. A

POSITA would have stored at least a portion of the lexical data (the context-specific recognition models of *Thelen*, as discussed regarding limitation 46(a)) in the shared cluster memories 104a-d of each of the clusters 100a-d taught by *Chen* for access by processors 102a-d during speech recognition such that *Chen's* shared cluster memories 104a-d perform the same conceptual purpose as memories 615a-c, storing the library reference pattern for each of the pattern recognition engines 525a-c. Thus, since each of these modified memories would have stored "lexical data," each would have been recognized by a POSITA to teach a "lexical memory," as claimed. Thus, this configuration of the system of *Thelen+Bailey+Chen* teaches a "plurality of processors connected in parallel to said input buffer for processing the speech parameters in parallel, said processors being arranged in groups of processors, each group of processors being connected to a lexical memory[,]" as claimed.

83. The intended mapping of the Petition is grouping *Thelen's* recognizers 331-333 in clusters of recognizers (where each recognizer already has a processor, per *Thelen*) and connecting each group of recognizers (and associated processor) to a memory, as taught by *Chen* and mapped for Claim 46(d). *Bailey* teaches a plurality of lexical memories 615a-c (see *Bailey*, FIG. 3A) and connecting a group of recognition engines with a lexical memory (FIG. 3B). Thus, in the modified *Thelen*, each group of recognizers is connected to a lexical memory, such that the plurality of groups of recognizers is respectively connected to the plurality of lexical

memories comprising, in combination, complete lexical data. A POSITA, in my opinion, would then have found it obvious and been motivated to group the plurality of processors in *Thelen* such that particular contexts (e.g., all sports contexts) are clustered together in a particular group for efficient word recognition and storage of lexical data using less memory. *Thelen*, 8:6-8, 8:54-56 (discussing “recognition models are hierarchically arranged from models with a more generic context to models with a more specific context”), 8:54–9:32 (discussing specific sports models such as football or baseball). As seen below in annotated Fig. 4, *Thelen* envisions a hierarchical grouping of contexts such that similar contexts are associated under similar higher-level contexts:

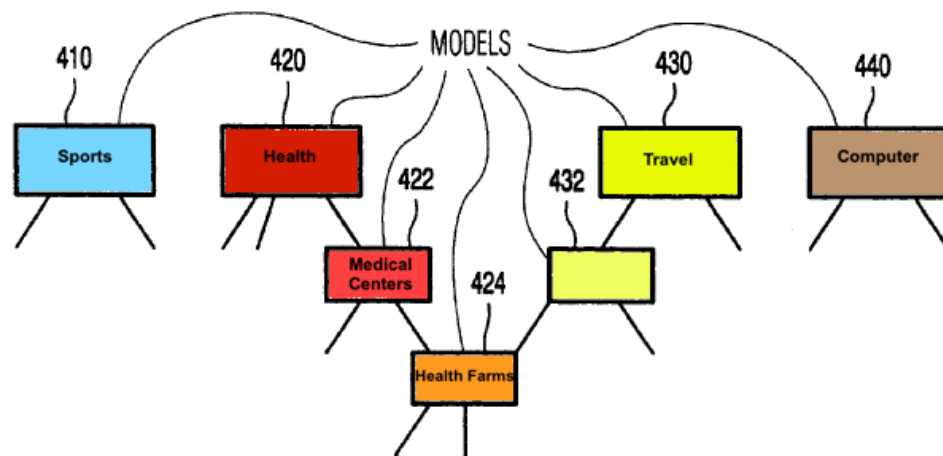


FIG. 4

Thelen, FIG. 4 (annotated). In the modified *Thelen-Bailey* circuit and as an example only, a first group of processors may be associated with the “sports” context 410, such that its lexical memory (e.g., one of the memories 615 taught in *Bailey*) stores

the recognition models for the sports contexts. A second group of processors relates to the “health” context, where, as taught by *Thelen*, one of the subdivisions (model 422) is “hospitals/medical centers.” *Thelen*, 8:62-66. An even further subdivision (model 424) may be “health farms.” *Thelen*, 8:65–9:1. *Thelen* notes how the “health farms” context may also be related to travel below model 432, stemming from a “travel” context. *Thelen*, 9:1-11. Each of these overarching contexts would be associated with a particular group of processors. A POSITA, then, would have found it natural to associate each of *Thelen*’s contexts with the respective memory of distinct clusters of processors of *Chen* being utilized as the PREs of *Bailey* yield the predictable result of consolidating speech recognition for a particular context within a particular group of recognition processors. Since context-related recognition models would then be stored in the same lexical memory, the modified system would have been configured to store the overlapping lexical data in fewer unique locations and transfer recognition models to/from the lexical memory for processing more efficiently. A POSITA would have recognized that associated contexts, such as the sports contexts (baseball, football, etc.) would have shared a larger amount of lexical data than contexts which are not associated. The overlapping data within a particular context would then be reproduced fewer times because the single copy of the data would be shared amongst the processors of that group.

84. A POSITA would have been motivated and found it obvious to modify the system of *Thelen+Bailey* to utilize groups of processors to reduce the duplication of lexical data as well as divide the processing load amongst many simple, commercially available processors to increase the processing power of the speech recognition circuit while speeding the responsiveness of the system and maintaining the keeping the financial cost of the overall system low. *Thelen's* stated goal of recognizing speech in "real time" in the sense that no significant delay occurs" between a word being received in the system and a recognizing the word would suggest to a POSITA that *Chen's* processor arrangement would have been a desirable improvement to increase processing resources and thus, recognition speed. Specifically, a POSITA would have been motivated to modify or replace the individual processors used for pattern recognition engines 525a-c taught by *Bailey* with the clustered processors taught by *Chen* to expedite the search process by leveraging multiple processors as the "processing resources," speeding *Thelen's* speech recognition. A POSITA would have had a reasonable expectation of success for the combination because *Chen* and *Thelen+Bailey* contemplates a hardware architecture using multiple, parallel processors, meaning using *Chen's* processor architecture in the combined system constituted a simple substitution of *Bailey's* single processors providing each pattern recognition engine for *Chen's* groups (clusters) of processors providing the same functionality. A POSITA would have

recognized that the data compression enabled by associating *Thelen's* high-level contexts with *Chen's* groups of processors (reducing the duplication of lexical data, as discussed) would have necessitated less memory to operate, further reducing the overall financial cost of the system. Other motivation for the modification would have come from well-known motivations documented in the prior art, such as *Hon*, praising parallel processing configurations for their flexibility and scalability in speech recognition systems. *Hon*, 19-20.

f) Claim 46(e): “a control processor controlling each processor to process said speech parameters using partial lexical data read from a respective said lexical memory; and”

85. Control processors administrating the processing of a plurality of other processors was well-known and used across many fields prior to the '319 Patent, including specifically in speech recognition. For example, each of *Wrench*, *Vensko*, and *Cecinati* teaches a master processor controlling a plurality of template processors to recognize speech in parallel. *Wrench* (Ex. 1010), 4:40-58, 6:40-7:7, Fig. 2; *Vensko* (Ex. 1036), 3:29-64, 6:14-41, Fig. 1; *Cecinati* (Ex. 1037), 1:16-20, 4:11-26, 5:1-23. *Thelen* itself teaches a model selector 360 that “is used to select for at least one of the speech recognizers the associated recognition model,” as well as a controller 350 that directs speech input to the recognizers and selects the final recognized word based on the outcomes of the recognizers' comparisons. *Thelen*, 7:50-8:5. *Thelen's* model selector 360 is expressly described as controlling at least

one of the test recognizers for which it has selected a recognition model, leading a POSITA to the conclusion that the model selector 360 in combination with controller 350 comprises a “processor.” *Thelen*, 8:36-41, 9:8-10. Thus, because *Thelen*’s teaches a model selector 360 that controls a recognizer comprising a processor, selecting, via controller 350, an associated recognition model from amongst a plurality of models, *Thelen* teaches a control processor as claimed (model selector 360 in combination with controller 350). Further, *Thelen*’s control processor directs speech parameters to recognizers via controller 350, which use partial lexical data (their respective associated recognition model) read from a each’s lexical memory (lexical memory 615 for the group of processors in the modified *Thelen+Bailey+Chen* system), further indicating to a POSITA that the control processor would comprise a processor. Thus, a POSITA would have recognized that controller 350 and model selector 360 comprise at least one processor.

86. *Bailey* adds more detail to the description of the control processor of the combined *Thelen+Bailey* system, explaining that CPU 510 “is responsible for splitting the processing work from among the separate PR engines[,]” by sending, during each recognition cycle, unknown patterns of speech to each PR engine 525 for comparison against the library of patterns stored in their lexical memories 615a-c. As described above, in the combined *Thelen+Bailey+Chen* circuit, *Chen*’s groups of processors are used to improve the PR engines 525 by leveraging powerful

clusters of processors for each engine 525, with the shared cluster memories 104a-d of the groups of clustered processors being used as the lexical memories 615a-c of *Bailey*. Further, in the combined *Thelen*+*Bailey*+*Chen* circuit, model selector 360 of *Thelen*, in combination with controller 350 of *Thelen*, is embodied as described by *Bailey* as CPU 510, controlling each of the processors in each of the clusters to process speech parameters from the input buffer using partial lexical data from each of these lexical memories. Though the embodiments taught by each of *Thelen*, *Bailey*, and *Chen* teach particular numbers of the relevant elements (e.g. *Thelen* illustrates 8 recognition models, *Bailey* illustrates 3 pattern recognizers, and *Chen* illustrates 4 clusters), a POSITA would have found it obvious and been capable of configuring the combined system to use any appropriate number of each of these elements to result in an operable speech recognition system. Thus, since *Bailey*'s CPU 510 splits the processing work among the PREs and for performing scheduling, synchronization, and result determination steps in the combined circuit, a POSITA would have understood or found obvious the *Bailey* CPU 510 controls each PRE (including processing resources associated with each PRE, including specifically processors) to process the received speech parameters.

87. A POSITA would have been motivated to realize the control processor of *Thelen* (controller 350 + model selector 360) using *Bailey*'s CPU 510 as an operable, efficient method of controlling the speech recognition process and

selecting a final, “winning” recognized word. Doing so would have apportioned these responsibilities to a single sophisticated processor, allowing each of the plurality of clustered processors to be simpler and therefore less expensive. A POSITA would have been motivated by *Bailey’s* express teaching that its system, may be used to increase the accuracy and performance regarding pattern recognition applications, providing “real-time responses of at most 0.1 seconds between unknown pattern input and pattern identification, thus providing a system that is fully user interactive[,]” further highlighting the desirability for speed and efficiency in the combination, a desire furthered by use of a control processor. *Bailey*, 12:2-11. A POSITA would have recognized this combination as combining prior art elements according to known methods without undue experimentation to yield a predictable result of a control processor (*Thelen’s* controller 350, embodied as *Bailey’s* CPU 510) controlling each processor (each processor 102a-d in each of *Chen’s* groups of clustered processors 100a-d, embodying *Bailey’s* PRs 525a-c) to process speech parameters using partial lexical data (*Thelen’s* recognition models 341-348) read from a respective said lexical memory (*Bailey’s* memories 615a-c modified pursuant to *Chen* to each be shared cluster memories 104a-d of the groups of clustered processors).

g) Claim 46(f): “a results memory storing the results of the processing of the speech parameters from said processors.”

88. After each of *Thelen*'s recognition models have been applied to the incoming unknown speech, *Thelen* teaches that “the output of the recognizers is directed to a controller 350 to make the final selection of a recognized word sequence.” *Thelen*, 7:50-58. Thus, *Thelen* teaches results of processing from said processors, which a POSITA would have taken as a strong indication that *Thelen* itself would have included a results memory, at least because the controller 350 would have used memory to hold the results for comparison amongst the candidates of recognized words. Processors need memory for performing such comparisons, and a POSITA would have recognized that the memory holding the results for comparison would have taught the claimed “results memory.” Further, *Bailey* directly teaches that the control processor (taught by *Thelen* and implemented using *Bailey*'s CPU 510, in the combined system as discussed above) may receive results of the comparisons via one of two schemes: (1) via a report generated at the end of each prototype reference comparison, or (2) at the end of comparisons to the total library. *Bailey*, 11:42-59, 12:41-61, 11:23-34, Fig. 4. Though portions of *Bailey*'s description regarding Figure 4 are discussing identifying unknown input from a stylus, a POSITA would have recognized that *Bailey* intends for its description to be equivalently applicable to unknown input from a stylus or a voice and would have

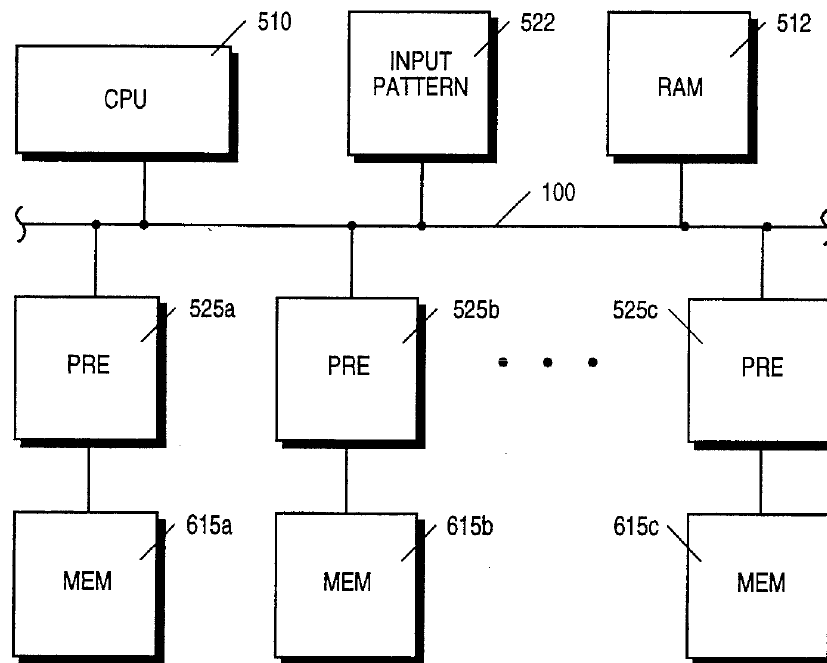
understood the applicability of the description to specifically determining unknown speech inputs. Returning to *Bailey*, regardless of which of the disclosed schemes is utilized, results are returned to CPU 510 via “an interrupt indicating a match complete” which “**generates a transfer of the results from the PR engine to the system memory 512**” so that the CPU 510 can post-process all of the results generated by the processors of the PR engine to formulate a final matched prototype word pattern based on candidates selected by the PR engine. *Bailey*, 11:42-59, 12:41-61. In this way, *Bailey* teaches a results memory (in RAM 512) storing the results of the processing of the speech parameters from each of the processors, as claimed.

89. A POSITA would have been motivated to transmit results via a results memory as taught by *Bailey* in the speech recognition circuit of *Thelen+Bailey+Chen* as a method for passing results to a master control processor well-known to be operable and effective prior to the '319 Patent, leading to a reasonable expectation of success for the combination without undue experimentation. For instance, U.S. Patent No. 4,907,278 to *Cecinati*, filed in 1988, teaches a speech recognition system including a plurality of slave processors under the control of a single master controller that transmits results from the slaves to the master via a memory block (MEME) in which results are stored. *Cecinati*, 1:16-20, 5:1-23, 11:63-68. Similarly, *Robinson* teaches a speech recognition circuit that

searches a lexical tree for an uttered word, storing results in a table in memory. *Robinson* (Ex. 1009), 1:5-20, 6:28-43, 6:63-7:2, 11:30-34. A POSITA would have recognized that storing the results in a memory would have enabled the plurality of pattern recognizers to complete their comparison computations at different times, allowing the system more flexibility in operation. Specifically, a POSITA would have realized that comparisons output by the plurality of processors in groups would naturally have to be reported to the control processor in order for the control processor to determine a final recognized word, and that storing the results in memory would have been the most expected, straightforward manner of configuring the speech recognition circuit. Further, as previously discussed, a POSITA would have been further motivated by *Bailey's* express statement that "the PR engine 525 of the present invention may be coupled with any general purpose computer system to increase that computer system's accuracy and performance regarding pattern recognition applications...to provide the above functionality in a real-time response of at most 0.1 seconds between unknown pattern input and pattern identification, thus providing a system that is fully user interactive." *Bailey*, 12:2-11.

90. I note that *Bailey's* results memory is physically located in RAM 512, the same-numbered memory location of *Bailey's* "input buffer" mapped in limitation 46(a). A POSITA would have understood that the input buffer would have been a separate logical portion of RAM 512 from the results memory portion in order to

prevent mingling of the data, potentially leading to an inoperable system. Further, a POSITA would have recognized that RAM 512 as described and illustrated in Fig. 3A below by *Bailey* is intended to *conceptually* describe the RAM available to the speech recognition circuit, which may physically be provided in multiple logical and/or distinct locations and portions of *Bailey's* circuit.



To the extent that the input buffer and results memory must be embodied as physically distinct memories, a POSITA would have also found it obvious to modify the circuit to split the two, providing separate RAM to cleanly distinguish between the incoming speech parameters and the results of the comparisons of the plurality of processors to the recognition models. RAM was extremely well-known prior to the '319 Patent and well within a POSITA's skill to add, configure, and modify for any desirable purpose, requiring only adding another piece of RAM (an element

already present in *Bailey*) leading to a reasonable expectation of success for the modification without undue experimentation.

2. ***Claim 50***

- a) **Claim 50: “A speech recognition circuit according to claim 46, wherein each processor compares the processed speech parameters with the lexical data in a corresponding lexical memory to identify words as a word recognition event and sends information identifying the identified words to said results memory as the processing results.”**

91. The '319 Patent describes a speech recognition circuit that searches a lexicon for a highest-scoring possible word, characterizing the occurrence of recognizing a word as a “word recognition event.” ’319 *Patent*, 2:18-30 (“...to identify words as a **word recognition event...**”), 6:40-57 (“...a **word end event** is reached and a word is **recognized.**”). *Thelen* teaches selecting a final recognized words based on the output of recognizers, and thus teaches “compar[ing] the processed speech parameters with the lexical data...to identify words as a word recognition event[,],” as claimed.

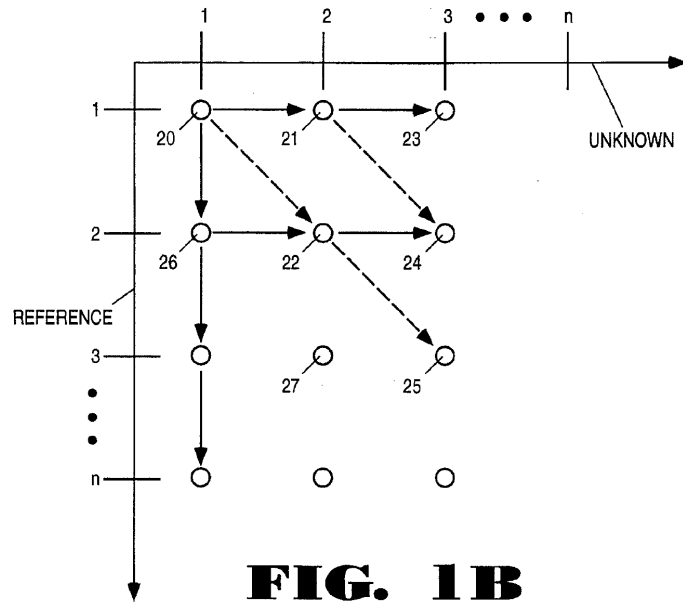
92. Further, as explained above, *Thelen*+*Bailey*+*Chen* teaches a circuit wherein each processor 102 in clusters 100 of *Chen* (acting as the pattern recognition engines 525 of *Bailey*) recognizes speech using a recognition model for a particular context from *Thelen*. See Section VI.C.2-VI.C.4. When a pattern comparison performed by the processors is complete, PR engine 525 generates an interrupt to a

controller (*Bailey's* CPU 510 operating pursuant to *Thelen's* control processor 350), which "generates a transfer of the results from the PR engine to the system memory 512[,]" the mapped "results memory." See Section VI.C.6 citing *Bailey*, 11:42-59, 12:41-61. Here, a POSITA would have recognized that when the system transmits an interrupt indicating the completion of a comparison task for transfer of the results, a "word recognition event" is occurring. Specifically, *Thelen* characterizes its output step as "mak[ing] the final selection of a recognized word sequence." *Thelen*, 7:50-58. *Bailey* directly characterizes transfers of data from PR engine as "events," leading a POSITA to understand that the interrupt performed by the combined system upon final word recognition is a "word recognition event," as claimed. A POSITA would have been motivated and found it obvious to modify *Thelen's* speech recognition circuit to generate word recognition events to assist *Thelen's* controller 350, notifying it as soon as possible that results were available for determining the final recognized word, speeding the recognition. A POSITA would have had a reasonable expectation of success for the combination at least because *Bailey* teaches that interrupts are "typically used to indicate completion" of a word recognition event, indicating to a POSITA that utilizing interrupts would have been a well-known, appropriate modification to *Thelen's* circuit.

3. ***Claim 67***

- a) **Claim 67: “A speech recognition circuit according to claim 46, wherein said lexical memories store said lexical data as Hidden Markov Models, and each processor performs the Viterbi search algorithm using a respective part of said lexical data.”**

93. As discussed above in my Background of the Technology, Hidden Markov Models were acoustic models well-known prior to the '319 Patent for representing speech. *Supra*, ¶¶ 45-48. As explained by *Schmandt*, “An HMM can be used to represent a word with internal states representing characteristic acoustic segments, possibly phonemes or allophones. The output of a state is a frame or vector of acoustic parameters or features...the Viterbi algorithm is a commonly used method of solving this classification problem[.]” *Supra*, ¶ 48 citing *Schmandt*, 59-60. The utilization of HMMs in the circuit of *Thelen+Bailey* reflects these well-known concepts. Specifically, *Thelen* teaches that each recognition model is characterized by a Hidden Markov Model (HMM) generated from training data. *Thelen*, 5:29-42. Further, *Bailey* explains that the Viterbi score procedure is used on HMMs to “compute the highest probability that a given set of observation points matches a reference pattern.” *Bailey*, 7:51-8:1. *Bailey* expressly discusses Viterbi search algorithm being applied as illustrated in Fig. 1B, reproduced below.



Bailey, Fig. 1B, 8:2-36. Though not expressly stated in *Thelen*, the well-known nature of HMMs being solved using the Viterbi algorithm and the description of *Bailey* would have indicated to a POSITA that the circuit described by *Thelen* would have used the “commonly used” Viterbi method of classifying unknown patterns of input acoustic data using HMMs.

94. Specifically, a POSITA would have been motivated to configure the *Thelen+Bailey+Chen* speech recognition circuit to perform a Viterbi search algorithm using each of the lexical processors to maximize the benefit gained and maintain consistency between the lexical processors, ensuring consistency and operability between the groups of processors. A POSITA would have recognized that the processors of *Thelen+Bailey+Chen* circuit, as modified, perform a Viterbi search algorithm “using a respective part of said lexical data,” as claimed, because

each of the processors in the modified system would have only utilized the lexical data stored in its respective lexical memory 615a-c, corresponding to the recognition models from *Thelen*. A motivation for combining the speech recognition circuits of *Thelen* and *Bailey* is provided above for limitation 46(b), and is incorporated herein by reference.

IX. OPINIONS REGARDING GROUND 2

A. Claims 47-49 Are Obvious Over *Thelen* In View Of *Bailey* And *Chen* In Further View Of *Huang*

1. Claim 47

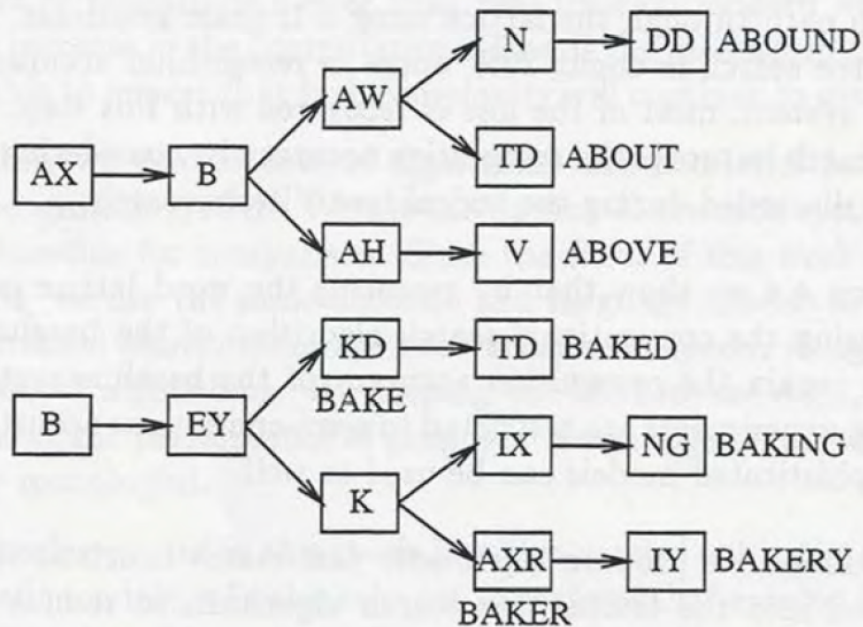
- a) **Claim 47: “A speech recognition circuit according to claim 46, wherein each lexical memory stores lexical data as at least one lexical tree data structure, each lexical tree data structure comprises a model of words having common prefix components, and an initial component of each lexical tree data structure is unique.”**

95. The circuit of *Thelen+Bailey+Chen* comprises a plurality of lexical memories, each containing recognition models corresponding to particular contexts but does expressly teach structuring its lexical data in a lexical tree data structure. As mentioned in my Background of the Technology discussing *Young*, a lexical tree was a well-known data structure for efficiently storing lexical data in prior art speech recognition circuits. *See*, ¶34 *citing Young*, 11:37-53, Figs. 5-6. A lexical tree stores data representing recognizable words in a lexicon in a cascading manner that reduces the required amount of storage by grouping all words sharing phonemes, splitting

words only as their phonemes diverge. This arrangement further provides an efficient structure for searching, scoring spoken HMMs representing phonemes against a plurality of words sharing those phonemes simultaneously. Per *Ravishankar*, “[t]he single largest source of computational efficiency” in searching HMMs is “organizing the HMMs to be searched as a phonetic tree, instead of the flat structure[.]” *Ravishankar*, 67. *Ravishankar* even provides an example of how a flat data structure (i.e., merely a list of the recognizable words) is arranged into a lexical tree data structure in Fig. 4.1 below:

ABOUND	AX B AW N DD
ABOUT	AX B AW TD
ABOVE	AX B AH V
BAKE	B EY KD
BAKED	B EY KD TD
BAKER	B EY K AXR
BAKERY	B EY K AXR IY
BAKING	B EY K IX NG

(a) Example Pronunciation Lexicon.



(b) Basephone Lexical Tree.

Figure 4.1: Basephone Lexical Tree Example.

Ravishankar, 68.

Another example of this well-known lexical tree structure is found in *Huang*, teaching a lexicon of “all words” organized in a “prefix tree,” which a POSITA would have recognized as at least one “lexical tree data structure” comprising a

“model of words[.]” *Huang*, 2:31-58. The following citation from *Huang* makes it clear that its prefix tree is indeed the well-known, typical, and expected structure of a “lexical tree,” as claimed:

In such a prefix tree, **each subword unit (such as a phoneme) is typically represented by a branch** which is associated with a particular acoustic model (such as an HMM). The phoneme branches are connected, at nodes, to subsequent phoneme branches. **All words in the lexicon which share the same first phoneme share the same first branch.** All words which have the same first and second phonemes share the same first and second branches. By contrast, **words which have a common first phoneme, but which have different second phonemes, share the same first branch in the prefix tree but have second branches which diverge at the first node in the prefix tree**, and so on. The tree structure continues in such a fashion such that **all words likely to be encountered by the system are represented by the end nodes of the tree (i.e., the leaves on the tree).**

Huang, 2:38-52. A POSITA would have understood each branch of a lexical tree, when separated, follows the same cascading structure, splitting words only as their phonemes diverge. Thus, each branch of *Huang*’s prefix tree itself represents a lexical tree, branching from a common initial node to terminating “leaves” representing words. Thus, though claim 47 refers to each of these distinct, branching data structures as a plurality of “lexical data tree structures,” a POSITA would have recognized *Huang*’s “branches” teach claim 47. *Huang*, 2:31-52, 7:21-27, 10:16-33, 12:43-46, *see also* 5:62-6:17, 8:8-9, and Figs. 1-2.

96. A POSITA would have been motivated by *Ravishankar's* description of “[t]he single largest source of computational efficiency” in searching HMMs being “organizing the HMMs to be searched as a phonetic tree, instead of the flat structure” to employ *Huang's* lexical tree data structure in the circuit of *Thelen+Bailey+Chen*, using a known technique improving similar devices in the same way. *Huang* itself states that lexical trees “reduc[e] the magnitude of computation required for speech recognition[,]” leading a POSITA to modify *Thelen's* HMM lexicon 134 into a lexical tree data structure. *Huang*, 2:31-37. This would have required only steps performable by a POSITA without undue experimentation of reprogramming of data stored in memory. Storing lexical data as a tree would have been desirable to a POSITA as a method of mapping sub-word units of words (e.g. phonemes) to nodes in the tree, efficiently organizing the lexical data for the “search and match” step of speech recognition. I note that *Thelen* and *Bailey* each already teach multiple processing units (*Thelen's* “recognizers” and *Bailey's* “PREs”) running in parallel to achieve better recognition results without increasing overall processing time between utterance and recognition, indicating to a POSITA that a lexical tree structure would have been an effective, desirable method of organizing the lexical content to efficiently distribute processing across multiple processors (as recited in Claim 46), since different branches could have been assigned to different processors or processor groups. *Thelen*, 2:30-37, 7:17-29;

Bailey, 11:9-34, FIG. 3A. A POSITA would have had a reasonable expectation of success for the modification of *Thelen's* HMM lexicon 134 into a lexical tree data structure because it was so well known and only required reprogramming to cause a rearrangement of data already present in the system, desirably increasing computational efficiency.

2. ***Claim 48***

- a) **Claim 48: “A speech recognition circuit according to claim 47, wherein each lexical memory stores said lexical tree data structure as an n phone model of words, where n is an integer, and said components comprise n phones.”**

97. *Huang* describes a speech recognition circuit comprising a training phase during which a monophone model of words is built, generating the lexical tree data structures stored in lexical memory subsequently used for speech recognition. *Huang*, 7:15-54, 8:16-20, 11:64-12:46, *see also* 6:62-7:18, Figs. 1-2 (teaching monophone storage). Here, a “monophone” model represents words using a single phone (mono=1), where 1 is an integer. Each phoneme in *Huang's* lexicon is modeled based on incoming training data as monophones which “are used in the recognition scheme to determine a most likely matching phoneme for an input speech utterance[.]” *Huang*, 7:45-54. After intermediate steps of dynamically generating context dependent triphone models, which I will discuss in detail below

for claim 49, the model is used for speech recognition, searching a prefix tree as described above for claim 47. *Huang*, 11:64-12:46.

98. *Huang's* lexical tree data structure has “components” (i.e., common prefix components) that are phonemes on each branch of *Huang's* lexical tree. Specifically, a POSITA would have recognized these are monophones because *Huang* describes them as not exceeding the total number (40-50) of phonemes used in the English language. *Huang*, 2:53-61, 8:24-25. A POSITA would have known that there are 40-50 **monophones** in the English language, with exponentially more for each n-phone, as n becomes larger. The '319 Patent itself admits that “If there are 50 mono phones, there are potentially 50^n lexical tree data structures. For tri-phones, this comprises 50^3 (i.e. 125,000).” This would have matched a POSITA's understanding – the English language has about 50 phonemes, and since each of the 3 phones (the preceding phone, the current phone, and the subsequent phone) being accounted for in a triphone model may be any of the 50 phonemes, there should be about $50^3 = 125,000$ triphones. Thus, a POSITA would have recognized that *Huang's* description is of a monophone model, teaching and rendering obvious a lexical tree data structure as an n phone model of words, where n is an integer, and said components comprise n phones.

99. In my opinion, a POSITA would have been motivated to store the lexical tree data structure described in claim 47 as a monophone (an “n phone”)

model of words to minimize the amount memory required to store the lexical data, requiring only reprogramming of data stored in memory, performable by a POSITA without undue experimentation. As explained for claim 47, formatting data in a lexical tree (and storing it in such a form) was well-known as being advantageous and desirable in speech recognition circuits prior to the '319 Patent, as attested by the prior art, leading to a reasonable expectation of success for the combination. *Ravishankar*, 68. Of course, to maintain consistency, a POSITA would have configured the system to store each of the recognizable words in the circuit as an n phone model of words (where n is an integer, and said components comprise n phones, as claimed), maximizing the benefit gained.

3. *Claim 49*

- a) **Claim 49: “A speech recognition circuit according to claim 48, wherein each lexical memory stores said lexical tree data structure as a mono phone model of words, and said lexical tree processors use said mono phone models to generate context dependant phone models of words dynamically for use in processing the speech parameters.”**

100. As discussed above for claim 48, *Huang* describes a speech recognition circuit comprising a training phase during which a monophone model of words is built, generating the lexical tree data structures stored in lexical memory subsequently used for speech recognition. *Huang*, 7:15-54, 8:16-20, 11:64-12:46. The '319 Patent teaches, in support of claim 49, that a monophone (or “mono

phone”) model is used to determine a plurality of triphones associated with its lexical tree data structure, dynamically generating context dependent phone models. ’319 Patent, 6:40-7:35, 9:7-16. Here, a POSITA would have understood that “context dependent” and “context dependant” (as claimed) are synonymous.

101. Specifically, *Huang* teaches comparing incoming speech to its monophone model to obtain likely phonemes. *Huang*, 11:64-12:8, 7:45-54. After further intermediate processing regarding senone states, *Huang* then details how the circuit dynamically generates context dependent phone models:

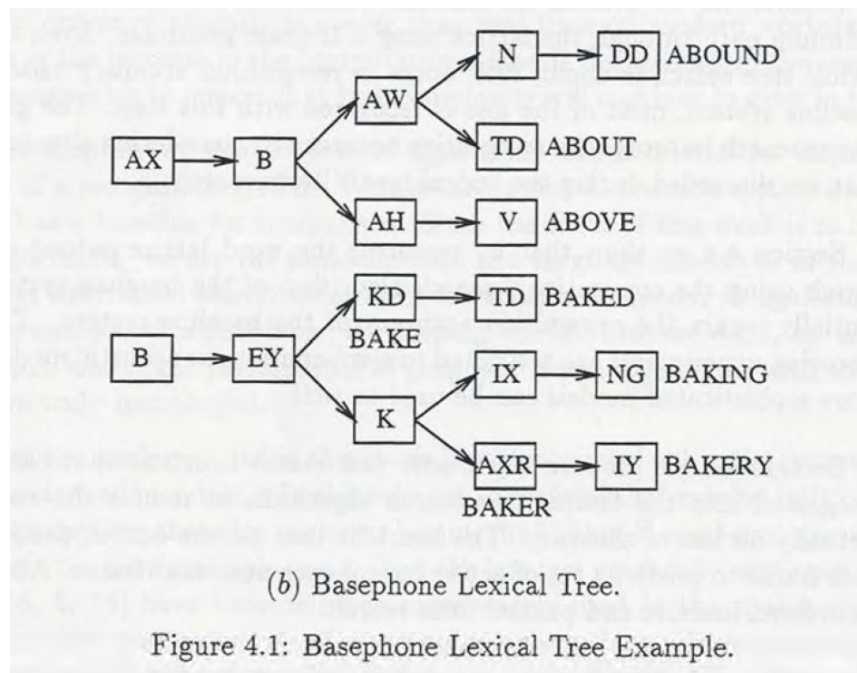
Search engine 80 then accesses the information stored in memory 74 and retrieves a likely triphone which has been mapped to the likely senone sequence determined. This is indicated by block 172.

Search engine 80 then determines whether all of the likely phonemes have been processed. If not, search engine 80 repeats the process and arrives at a likely senone sequence for each likely phoneme (and thus arrives at N likely triphones to be associated with the target phoneme) based upon the probability scores determined during comparison. This is indicated by blocks 174 and 176.

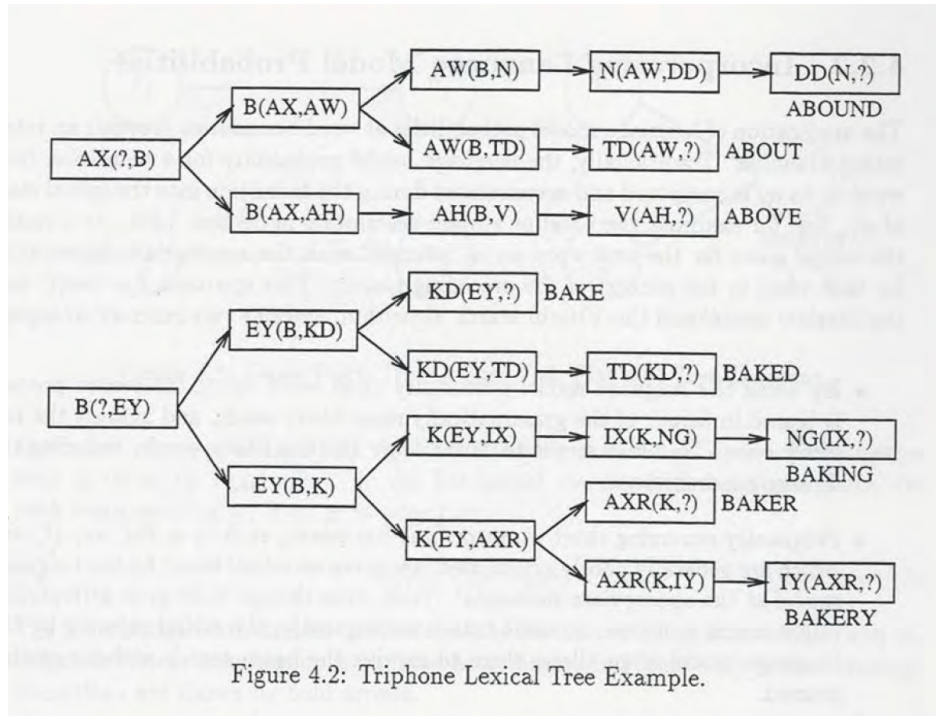
Once the N likely triphones have been identified, search engine 80 accesses the prefix tree in memory 76.

Huang, 12:32-46. *Huang* expressly characterizes a triphone as “a phoneme with corresponding right and left **context**[,]” leading to a POSITA’s understanding that using monophone models to generate triphone models, as taught by *Huang*, teaches using a mono phone model to generate context dependent phone models for words, as claimed. *Huang*, 7:64-66.

102. As seen in *Ravishankar*, context dependent models based on triphones were well-known prior to the '319 Patent to increase the accuracy of speech recognition circuits, leading a POSITA to a reasonable expectation of success for combining *Huang's* dynamic generation of context dependent models into the circuit of *Thelen+Bailey+Chen* and motivating the combination. *Ravishankar*, 70-71 (“...we wish to use triphone acoustic models rather than simple base phone models for high recognition accuracy. Hence, the lexical tree has to be built out of triphone nodes...”). As seen in the figures from *Ravishankar* below, it was a well-known technique to use a basephone lexical tree to dynamically generate triphone models during recognition.



Ravishankar, 68.



Ravishankar, 71.

103. A POSITA would have recognized this would have constituted using a known technique (using monophone models to determine triphone models) to improve similar devices (the speech recognition circuits of *Thelen+Bailey+Chen* and *Huang*) in the same way, requiring only reprogramming of the functionality of the speech recognition circuit using hardware already present, performable by a POSITA without undue experimentation. The result would have been, predictably, increasing accuracy provided by accounting for the preceding phone as taught by *Huang* without greatly increasing the required computational load. Specifically, to maintain consistency and maximize the benefit gained by the modification, a

POSITA would have modified the speech recognition circuit to generate context dependent triphone models using each of the lexical tree processors.

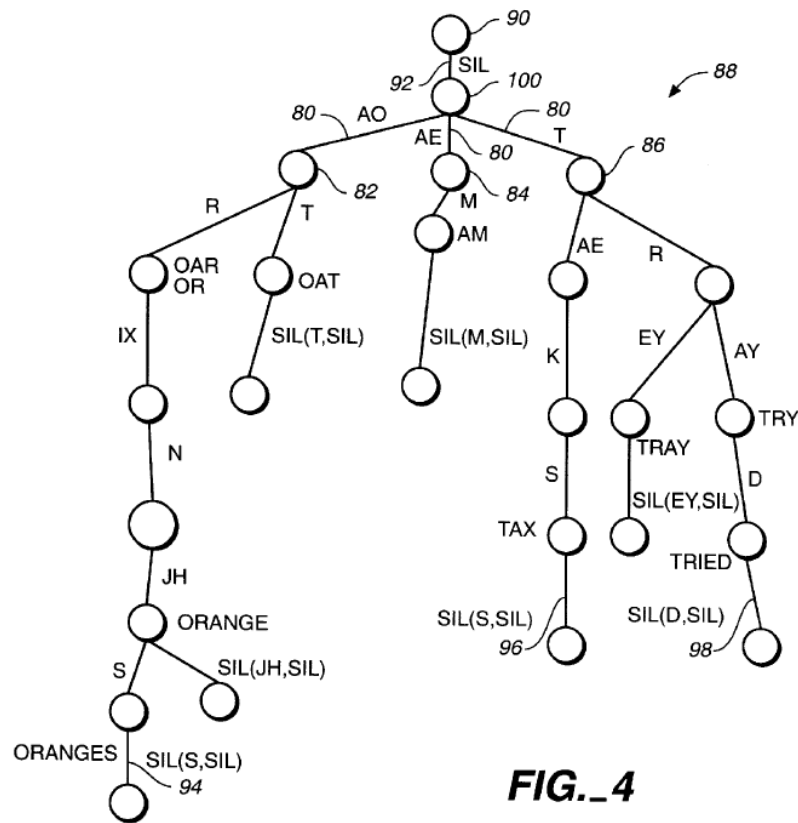
X. OPINIONS REGARDING GROUND 3

A. Claims 53 And 58-59 Are Obvious Over *Thelen* In View Of *Bailey* And *Chen* In Further View Of *Jiang*

1. *Dependent Claim 53*

- a) Claim 53: “A speech recognition circuit according to claim 46, wherein each processor determines and outputs scores for words in the processing results at word recognition events.”**

104. *Jiang* teaches a speech recognition circuit comprising a CPU 21, “which may include one or more processors” employing a lexical tree (or “prefix tree”) associating phonetic models, such as those modeled by HMMs, with the branches of the prefix tree. *Jiang*, 2:47-59, 4:8-9, 6:39-42, 7:34-38. *Jiang* searches the prefix tree in a “known manner,” calculating scores for pruning until reaching a “leaf node” representing a final word recognized spoken utterance (such as “orange,” in the example described at 8:16-64). *Jiang*, 2:3-15, 1:19-32, 8:16-51, Fig. 4. Specifically, *Jiang* computes “probability scores” for each state of an acoustic model stored in memory, such as the Hidden Markov Model. *Id.* In this way, *Jiang*, like the circuit of *Thelen+Bailey+Chen*, determines and outputs scores for words in its processing results.



Jiang, Fig. 4. *Jiang* refers to the action of outputting determined words at word recognition events, as “word boundaries.” *Jiang* 7:48-64, 8:65-9:3, 9:28-42. When silence is detected between spoken words, the lexical tree is searched for the previous spoken word. Once a leaf node is reached, the score for the recognized words would have been output to the control processor.

105. A POSITA would have been motivated to modify the *Thelen+Bailey+Chen* to output scores at word recognition events per *Jiang* to determine a final recognized word in an efficient and operable manner using each of

the plurality of processors of the circuit. Doing so would have had a reasonable expectation of success because it would have provided the control processor with the information necessary for determining a final “winning” recognized word from amongst *Thelen’s* plurality of words recognized by the plurality of recognition models. This arrangement would have allowed the competing recognized words to be fairly and quickly judged, speeding the circuit’s operation and motivating the modification. A POSITA would have recognized that the scores would have needed to be output at word recognition events to have serve appropriate scores to the control processors for judgement.

2. Dependent Claim 58

- a) **Claim 58: “A speech recognition circuit according to claim 46, wherein said processors determine and output scores in the processing results during the processing of said speech parameters.”**

106. As explained above, *Thelen+Bailey+Chen* renders obvious a speech recognition circuit comprising processors that determine scores at word recognition events. *See* Claim 50. Further, *Jiang* teaches a speech recognition circuit that determines and outputs scores *during* the processing of speech parameters, in order to enable the increased efficiency provided by pruning. In detail, *Jiang* teaches a “pruning technique” employed during a lexical tree search accomplished by “comparing, at each node, the scores assigned to that node, with the largest score on any of the other nodes corresponding to the frame being considered. If

the score at a particular node is sufficiently low compared to the maximum score for other corresponding nodes in tree 77, it is assumed that the branches leading to the node under consideration (the node with the lower score) will not likely be part of the completed most likely phoneme sequence (i.e., the phonemes in that sequence will not likely be part of the final word recognized by the system). Thus, **that branch is dropped (or pruned) from tree 77 and is no longer considered in further processing.**” *Jiang*, 8:52-64, *see also* 2:25-46, 10:51-11:15. *Jiang* is describing a technique, well-known and widely-used by POSITAs, of reducing the number of candidate words being considered by removing any word candidate that falls below a threshold. Specifically, *Jiang* teaches that scores are determined “**during the processing of said speech parameters,**” in order to obtain the “computational savings” afforded by no longer processing pruned branches, making the system “more efficient” by “reduc[ing] the search space more quickly,” which a POSITA would have realized would have hastened the speech recognition process and potentially enabled the system to utilize less sophisticated (and thus potentially less expensive) processors. *Jiang*, 10:51-64, 11:8-24. Thus, a POSITA would have been motivated to configure the circuit of *Thelen+Bailey+Chen* to determine and output scores during the processing of said speech parameters, as claimed, in order to gain the “computational savings” taught by *Jiang*. This would have constituted a known technique (determining scores for pruning) being used to improve similar devices

(the speech recognition circuits of *Thelen+Bailey+Chen* and *Jiang*) in the same way, leading a POSITA to a reasonable expectation of success for the combination.

XI. OPINIONS REGARDING GROUND 4

A. Claims 51 Is Obvious Over *Thelen* In View Of *Bailey* And *Chen* In Further View Of *Katz*

1. *Dependent Claim 51*

- a) **Claim 51: “A speech recognition circuit according to claim 50, a including language model processor providing a language model output for modifying the processing results at a word recognition event by a said processor.”**

107. *Thelen* teaches a sentence level matching system 140 for applying an n-gram language model at the sentence level to constrain the recognition such that “word sequences which are proper sequences as specified by the language model.”

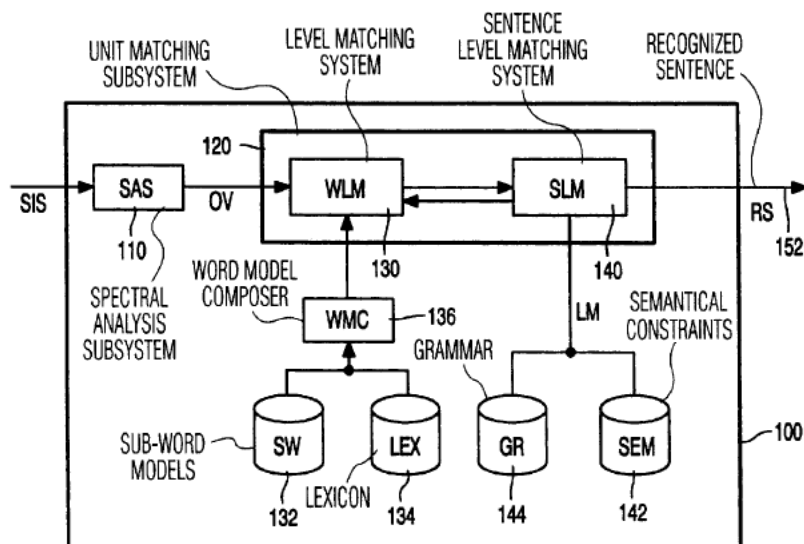
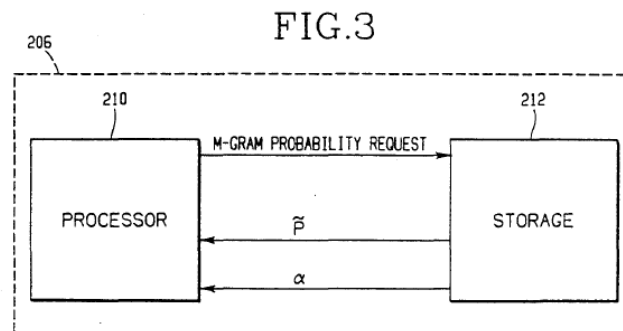
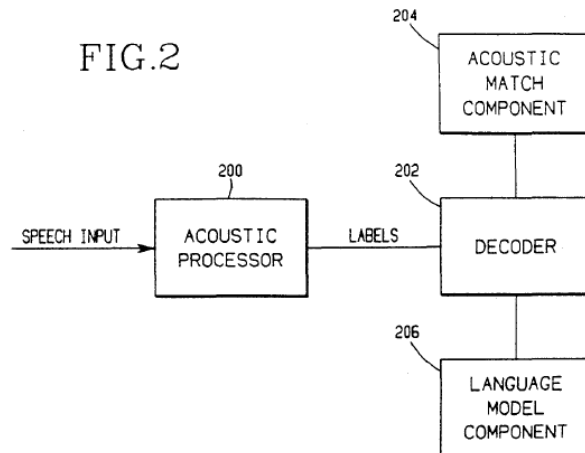


FIG. 1

Thelen, 4:20-25, 5:54-6:10, Fig. 1. Since this language model is applied as a particular word is recognized, a POSITA would have recognized *Thelen's* language model is applied “at a word recognition event,” as discussed above regarding claim 50. Specifically, *Thelen* describes determining a “recognized sequence” of words by combining the processing results of word recognition (the “results of the acoustic model” comparison) with results from the language-level comparison, teaching the claimed “providing a language model output for modifying the processing results at a word recognition event.” *Thelen*, 5:54-6:10. Of course, this sentence-level improvement to *Thelen's* recognition process is only performed in addition to the individual word-level recognition, rather than in replacement or as an alternative to word-level recognition. *Thelen*, 7:30-43. *Thelen* does not, however, expressly describe a language model processor dedicated to the task of applying the language model functionality provided by sentence level matching system 140, though a POSITA would have recognized a processor would have performed this task in *Thelen's* system.

108. *Katz* does expressly teach a dedicated language model processor via processor 210, which, alongside storage 212, comprises its “language model component 206,” as illustrated in Figs. 2-3 below.



Katz, 6:33-41, 1:44-54, 2:34-44, Figs. 2-3. Like *Thelen's* sentence level matching system 140, *Katz's* language model component 206 functions to determine a probability that a selected vocabulary word is the next word in a string, given that the previous words in the string are known based on a sample text of phrases (such as three word combinations known as "tri-grams"). For this reason, a POSITA would have been motivated and found it obvious based on *Katz* to modify the *Thelen+Bailey+Chen* circuit to execute the language model comparisons at the sentence-level via a language model processor, as claimed. A POSITA would have recognized and desired the processing advantage granted by delegating this task to

a language model processor and would have been capable of doing so by simple combination of known hardware elements using known techniques. *Katz* and *Thelen* each already describe systems based on tri-gram language models, leading to a reasonable expectation of success for the combination without undue experimentation. *Thelen*, 5:54-6:11; *Katz*, 1:44-54.

XII. CONCLUSION

109. For the aforementioned reasons, in my opinion, to a POSITA, the claims of the '319 Patent are rendered obvious by the prior art.

110. I hereby declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I understand that willful false statements are punishable by fine or imprisonment or both. See 18 U.S.C. § 1001.

Date: December 2, 2022

Respectfully submitted:

By: 

Christopher Schmandt

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

- *Facebook v. Windy City*, 2017
- *Facebook v. EVERYMD.COM LLC*, 2017
- *Comcast v. Promptu*, 2017
- *Express Mobile, Inc v. Big Commerce*, 2017
- *Facebook v. Hypermedia*, 2017
- *Microsoft v. Uniloc 2017 LLC*, 2019
- *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

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"
- 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 on grammar"

- 20030158903 "Calendar bar interface for electronic mail interaction"

Publications

Most publications can be found at <http://www.media.mit.edu/speech/publications> or <http://living.media.mit.edu/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.

Amazon / Zentian Limited

Exhibit 1003

Page 111

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. Angebrannndt, 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.
