

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

DIALECT, LLC,
Patent Owner.

IPR2025-01193
Patent: 7,917,367 B2
Issued: March 29, 2011
Application No. 12/617,506
Filed: November 12, 2009

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

DECLARATION OF PAUL JACOBS

MICROSOFT CORP.
EXHIBIT 1003

TABLE OF CONTENTS

	Page
I. INTRODUCTION, OVERVIEW, AND ENGAGEMENT	5
II. BACKGROUND AND QUALIFICATIONS	5
III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '367 PATENT	10
IV. UNDERSTANDING OF PATENT LAW	12
V. GROUNDS	15
VI. STATE OF THE ART	16
A. Networked Systems Regularly Used Some Form Of Device Registration To Track And Communicate With Devices	16
B. Event Subscription Services Were Commonly Used To Update Network Devices Or Components With New Information.....	20
C. Conventional Natural Language Speech Processing Used Context ...	22
VII. THE '367 PATENT.....	24
A. Prosecution History Of The '367 Patent	26
VIII. LEVEL OF SKILL IN THE ART	26
IX. CLAIM CONSTRUCTION	27
A. Claim Construction Order For Related '039 Patent.....	27
X. GROUND 1: CLAIMS 11-12, 15 ARE OBVIOUS OVER BELFIORE	27
A. Belfiore (EX1004).....	28
B. Obvious Implementations Of Belfiore's Teachings.....	33
C. Ground 1 Claim Mappings.....	38
1. Claim 11	38

a)	Element [11.1] (Preamble)	38
b)	Element [11.2]	41
c)	Element [11.3]	51
d)	Element [11.4]	53
e)	Element [11.5]	55
2.	Claim 12	56
3.	Claim 15	57
XI.	GROUND 2: CLAIMS 11, 12, 15, 17, AND 18 ARE OBVIOUS OVER KENNEWICK (EX1005) AND BELFIORE (EX1004)	60
A.	Kennewick	60
B.	Combining The Teachings Of Kennewick And Belfiore	61
1.	Device Synchronization	61
2.	Multi-Modal Interfaces	63
3.	The Combined Kennewick-Belfiore System	65
4.	Claim 11	68
a)	Element [11.1] (Preamble)	68
b)	Element [11.2]	69
c)	Element [11.3]	69
d)	Element [11.4]	69
e)	Element [11.5]	70
5.	Claim 12	71
6.	Claim 15	71
7.	Claim 17	72

8.	Claim 18.....	89
XII.	GROUND 3: CLAIMS 17 AND 18 ARE OBVIOUS OVER KENNEWICK, BELFIORE, AND ROSS (EX1007).....	91
A.	Ross	91
B.	The Combined Kennewick-Belfiore-Ross Implementation.....	92
C.	Ground 3 Claim Mappings.....	93

I, Paul Jacobs, do hereby declare as follows:

I. INTRODUCTION, OVERVIEW, AND ENGAGEMENT

1. I have been retained as an independent expert on behalf of Microsoft Corporation in connection with the above-captioned Petition for *Inter Partes* Review (“IPR”) to provide my analyses and opinions on certain technical issues related to U.S. Patent No. 7,917,367 (hereinafter “the ’367 patent”).

2. Specifically, I have been asked to provide my opinions regarding whether claims 11-12, 15, and 17-18 (each a “Challenged Claim” and collectively the “Challenged Claims”) of the ’367 patent would have been obvious to a person having ordinary skill in the art (“POSITA”) as of August 5, 2005. It is my opinion that each Challenged Claim would have been obvious to one of ordinary skill after reviewing the prior art discussed herein.

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

II. BACKGROUND AND QUALIFICATIONS

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

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

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

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

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

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

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

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

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

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

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

13. From my initial academic training in artificial intelligence, which was supervised by Marvin Minsky at MIT's Artificial Intelligence Laboratory while I was completing my Master's work at Harvard, through my career in commercial and academic R&D and as a graduate instructor, I was closely involved in the advances in speech and language technology. I visited Verbex, the predecessor of Dragon Systems, while at Harvard, made multiple invited trips to Bell Laboratories and while at Berkeley regularly participated in seminars and collaboration with SRI, where I witnessed the birth of the company that later became Nuance, to which the '367 patent was assigned (the commercial heritage of Nuance is confusing, because the name belonged to another company called Scansoft that was in a different business, but Nuance's speech technology that was later used in Siri came from the SRI spinoff that was purchased by Nuance). As a DARPA principal investigator, I participated regularly in some of the first Speech and Language Workshops (sometimes known as Human Language Technology workshops) where we collaborated and tested against shared "benchmarks". As mentioned above, I later collaborated with Nuance and others in integrating question answering technology with speaker-independent spoken language recognition.

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

the capabilities of one of ordinary skill—and was so in August 2005 and November 2009—I am qualified to opine on what one of ordinary skill would have known and understood as of the priority date of the '367 Patent.

**III. MATERIALS CONSIDERED AND
INFORMATION RELIED UPON REGARDING '367 PATENT**

15. In preparing this declaration, I have reviewed the following materials bearing Exhibit Nos. that I understand correspond to exhibits that are being filed by counsel for Petitioner:

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 7,917,367 (“the ’367 patent”)
1002	File History of U.S. Patent No. 9,917,367
1004	U.S. Patent Application Pub. No. 2002/0059425 to Belfiore et al. (“ Belfiore ”)
1005	U.S. Patent Application Pub. No. 2004/0044516 to Kennewick et al. (“ Kennewick ”)
1006	Redline showing differences between text, aside from the claims, of Kennewick (EX1005) and 367 patent (EX1001).
1007	U.S. Patent Application Pub. No. 2002/0133354 to Ross et al. (“ Ross ”)
1010	Akiyoshi Ochi, <i>et al.</i> , <i>Network Applications for Mobile Computing</i> , Fujitsu Sci. Tech. J., 34(1), pp. 41-49 (Sept. 1998)
1011	Robin Garner, <i>Pelican DHCP Automated Self-Registration System: Distributed Registration and Centralized Management</i> ,

No.	Description
	Proc. of the LISA 2001 15 th Sys. Admin. Conf., pp. 257-266 (Dec. 2001)
1012	Adam Rifkin & Rohit Khare, <i>The Evolution of Internet-Scale Event Notification Services</i> , Workshop on Internet-Scale Event Notification (July 13, 1998)
1013	Antonio Carzaniga, <i>et al.</i> , <i>Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service</i> , PODC '00: Proc. Nineteenth ACM Symp. Principles of Distributed Computing, pp. 219-227 (2000)
1014	Frederick Jelinek, <i>The Development of an Experimental Discrete Dictation Recognizer</i> , Proc. of the IEEE, 73(11), pp. 1616-1624 (1985)
1015	Andreas Stolcke, <i>et al.</i> , <i>Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech</i> , Computational Linguistics, 26(3), pp. 339-373 (Sept. 2000)
1016	E. Levin, <i>et al.</i> , <i>The AT&T-DARPA Communicator Mixed-Initiative Spoken Dialog System</i> , Sixth Int'l Conf. on Spoken Language Processing (2000)
1017	International Pub. No. WO 01/78065 to Weber et al.
1019	Andrew S. Tanenbaum, <i>Computer Networks</i> (Pearson Education, Inc., 4 th ed. 2003)
1020	L. Rau et al., <i>Information Extraction And Text Summarization Using Linguistic Knowledge Acquisition</i> , Information Processing & Management Vol. 25, No. 4, pp. 419-428 (1989)
1021	Paul S Jacobs, <i>Joining Statistics with NLP for Text Categorization</i> , Third Conference on Applied Natural Language Processing, Association for Computational Linguistics, pp. 178-185, Mar. 31, 1992-Apr. 3, 1992

No.	Description
1023	Paul S. Jacobs, <i>A Knowledge Framework for Natural Language Analysis</i> , IJCAI 87 Proceedings of the Tenth International Conference on Artificial Intelligence, 675-678 (1987)

IV. UNDERSTANDING OF PATENT LAW

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

17. I understand that, when considering the scope of the claims of a patent, the patent claim terms should generally be given the ordinary meaning that the terms would have to a person of ordinary skill in the art in question after reading the patent as of the earliest claimed priority date.

18. I understand that the person of ordinary skill in the art (“POSITA” or “person of ordinary skill,” for short) is deemed to read the claim term not only in the context of the particular claim in which the term appears, but in the context of the entire patent, including the other claims and the specification of the application as filed. I further understand that the principal considerations regarding the scope and meaning of the claims are the plain language of the claim (including the surrounding claim language and context), the patent specification, and the prosecution history.

19. I understand that while a claim is to be read in light of the specification, one must generally avoid importing limitations into the claim from the specification.

I am also informed that the prosecution history can often inform the meaning of the claim by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be. I applied these understandings when considering the scope and meaning of the claims of the '367 patent.

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

21. I further understand that a claim would have been obvious if it unites old elements with no change to their respective functions, or merely substitutes one element for another known in the field, and that combination yields predictable results. While it may be helpful to identify a reason for this combination, I understand that there is no strict requirement of finding an express teaching, suggestion, or motivation to combine within the references. When a product is available, design incentives and other market forces can prompt variations of it,

either in the same field or different one. If one of ordinary skill can implement a predictable variation, obviousness likely bars its patentability. For the same reason, if a technique has been used to improve one device and one of ordinary skill would recognize that it would improve similar devices in the same way, using the technique would have been obvious. I understand that a claim would have been obvious if common sense directs one of ordinary skill, without hindsight bias caused by knowledge of the challenged claims, to combine multiple prior art references or add missing features to reproduce the alleged invention recited in the claims.

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

Ground 3	Kennewick, Belfiore, Ross (EX1007) ³	§ 103 – obviousness	17-18
-----------------	---	---------------------	-------

VI. STATE OF THE ART

A. Networked Systems Regularly Used Some Form Of Device Registration To Track And Communicate With Devices

27. Distributed applications that exchanged information content with various networked devices were in widespread use prior to the popularization of the World Wide Web in 1993. For example, many organizations deployed Lotus Notes® to share e-mail, databases, spreadsheets and other content and to synchronize that content among numerous users on personal computers, laptops, and other devices. In addition to authenticating users and allowing them to share information, Notes provided the ability for servers and client devices to maintain replicas of databases and their contents, ensuring that consistency of data was maintained across networks.

28. The Web itself eventually replaced and upgraded such proprietary technology for organizing and distributing content, as the Web led very rapidly to standards that would be quickly accepted by many millions of users and many thousands of application developers. These standards provided an economical framework for distributed content. For example, in 1994 I remember being

³ U.S. Patent Application Pub. No. 2002/0133354.

personally involved with at least three commercial distributed information systems – a photo-sharing app by a company known as Picture Network International, a “Homework Helper” app by Infonautics, and a news processing system that later became known as NetOwl® . All of these distributed content to multiple users on a variety of platforms, and within a few years in the 1990’s moved from proprietary standards (such as Prodigy® in the case of Homework Helper) to Web-based platforms.

29. By at least 1998, the Web was well known and widely deployed, as were mobile devices. It had long been well known that network communications involved keeping track of the various user devices on the network to properly communicate with them. EX1010, 43 (“A mechanism is required to verify users attempting to access the network.”), 47 (“When considering the means of access [...] [a] mobile computer...requires registration in the mainframe computer beforehand.”). Sometimes referred to as registration or authentication or both, this functionality has long allowed networks to ensure that messages are getting to and coming from properly registered / authenticated devices, and only such devices. EX1019, 282 (“When a new host enters an area ... his computer must register itself with the foreign agent there.”); *see also id.*, 353; EX1011, 262 (describing registration process that includes verifying username and password submitted by the device before registration); *see also id.*, 257-263 (describing detailed examples of

registration workflows and communications). As I have explained above, this messaging often involved synchronization, such as in Lotus Notes.

30. The Web supports registration and authentication via standards at multiple levels. The well-known IP (Internet Protocol) layer, which predates the web by about 30 years, identifies devices by IP address. In the application layer, HTML (using HTTP) permits devices to exchange content, and through that layer devices and/or users are often authenticated, such as by password. Many of the basic tools for keeping track of registered devices and users, such as different types of “cookies” which helped to keep track of users, maintain consistent user information for a “session” in which a user interacted with content, and update the appropriate content, stem from those first couple of years of the Web, mainly 1994. Apple came up with its first mobile Web browser that same year.

31. DHCP (Dynamic Host Control Protocol) was a 1993 standard that allowed a central access point or server to assign IP addresses to remote devices. The early Web was notoriously weak on security, but many organizations quickly moved to address those issues using registration and/or authentication mechanisms. EX1011, 257 (“Some institutions enable nomadic computing by allowing anonymous access to their DHCP services but Tufts’ IT governing committees have forbidden anonymous access to our networks; all devices must be registered with the DHCP server(s) in order to be granted a lease.”); EX1019, 282 (“When a new host

enters an area, either by connecting to it (e.g., plugging into the LAN) or just wandering into the cell, his computer must register itself with the foreign agent there.”), 353 (“When a mobile host shows up at a foreign site, it contacts the foreign host there and registers.”). It was also well-known to remove registered devices when the devices left the network or after a period of time (e.g., when a validity of the registration expires). EX1019, 283 (“Ideally, when a host leaves an area, that, too, should be announced to allow deregistration, but many users abruptly turn off their computers when done.”), 354 (“Another problem that had to be solved is what to do about impolite mobile hosts that leave without saying goodbye. The solution is to make registration valid only for a fixed time interval. If it is not refreshed periodically, it times out, so the foreign host can clear its tables.”); EX1011, 263 (“Automated client expiration prevents the accumulation of stale data in the database. Every week a cron-initiated script is run which deletes any client whose registration has expired.”). Accordingly, network-related protocols often employed a registration and / or authentication scheme to control access to networks and track devices on the network, such that devices securely and properly interacted within the network infrastructure. For example, automated DHCP self-registration tools allowed mobile devices to register with a network in a secure manner to allow the devices to utilize the services of the network. *E.g.*, EX1011, 262 (describing registration process that includes verifying username and password submitted by the

device before registration; “After verifying that the given username and password correspond to a valid email or shell account, Pelican extracts authorization information from the University LDAP system. The authorization information is used to derive the user’s DHCP class (“student” or “staff”) and affiliation (department, school, etc.). Pelican confirms that this class of user is permitted to register on the client’s subnet.”); *see also id.*, 257-263 (describing detailed examples of registration workflows and communications). Thus, one of ordinary skill in the 1990s, and certainly in 2005, understood that network access typically involved registration/authentication of devices to ensure that the devices are known and able to be tracked by the network, to update and distribute content across these devices, and to safeguard that these devices were involved only in authorized and appropriate communications.

32. The ’367 patent seems to recognize that skilled artisans knew to register mobile devices, as it only describes device registration in passing (EX1001, 13:26-29) with no details on the manner of doing so.

B. Event Subscription Services Were Commonly Used To Update Network Devices Or Components With New Information

33. Event-based notifications have long been commonly used to facilitate changing the state of networked devices based on certain events and, relatedly, synchronizing the state of different devices on the network. *E.g.*, EX1012, 15 (describing examples of event notifications for “maintaining awareness of people,

devices” such as “Changes in state occur on 0(minutes)”), 19 (“Notifications are usually small and stateful (for example, delta updates)”). Many users of the Internet in 1996 remember the rapid development of “push” technology on the Internet, particularly a system called PointCast, which updated Internet users by “pushing” news updates into their screensavers. PointCast was somewhat infamous in corporations at the time because, while it used local computer resources only when they weren’t active for other purposes, it voraciously consumed precious Internet bandwidth. Of course, in order to make such technology work, event information was exchanged between local devices and servers. The server received a notification that a device was in screen-saver mode, along with user profile information, and the local device received a news update tailored to that user’s profile.

34. Thus in the 1990s, event notification services in networking environments provided notifications of events to clients within both local area networks and wide area networks to maintain awareness of people and devices on the networks. EX1012, 2 (describing how, in the past, “Event Notification Services (ENS) expanded their range, from hosts to LANs to WANs.”). These event notifications were typically part of a service that allowed clients to subscribe to events of interest and receive notifications of published events from other clients. *Id.*, 28 (describing the lifecycle of an event as including “[e]xpression of interest in an event or pattern[,] [o]ccurrence of an event[,] [o]bservation[,] [r]elation of an

event to a pattern of interest[,] [n]otification to an application.”); EX1013, 219 (“SIENA is responsible for *selecting* the notifications that are of interest to clients (as expressed in client subscriptions) and then *delivering* those notifications to the clients via access points.”), Figure 1 (showing a distributed event notification service and an “advertise,” “subscribe,” “publish,” “notify” model.). Therefore, one of ordinary skill understood that networks typically provided services allowing devices to subscribe to event notifications to receive updates relating to other devices in the network.

C. Conventional Natural Language Speech Processing Used Context

35. As early as 1985, natural language processing (“NLP”) systems were known to use statistical and rule-based models to incorporate contextual information for improved recognition and understanding of spoken language. *E.g.*, EX1014, 1616 (A “self-organized, statistical approach underlying the basic algorithms of [a dictation] recognizer” where “the recognizer uses only algorithms also applicable to continuous speech recognition.”). For example, a basic concept of natural language processing involves the use of n-gram language models that predict a next word in a sequence based on previous word(s). *Id.*, 1622 (“Consistent with the letters guessed so far, the possible words of the vocabulary are shown together with their probabilities as computed by the trigram language model (given the preceding two words of the text).”) Since a vocabulary of words recognizable by a speech

recognition system may be large in a natural language environment (e.g., where a user provides speech input in a natural speaking manner), such statistical techniques were known to be used to probabilistically reduce the amount of searching of the vocabulary for matches to spoken utterances. *Id.*, 1619 (“The problem of speech recognition can be thought of as one of finding a complete tree path (that corresponds to one definite string of words W) that is most probable given the acoustic evidence A [...] [t]he search is efficient if relative few paths are examined.”).

36. My own research emphasized knowledge-based approaches to language generation and interpretation. EX1023. By the late 1980s and early 1990s my team had begun to combine some of the statistical learning models that were by then common in spoken language research with our knowledge-based methods to effectively learn new words and summarize information using context. EX1020; EX1021.

37. Prior to 2005, it was also common for natural language processing systems to use dialog acts and histories to provide context for speech recognition models to improve accuracy and efficiency of natural language speech processing by constraining the grammar or language models used for recognizing speech input based on a context of the dialog. *E.g.*, EX1015, 340: (“The ability to model and automatically detect discourse structure is an important step toward understanding spontaneous dialogue. [...] [S]ome agreement exists that a useful first level of

analysis involves the identification of dialogue acts (DAs). [...] [A] speech recognizer could be constrained by expectations of likely DAs in a given context, constraining the potential recognition hypotheses so as to improve accuracy.”); EX1016, 1 (“[S]poken language systems are typically implemented by [...] limiting the scope of the underlying language models and grammars for ASR.”), 3 (“In system-initiative dialogs, the grammar (or language models) can be constrained to the few possible inputs expected at this point in the dialog thereby increasing the underlying ASR performance.”). Further, it was known to use domain-specific grammars that correspond to a topic of discussion (that is, the topic part of the conversational context) in prior queries by a user (e.g., “news,” “weather,” “stocks,” etc.) based on a prior utterance. *E.g.*, EX1017, 10:22-11:1 (“examples of contexts may be ‘news’, or ‘weather’, or ‘stocks’.”), 10:7-8 (“For example, the statement ‘I would like to see movies,’ would be linked to the context of ‘movies’ or ‘film.’”), 20:4-6 (“when the next utterance is provided to the speech recognition processor 200 in block 300 of FIG. 3A, it would search the enabled grammar 212 for ‘movies’ based on the dialog context before searching the general grammar 214.”). Therefore, one of ordinary skill understood that natural language speech processing typically incorporated contextual information to facilitate the recognition and interpretation of the speech input.

VII. THE '367 PATENT

38. The '367 patent issued March 29, 2011, from an application filed November 12, 2009, and alleges priority to an application filed August 5, 2005. EX1001, 1. It is titled “Systems and Methods for Responding to Natural Language Speech Utterance.” *Id.*

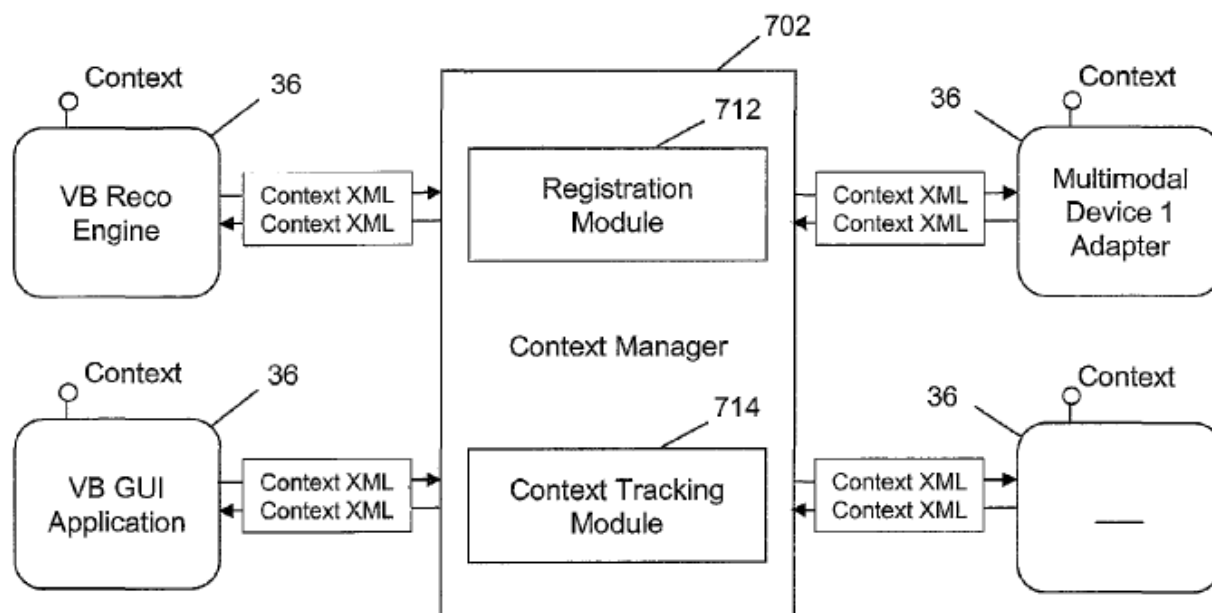
39. The '367 patent describes a system including a main unit, a speech unit, and a multi-modal device. EX1001, 11:36-37. The main unit may process input commands and questions to create a response string for presentation via the speech unit or multi-modal device. EX1001, 12:56-64, 13:7-22.

40. The '367 patent describes one embodiment in which a “context manager 702” communicates with “multiple mobile devices 36.” EX1001, 13:23-27. The multiple devices may “register via registration module 712” and “registration may indicate events that the mobile devices 36 may subscribe to.” *Id.*, 13:26-29. “[M]obile devices 36 may be informed of a context changes [sic] through context tracking module 714, thereby synchronizing the context across the registered mobile devices 36.” *Id.*, 13:31-34.

41. As shown in FIG. 7, the “registration module 712” and “context tracking module 714” are depicted as black boxes within the “context manager 702.” The '367 patent provides no meaningful detail on the structure of these modules or how they carry out their functions.

FIG. 7

700



EX1001, FIG. 7.

A. Prosecution History Of The '367 Patent

42. I have reviewed the file history of the '367 patent, however, at this time, the file history does not inform the meaning of the claims or prior art.

VIII. LEVEL OF SKILL IN THE ART

43. One of ordinary skill at the earliest alleged priority date in August 2005 would have had a bachelor's degree in electrical engineering, computer science, computer engineering, or equivalent field, and two years of experience working with speech recognition and natural language processing systems. Additional work experience could make up for less education and vice versa. In this Declaration, I

use the terms “one of ordinary skill” and “skilled artisan” as synonymous with the “person of ordinary skill in the art.”

IX. CLAIM CONSTRUCTION

44. Unless otherwise discussed below, I have been instructed to apply the plain and ordinary meaning of all claim terms. I understand that Petitioner reserves the right to argue in any district court case or other proceeding that terms in the ’367 patent are indefinite or otherwise, and to raise additional issues of claim construction.

A. Claim Construction Order For Related ’039 Patent

45. I understand that certain claim terms appearing in claims 17 and 18 are similar to or the same as terms appearing in claims of related patents, and have been construed by a court as indicated below. As explained below, the claims are obvious at least under these constructions.

Term	Construction for ’039 patent in E.D. Va.
context	“the subject matter to which a particular user input is directed and which is used to determine the meaning of the user input” (EX1008, 1)
context stack	“a data structure that stores or references contexts in a manner that can be ordered” (EX1008, 1)
context description grammar	“a data structure containing entries constituting or referencing sets of rules, wherein each of those sets describes the structure of natural language in a particular context” (EX1008, 2)

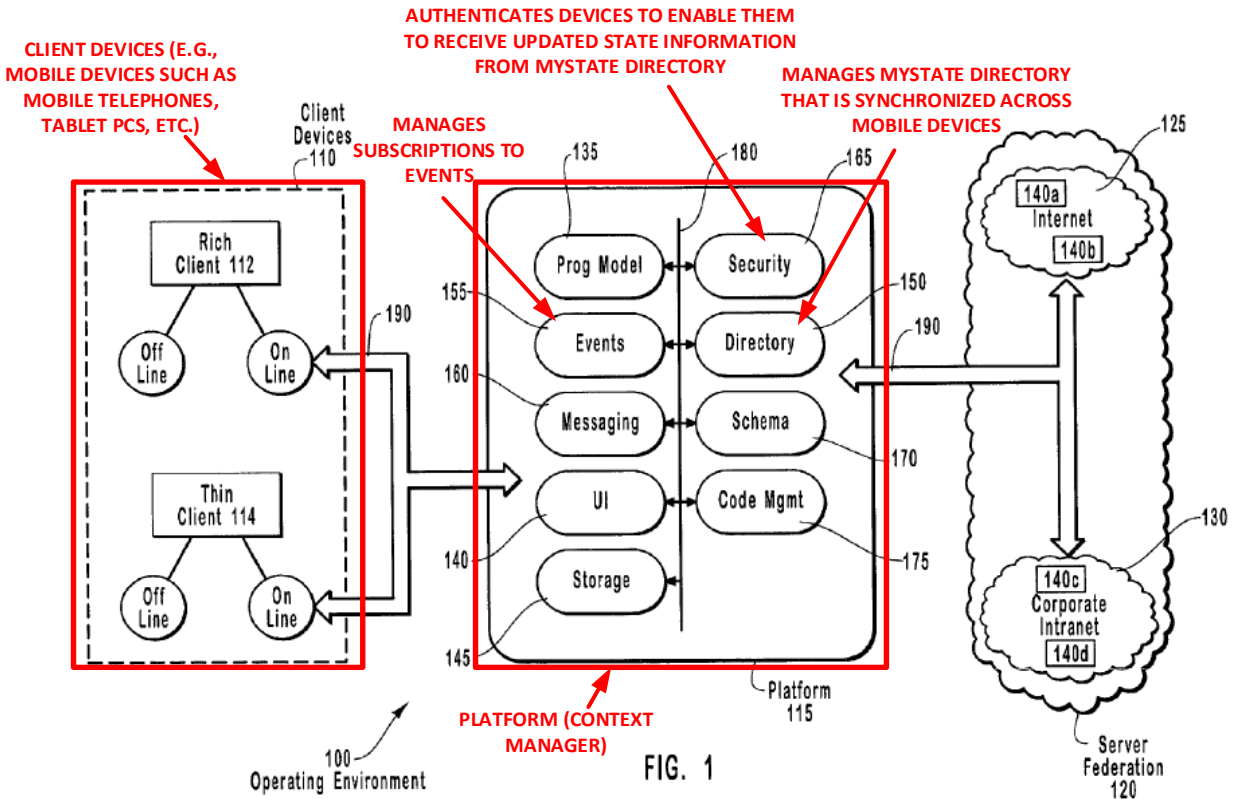
X. GROUND 1: CLAIMS 11-12, 15 ARE OBVIOUS OVER BELFIORE

46. As explained below, claims 11-12 and 15 are obvious over Belfiore. Indeed, claims 21-35 of Belfiore describe a system and method that is strikingly similar to what was later claimed in claims 11-12 and 15 of the '367 patent.

A. Belfiore (EX1004)

47. Belfiore (EX1004) is a patent application that published May 16, 2002, and which I understand to qualify as prior art under at least 35 U.S.C. § 102(b). Microsoft Corporation is the named assignee. EX1004, 1. Belfiore is analogous art to the '367 patent at least because it describes systems and methods for improved distributed computing services, including synchronizing of user and session information across mobile devices with multi-modal natural language user interfaces. *E.g., Id.*, Title, [0003], [0019], [0083], [0095-96], [0104], claims 21-35.

48. Belfiore discloses “a distributed computing services platform” that, among other things, provides “more consistent user interaction and collaboration” and “more personalized, relevant information to users.” *Id.*, [0003], [0014], [0109]. Belfiore’s Platform 115 includes various components (e.g., user interface (“UI”) 140, Directory 150, Events 155, Security 165, Schema 170, etc.), as shown in FIG. 1:



EX1004, FIG. 1 (red annotations added).

49. For example, as described in Belfiore, the client devices may include mobile devices, such as mobile telephones, tablet PCs, etc. EX1004, [0093], [0099]. The security component 165 authenticates the client devices before preference and session information are provided to them. *Id.*, [0098] (“When the user switches to another client device, after proper authentication using security component 165, the preference and session information are provided to the new client device.”) The directory component 150 includes adaptive directories, such as the MyState directory, which maintains updated status information. *Id.* (“[T]he preference information may be stored in the MyState adapted directory 905 in the

server federation 120. As a user works on a client device, the session status information is regularly updated in the MyState directory 905.”), *see also* [0203] (“the ‘MyState’ adapted directory 905, discussed above and provided by the directory component 150.”). The event component 155 manages subscriptions to events, such as updates to the MyState directory. *Id.*, [0184] (“Through use of the event component 155 and messaging component 160, the MyState directories are also capable of notifying other entities of changes and updates to the MyState directory...”).

50. Belfiore describes a “multi-modal” user interface (EX1004, [0118]) that works with various input / output types including “typed (keyboard), spoken, hand-written,” and any of these may be used to “perform a natural language query or command.” *Id.*, [0104-06].

51. Belfiore teaches synchronizing session status and other information across a user’s multiple devices, such that the user interface “adapts and/or changes across a number of client devices 110.” *Id.*, [0097]. Synchronization occurs through events that can be subscribed to. *Id.*, [0020]. For example, Belfiore teaches events that include “the status of system components, the activity of a user, a measurement of the environment, and so forth.” Belfiore teaches an event component that “facilitates the distributed communication of events between any software component that publishes or generates events (‘event source’) and any software

component that subscribes to or receives event notifications (‘event sink’),” and that this event system “enables software components and architectures to have access to continually updated information about their context.” *Id.*, [0119].

52. The following annotated version of Belfiore FIG. 4 illustrates an implementation of the Belfiore platform carrying out the synchronization of Belfiore claim 21 and its dependents:

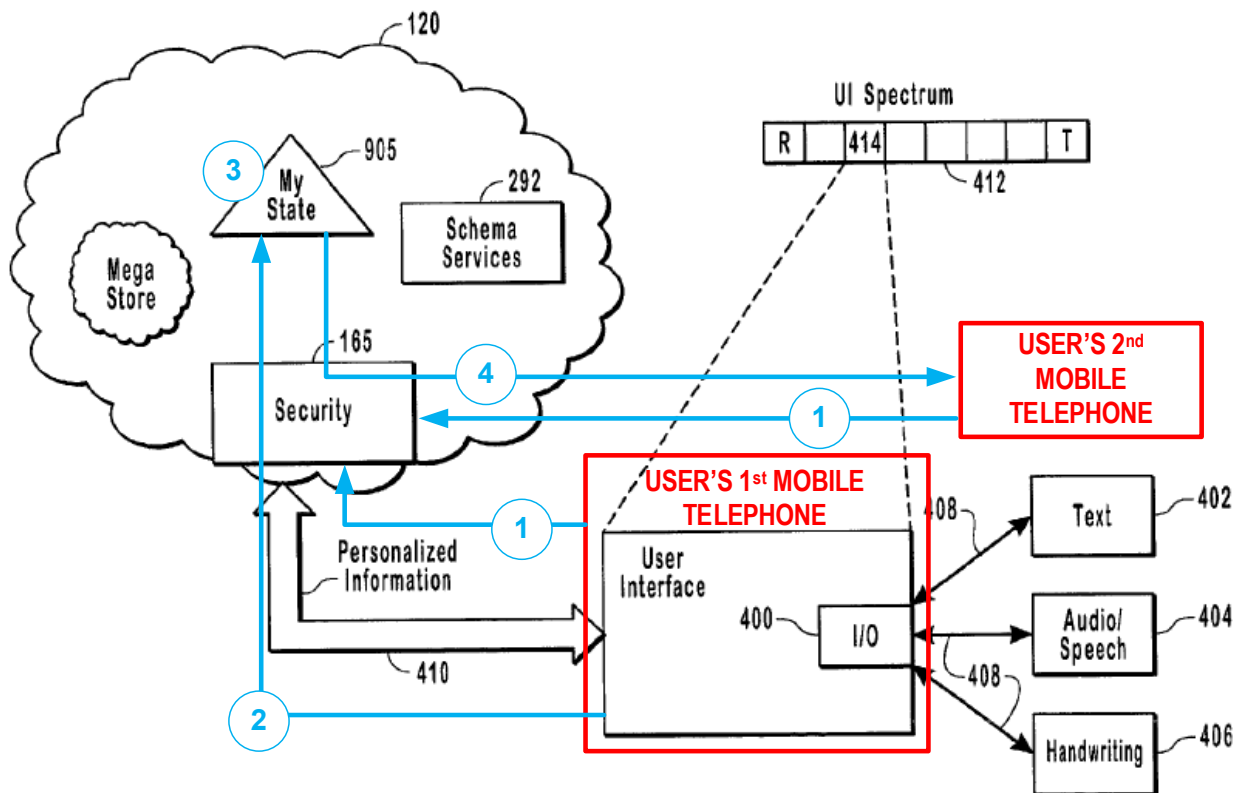


FIG. 4

Circled numbers show process flow for synchronizing context across devices:

- 1 Devices transmit user log-in/authentication information to platform's security component, which authenticates and registers each device (e.g., claims 22, 23)
- 2 Update session and other user information, based on user interactions with first mobile telephone (e.g., claims 21, 34)
- 3 Store the updated user information on the server, e.g., in the user's MyState directory (e.g., claim 21)
- 4 Inform second mobile telephone of the changed user information, thus synchronizing the user's two mobile devices (claims 21, 35)

EX 1004, FIG. 4 (red and blue annotations added).

53. For example, Belfiore describes the user interface component as “allow[ing] multi-modal input and output” from client devices (such as mobile telephones) as shown by text 402, audio/speech 404, and handwriting 406 in FIG. 4. EX1004, [0091], cl. 33-35. The mobile telephones authenticate with the security component, as shown in the FIG. 4 annotations above at “1” in order to register for updating/receiving updates from the MyState directory, as described in more detail below. *Id.*, [0098]. “As a user works on a client device, the session status information is regularly updated in the MyState directory 905,” as shown in the FIG. 4 annotations above at “2” and “3.” *Id.* “When the user switches to another client device, after proper authentication using security component 165, the preference and session information are provided to the new client device,” as shown in the FIG. 4 annotations above at “4.” *Id.* This propagation of the updates to the MyState directory thus synchronizes the context across mobile devices, so that the user interface “adapts and/or changes based on the user’s state and context across a number of client devices 110” and so that when a user switches to a new client device, that new client device is able to “continue in the session where the old client device left off.” EX1004, [0097-98].

B. Obvious Implementations Of Belfiore’s Teachings

54. One particular implementation appears in Belfiore claim 21 and its dependent claims, which disclose synchronizing user information and session status

across multiple mobile devices as part of “a method for facilitating a mid-session transition between [a] first client device and [a] second client device.” Claim 21 includes:

retrieving state information stored on at least one server; interacting in a session with the first client device resulting in an update of the state information; storing the updated information on the at least one server; retrieving the updated state information when the session is continued on the second client device; and interacting in the continued session with the second client device based on the updated state information.

55. EX1004, cl. 21. Dependent claims provide further details, such as the first and second client devices both being mobile telephones (claims 34-35) and authenticating the user on each mobile device (claims 22-23).

56. **One obvious implementation of Belfiore** follows from claim 21 and its dependent claims, as informed by Belfiore’s figures and other disclosures. That implementation includes:

- At least **two mobile client devices**, such as the “first mobile telephone” and “second mobile telephone” disclosed in Belfiore claims 34 and 35. *See also* EX1004, [0041], [0044], [0099].
- The mobile client devices providing user interaction (*id.*, claims 21-23) through a **multi-modal natural language interface**, such as a “unified

command line” that carries out “a natural language query or command” (*id.*, [0104-05]), where “all forms of input—typed (keyboard), spoken, handwritten may be received by the user interface” (*Id.*, [0106]). *See also id.*, [0104] (“In a preferred embodiment, the unified command line would be provided in some form on every type of device.”).

- A plurality of “**servers**,” as disclosed in claim 21.
- A server **storing “state information”** (*id.*, claim 21), including “user interface information” (*id.*, claim 27), user “**preference information**” (*id.*, claim 38; [0019]), and “**session status**” information (*Id.*, [0097]).
As one example, Belfiore teaches storing this information in its MyState directory. *Id.*, [0098].
- A “**platform**” that “**facilitates communication** between” the mobile client devices and the servers (*e.g.*, *id.*, FIG. 1; [0016]), including communications related to storing state information on the server (*id.*, claims 21 and 39) and retrieving information from the server to the client devices (*Id.*, claims 21, 28, 38).
- A security component with an authentication module, for **authenticating** a user on **each mobile client device** (*id.*, claims 22-23), for example based on the “**security component 165**” receiving a **message** with the user’s **Global ID** and / or “**authentication**

credentials” from each mobile client device (*Id.*, [0199]-[0202], FIGS. 4, 10).

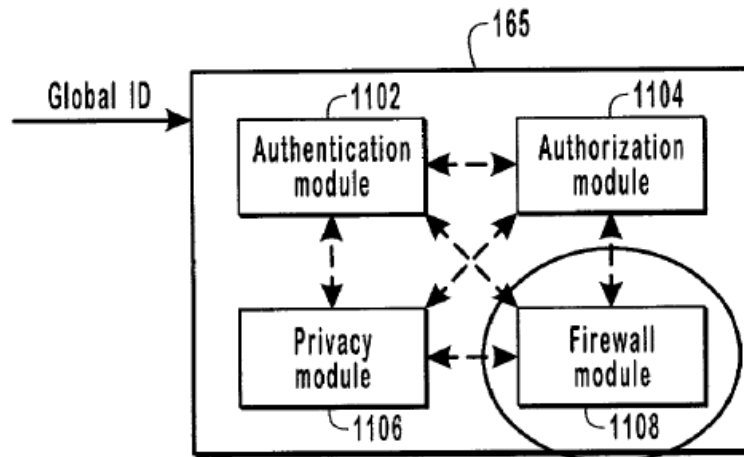


FIG. 10

- A directory component for *inter alia* keeping track of a user’s currently authenticated devices (claims 22-23), e.g., by adding information on a device a user has logged into and removing that information if a user logs out of the device (EX1004, [0023], [0098]), thus allowing for communications with a user’s currently authenticated devices (claim 21).
- An **event component** used to **subscribe each of a user’s current devices** (or the software running thereon) to a group subscription providing updates based on events indicating a change in information on any device. “An ‘event subscription’ is a request that the event

infrastructure notify the subscribing component when a certain condition occurs.” EX1004, [0133]. “The event component 155 provides that events can be distributed, registered and accessed across the server federation 120 and client devices 100.” *Id.*, [0124]; [0136] (“[A]s new machines enter the group, the subscription is automatically applied to them.”); *see also id.*, [0114] (describing event triggered “when an individual logs onto a particular device”).

- **Synchronizing state information** between authenticated mobile client devices, e.g., via Belfiore’s group subscription. One of ordinary skill would readily appreciate group event subscriptions as a way of **updating state information** at the server based on a user session at one mobile device (claim 21) and **retrieving updated** state information to another device when a user continues the session on the other device (claim 21). Indeed, Belfiore teaches that “events within the platform are used to synchronize” (EX1004, [0020]) and using events to carry out state synchronization achieves Belfiore’s stated purpose of “facilitating a mid-session transition between” devices (claim 21 preamble) and carry out the steps of storing updated information to the server and retrieving that information to the second device (claim 21). *See also id.*,

[0098] (“As a user works on a client device, the session status information is regularly updated in the MyState directory 905.”).

57. Belfiore either expressly contemplates these features/steps in a single embodiment, by reciting them in claim 21 and its dependents, or teaches using them in providing a platform that facilitates the mid-session device-to-device transitions that is the purpose of Belfiore claim 21. As explained in the following section, these and other obvious variants of Belfiore render obvious challenged claims 11-12, and 15.

C. Ground 1 Claim Mappings

58. As set out below, Belfiore renders obvious claims 11-12 and 15.

1. Claim 11

a) Element [11.1] (Preamble)

[11.1]⁴ A method for processing multi-modal natural language inputs, comprising:

59. If the preamble is limiting, Belfiore discloses it. Belfiore discloses user interaction through a “user interface (UI)” that “is multi-modal, intelligent and responsive.” EX1004, [0019]. This multi-modal interface provides “natural language communication.” *Id.*, [0096]. Belfiore teaches that preferably every device

⁴ Reference numbers in the format “[claim#.element#]” added throughout.

of the system has “a unified command line” that may “perform a natural language query or command.” *Id.*, [0104]; *see also id.*, [0105] (describing processing of “natural language command”) and [0106] (describing parsing the “query or command”). “All levels of intelligence-keyword, Boolean, and natural language (fragments, noun phrases, and/or well formed sentences) may also be received and processed by the user interface.” *Id.*; *see also id.*, [0092] (describing “speech-to-text technology” and “handwriting-to-text technology”); [0093].

60. The “multi-modal” interface’s multiple types of user input include “keyboard entries or mouse clicks,” “audio/voice input,” “or stylus/touch input.” *Id.* [0019]; *see also id.*, [0090]; [0091] (“The user interface component 140 includes an advanced input/output component 400 to allow multi-modal input and output,” e.g., it “receives text 402,” “audio/speech input,” “stylus/touch input,” and “handwriting input.”). Belfiore FIG. 4 shows exemplary user inputs 402, 404, 406:

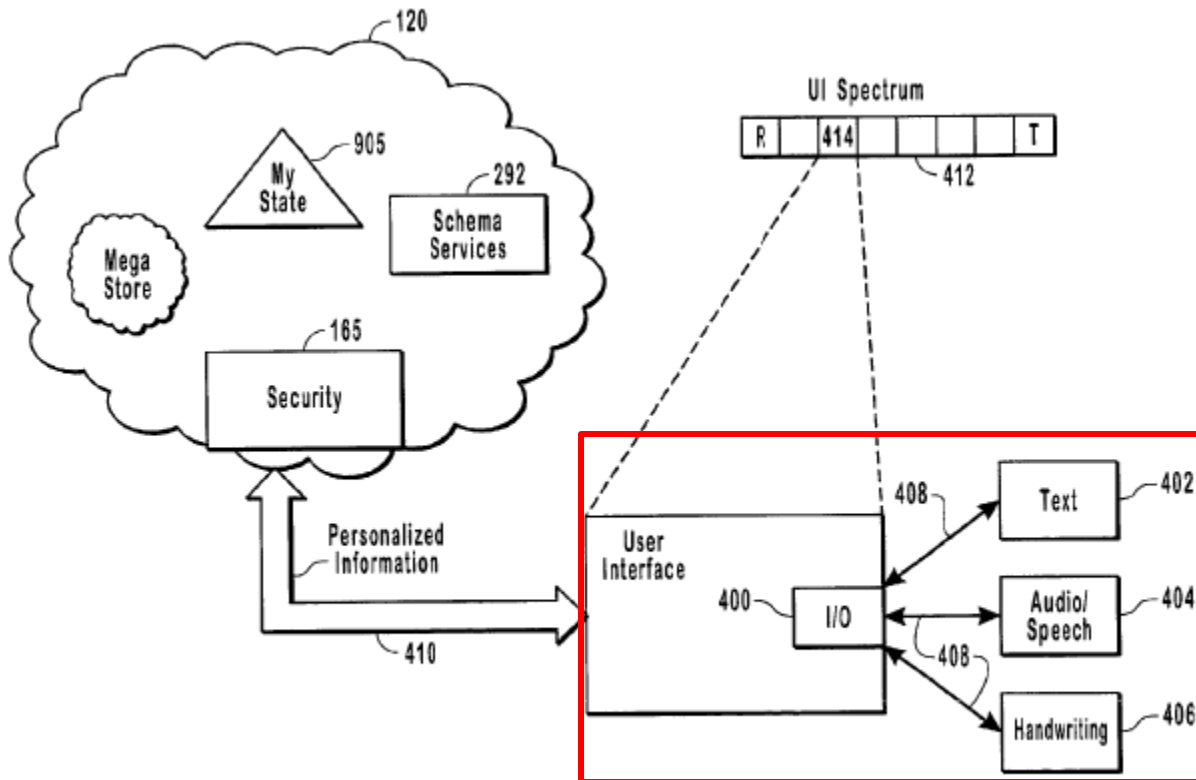


FIG. 4

61. EX1004, FIG. 4 (red annotations added). These multi-modal inputs align with those discussed in the '367 patent. *E.g.*, EX1001, 2:44-45 (“multimodal interactions, such as speech and/or non-speech interactions”); *id.*, 2:50-54 (“multimodal input such as text-based commands and questions and/or voice

commands and questions ...”); *see also supra* Section X.B. As explained in the following sub-sections, this method meets all the elements⁵ of claim 11.

b) Element [11.2]

[11.2] registering a plurality of mobile devices with a context manager in response to a registration module associated with the context manager receiving a communication from the plurality of mobile devices;

62. Belfiore discloses and teaches this element.

63. One of ordinary skill would understand that “registering” a device on a network includes, as an example, identifying that device to the network so that information or services may be exchanged with the device through the network. In the context of mobile phones, for example, this generally means identifying the handset to the mobile carrier so that the phone can send or receive calls and text messages and use the internet. In my brief description *supra* Section VI.A. of the evolution of the Web standards for network modelling, this type of registration on the Web is handled by assigning an IP address to a device and initiating a TCP/IP or UDP session. For example, when a user accesses a remote Web page, the user’s device is uniquely identified to the web server via IP address and a TCP/IP session

⁵ Petitioner uses the terms “limitation” and “element” loosely throughout the Petition and, unless otherwise noted, without taking positions on whether the claim language is entitled to patentable weight or is merely part of the claim’s environment.

allows the web server and that device to exchange information until, for example, a page is complete, even if the session is interrupted. In mobile phones, registration might involve, for example, entering a user ID and password to get your gmail on that phone. This process is more often referred to as “authentication.”

64. In my historical overview above in paragraph 33, I gave the example of the widespread use of PointCast in 1996, where a user’s desktop or laptop would automatically start downloading and displaying news content corresponding to a profile whenever that device went into screensaver mode. This is one example of registration at both the network and application levels – in the network layer, the device is identified by IP address and content is exchanged in the IP session, and at the application level the device is authenticated so that suitable news can be displayed (that is not displayed to other devices that don’t have the application or aren’t in screen saver mode).

65. As explained above, Belfiore teaches authenticating multiple mobile devices (e.g., two mobile telephones) with the Platform (corresponding to the context manager in the ‘367 claims) after the Platform’s security component receives a communication from the devices, e.g., a message with user ID and log-in credentials. *Supra* Section X.B. Belfiore’s “distributed computing services platform” or just “Platform” is the claimed “context manager” because it uses events, MyState, schemas, and other features to manage context as a user carries out a

session on one or more devices. *Id.*; *see also id.*, [0016]. And Belfiore claims 34-35 describe a method with two “mobile telephones” and Belfiore elsewhere teaches mobile client devices such as cellular phones, pagers, and other hand-held devices. *E.g.*, EX1004, [0044] (“client devices 110 include” “cellular phones and pagers” as well as “hand-held or Tablet PCs”); *id.*, claims 21, 34-35; [0045-47], FIG. 1.

66. Belfiore teaches the claimed “registering a plurality of mobile devices ...” at least because its authentication process results in the platform associating the devices with a given user and the platform thereafter is able to appropriately communicate with the devices. *E.g.*, *supra* Section X.B. This conforms with the ’367 patent’s limited disclosure of registering a device. The Belfiore authentication occurs in response to communications from the mobile devices being received at a registration module associated with the context manager. For example, Belfiore discloses “authenticating” a user on a given device in response to the platform’s “security component” (corresponding to the ’367 registration module) receiving log on / authentication communications from the mobile devices. *E.g.*, EX1004, [0098]. After completing this authentication, the platform may provide authenticated devices with a user’s “preference and session information,” e.g., when the user switches from device to device. *Id.*, claims 21-23, [0098], FIGS. 1, 10; *supra* Section X.B. This authentication / registration of a given mobile device involves:

- The security component of the platform receiving messages from the device, e.g., messages including a user's log-in / authentication information, which might include a Global ID and password. *E.g.*, EX1004, [0109], [0200-202], FIGS. 4, 10.
- The security component establishing the device as "authenticated" on the platform, so that the platform can provide updated session status and other user information to the device, e.g., as might be stored in the user's MyState directory. *E.g.*, EX1004, claims. 21-23, [0202]; *supra* Section X.B.
- Recording on the platform, such as in a directory, information indicating that the device has been authenticated by the user, to facilitate communications with that device, such as providing updated session status and other user information. *Supra* Section X.B.

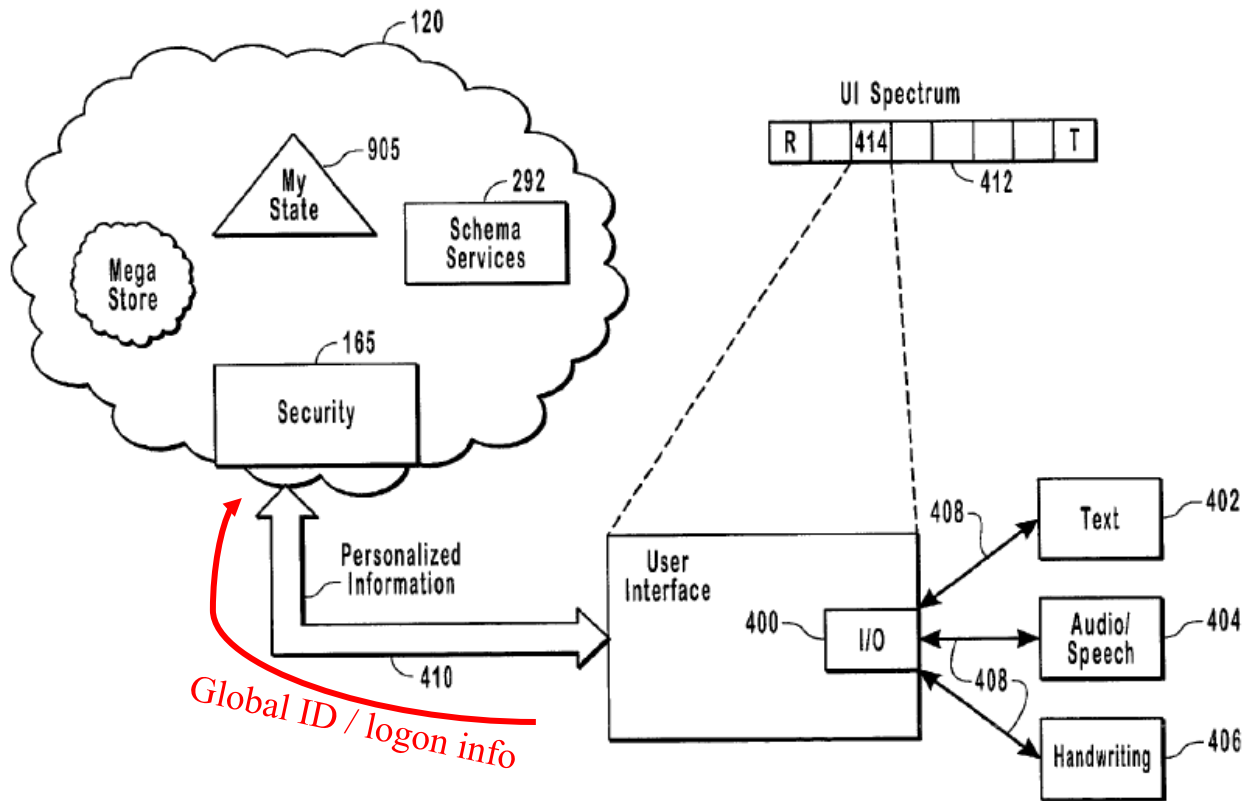


FIG. 4

EX1004, FIG. 4 (red annotations added).

67. The log on / authentication of a given device through the security component registers the device with the platform at least because it allows the platform to securely provide user preference and state information to the device. EX1004, [0175]; *see also id.*, [0109] (explaining how “customization is stored in the MyState directory 905 in the form of preferences” and “[o]nce a user logs onto the client device and authenticates to the security component 165, the federation of server 120 cooperatively obtains personalized information . . . following these preference[s] and provides the information to the user interface over the link.”),

[0175] (explaining how “after the user is authenticated, the directory component 150 may be used to determine where the user’s information is stored” and “[t]he storage component 145 then retrieves the data and provides it to the user interface component 140” so that “as a user moves from device to device, their data and applications follow them.”).

68. Note also that Belfiore expressly teaches using IP and UDP in the security component (*Id.* [0217]), which, for reasons explained *supra* ¶¶ 27-32 and 63-67 is adequate for disclosure of “registering a mobile” device at the internet network level. Belfiore’s platform could not complete a secure transmission to a device (or from a device) if that device were not registered.

69. At the application level, Belfiore teaches “registering” a schema to allow “collaboration” among clients and servers through the Platform (*Id.* [0078]). As with the ‘367 Context Manager, content in Belfiore’s schemas is exchanged in XML format (*Id.* [0077]).

70. One of ordinary skill would recognize that, for a user login at a client device to authenticate the device with the security component of the platform, a communication would be sent between the client device and the platform (e.g., over the communication (network) link 190 of Belfiore). As Belfiore describes, this communication would use an IP address for the device at the internet layer and UDP at the transport layer. Since this authentication allows the client device to receive

personalized information from the platform, one of ordinary skill would recognize that Belfiore's Platform would perform authentication at the application layer (Belfiore, [0109] explaining a process in which "a user logs onto the client device and authenticates to the security component" and is provided with personalized information from the platform, stored in the MyState directory, in response). This confirms that one of ordinary skill would have implemented Belfiore to perform registration in the same secure interaction between the Platform and a mobile device.

71. Thus, skilled artisans would have implemented Belfiore such that user login on a given mobile client device involves registering that device with the platform. Through this registration, the platform would remain aware of the device's authenticated status and able to effectively communicate with the device. This was state of the art for any networked system (particularly for mobile devices), and a skilled artisan would have thus been motivated (if not compelled) to register each user of device. *Supra* Section VI.A; *see also* EX1019, 282 (describing how a new "mobile host" "must register itself"); *id.*, 282-83, 352-54.

72. Device registration would beneficially allow communications with the mobile devices and facilitate the post-authentication functionality specified by Belfiore, such as receiving / providing preferences, session-status, and other user-associated information from / to authenticated devices (and only authenticated devices). *E.g.*, EX1004, [0098] ("When the user switches to another client device,

after proper authentication using security component 165, the preference and session information are provided to the new client device.”); *id.*, claims 21-23; *supra* Section X.B (explaining obvious implementation in which group of authenticated devices provided updates to one another). For example, registering a device would facilitate that device accessing information from the MyState directory (e.g., the preferences, session status, etc.) associated with the user’s Global ID. *E.g.*, EX1004, [0203] (“The MyState adapted directory 905 provides a place to store user profiles containing user data according to an agreed-upon schema. Generally, each Global ID 1220a-1220n provides an index to profile information stored in the individual portions of MyState service.”); *see also id.*, claim 21.

73. Further, registration would beneficially facilitate the system receiving similar information back from the device (e.g., session status information), as also taught by Belfiore. *E.g.*, EX1004, [0098] (“As a user works on a client device, the session status information is regularly updated in the MyState directory 905. When the user switches to another client device, after proper authentication using security component 165, the preference and session information are provided to the new client device.”), [0184] (“Through use of the event component 155 and messaging component 160, the MyState directories are also capable of notifying other entities of changes and updates to the MyState directory as well as terminating a relationship between an entity and the MyState directory.”), claim 21.

74. One of ordinary skill would have had a reasonable expectation of success in implementing the Belfiore system to register client devices upon login from those devices because Belfiore already teaches authenticating such devices and providing information to and from such devices, and further teaches Directory Components and other functionality for maintaining device registration information for a given user. For example, in describing the use of subscriptions to events within the system, Belfiore confirms what any skilled artisan would expect—that “[w]hen a new machine enters the cloud,” it would rely on the exchange and availability of addressable information for that new machine, such as a “routing configuration that instructs the forwarding service where the events should be sent” (i.e., a registration of the new machine), to enable subscriptions to be sent to the new machine. EX1004, [0137]. The cloud in Belfiore is also referred to as a “mega-store,” which is used to store “user preferences and files” that are accessible “from any location using any device” “after proper authentication.” *Id.*, [0022], [0273], *see also id.*, [0175] (“The mega-store is illustrated by the following scenario. A user walks up to a PC and logs on. This log on authenticates the user with the Internet authentication service provided by the security component 165. After the user is authenticated, the directory component 150 may be used to determine where the user's information is stored. ... Thus, as a user moves from device to device, their data and applications

follow them.”). Moreover, device registration was state of the art. *Supra* Section VI.A.

75. For example, one logical location for this client device information would be within the Platform’s Directory structure and, in particular, within a given user’s MyState directory. Belfiore teaches that the MyState directory contains other regularly updated user information, including “name, address, phone, and the like.” EX1004, [0182]. Thus, one of ordinary skill would have been motivated to implement the MyState directory to store the list of client devices on which the user was currently logged in, and to update that directory as the user logged into / logged out of client devices. Doing so would allow a particular user’s information to be provided to those devices that had been authenticated with a given user’s Global ID, as taught by Belfiore. It would also ensure that such information was provided *only to* those devices properly authenticated with the user’s Global ID. Implementing the MyState Directory to maintain this registration information would have required minimal effort as Belfiore already teaches maintaining similar user information in that directory.

76. The benefits and expectation of success in implementing Belfiore to register a given user’s authenticated client devices are clear from the ’367 patent itself, which provides no explanation whatsoever as to why or how registration is performed.

77. Belfiore discloses its platform components as software that runs on computers to carry out certain functions. EX1004, [0041-42], [0046], [0199]. Thus, it discloses one or more computers programmed to manage context and register devices (as described in the preceding paragraphs).

c) **Element [11.3]**

[11.3] subscribing the plurality of mobile devices registered with the context manager to one or more context events;

78. Belfiore teaches events that allow its system components “to have access to continually updated information about their context.” EX, [0119] (“[A]n event is an observation about one or more states such as, for example, the status of system components, the activity of a user, a measurement of the environment, and so forth.”); *see also id.*, [0123] (explaining that events are about context such as “a user’s location, activity, and active devices”). Components both “publish[]” events (serving as “event source[s]”) and “subscribe[]” to events (serving as “event sink[s]”). EX1004, [0119]; *see also id.*, FIGS. 1, 5. An “event component”—much like the ’367 “event manager”—then “facilitates the distributed communication of events between” the different events sources and sinks. EX1004, [0119]; *see also id.*, [0125] (“[A]ny event source or sink [may] publish, subscribe, or listen to events.”). “[E]vents can be distributed, registered and accessed across the server federation 120 and client devices”). *Id.*, [0124].

79. Belfiore also teaches maintaining consistent “state and context across a number of client devices,” e.g., using a MyState directory, so that users may switch between devices mid-session. EX1004, [0097]; *see also id.*, claim 21. For example, “[a]s a user works on a client device,” “preference and session information” may be regularly updated in a user’s MyState directory and then provided to other user devices. *Id.*, [0098]. Devices are synchronized in this fashion only after being authenticated (registered) by a user. *Id.*; *id.*, [0109], claims 22-23.

80. Belfiore further specifies using its event component to notify devices of MyState updates. *E.g.*, EX1004, [0184] (“Through use of the event component 155 and messaging component 160, the MyState directories are also capable of notifying other entities of changes and updates to the MyState directory.”). Accordingly, one of ordinary skill would have implemented Belfiore to subscribe each of a user’s authenticated mobile client devices to MyState update events so that “state and context” were consistent across devices. *Id.*, [0097]. Each mobile device would be subscribed to these events through an “event subscription” requesting notification of changes and updates to the MyState directory. *Id.*, [0133]. Thereafter, any changes or updates to the MyState directory would be sent to the subscribed devices to notify them of the changes and updates to the MyState directory. *Id.*, [0184]. Any mobile device that a user logged into would become both a source and sink for events caused by updates to the user’s MyState directory. As a user carried

out a session on a given mobile device (e.g., the first mobile telephone of claim 34), updates would be provided to their MyState directory and then distributed to other devices logged into by the user (e.g., the second mobile telephone of claim 35). These events thus serve to synchronize “state and context across a number of client devices.” *Id.*, [0097]; *id.*, [0020] (“Events within the platform are used to synchronize.”); *see also supra* Section X.B. These subscribed-to events are “context events” at least because they relate to changes in context, such as a user’s preference or session status. *See also id.*, [0119] (describing how “event system enables” “continually updated information” about “context”); *infra* Section X.C.1.e. The ’367 patent itself never uses the term “context event” outside the claims.

d) Element [11.4]

[11.4] receiving, at the context manager, a context input from one or more of the plurality of mobile devices registered with the context manager, wherein the context input includes a context change event; and

81. As just explained, Belfiore teaches using events to provide updates from a user’s authenticated devices to the user’s MyState directory and other devices. *Supra* Section X.C.1.c. These events are “context inputs” that include “context change events,” as they include information on updates (i.e., changes), to “the user’s state and context.” EX1004, [0097]; *see also id.*, [0085] (describing actions taken based on a “users preferences of [sic] context”); [0119] (“The event system enables software components and architectures to have access to continually

updated information about their context.”). As the event component and MyState directory both are part of the Belfiore Platform (e.g., EX1004, FIG. 1), the events received by these entities are received “at the context manager,” namely Belfiore’s Platform. *Supra* Section X.C.1.b; *see also supra* Section X.B. Belfiore thus satisfies Element 11.4.

82. Furthermore, the events of Belfiore align with the description of context in the ‘367 patent, which states that “the location data can be used by the invention as part of the context” and further describes “identify[ing] context such as time, location,...” EX1001, 7:48-49, 14:14-15. Belfiore similarly explains providing “events, such as an update on the general location of a user” ([0020]) or “a stream of events about a user’s location, activity, and active devices” ([0123]). *See also id.*, [0123] (“[A] stream of events...can be analyzed by a Bayesian model to provide a set of probabilities about a user’s goals.”). When a user moves from an old location to a new location, any event (e.g., observation of state) reflecting the new location would be a context change event (e.g., from the old location to the new location). EX1004, [0123].

83. The registered (i.e., authenticated) client devices, including mobile devices, of Belfiore are described as being capable of communicating with the platform (e.g., including the events component 155 of the platform) and include software components (e.g., user interface, applications, browser, etc.), which are

described in Belfiore as being usable as event sources. EX1004, [0020], [0022], [0044-45], [0051], [0119], [0143], [0251]. Thus, one of ordinary skill would recognize the Platform of Belfiore as receiving context input from the registered mobile devices, the context input including context change events.

e) **Element [11.5]**

[11.5] informing the plurality of mobile devices registered with the context manager of the context change event,

wherein informing the plurality of mobile devices registered with the context manager of the context change event synchronizes a context across the plurality of mobile devices.

84. As explained for Element 11.3, Belfiore teaches using events to provide updates from a user's authenticated devices to the user's MyState directory *and other devices*. *Supra* Section X.C.1.c. And as explained for Element 11.4, these events are "context inputs" that include "context change events." *Supra* Section X.C.1.d. Thus, when these events are used to provide updates to the user's other authenticated mobile devices, the events inform the user's authenticated mobile devices (i.e., the user's registered devices) of the context change event. In informing the mobile devices of these context change events, the Belfiore event component ensures that all devices are "consistent" with one another in terms of user MyState information—such as the "user's state and context" and "session status information" and "preferences" (EX1004, [0097-98]), thus synchronizing a context across the

plurality of mobile devices. *See also supra* Section X.B. Belfiore thus satisfies Element 11.5.

2. Claim 12

[12] The method of claim 11, wherein a context tracking module associated with the context manager informs the plurality of mobile devices registered with the context manager of the context change event.

85. Belfiore discloses a context tracking module of the context manager (i.e., events component 155 of Belfiore's platform) that informs registered mobile devices (e.g., client devices authenticated with the platform and subscribed to event notifications) of a context change event. As explained for Element 11.5, Belfiore's context events notifications are provided by the Platform's events component 155. *Supra* Section X.C.1.e; *see also* EX1004, [0119-20], [0103] (triggering events based on inferring user context of performing a word processing activity). The events component 155 includes, among other things, event storage and logging mechanisms that facilitate tracking the context events (*id.*, [0126]) and also utilizes directories such as the MyState directory to track and distribute context events (*supra* Section X.C.1.e). Thus, Belfiore at least renders obvious claim 12.

3. Claim 15

[15] The method of claim 11, further comprising removing, by the context manager, one or more of the plurality of mobile devices from the plurality of mobile devices registered with the context manager.

86. Belfiore discloses removing, by the context manager (the platform) one or more of the mobile devices (mobile client devices) from being registered (e.g., authenticated) with the platform. For example, Belfiore discloses adapted directories 905 and 910, which include information for identities of a user. EX1004, [0183-84]. These adapted directories control access of entities to the directories (and notifications of changes and updates to the directories). *Id.* As described above at Section X.C.1.b, access to the user preference information and state information in the adaptive directories is provided after a user logs onto a client device and authenticates to a security component. EX1004, [0109], [0175]. However, as further described in Belfiore, a user may log onto another user's device as a guest user, suggesting a transient relationship between the guest user and the device in which a user would subsequently log out. *Id.*, [0251] (“She [Alex] borrows Sandra's Tablet PC and logs in as a guest user [...] the storage, directory and security components described above allow Alex to authenticate herself on a new client device and have all of her personalized information available to her through the client device.”), *see also id.*, [0197] (“Many consumers also use multiple service providers with whom they have widely varying business relationships and implicit trust (subscription,

transient transactions, none).”). One of ordinary skill would thus have understood or found it obvious that the borrowed device would no longer be registered once the guest user logs off of the borrowed device. EX1019, 283 (“Ideally, when a host leaves an area, that, too, should be announced to allow deregistration...”). For example, one of ordinary skill would have been motivated to remove the registration of the borrowed device in order to protect the security and privacy of the guest user by ensuring that the owner/primary user of the device is not able to access the personal information of the guest user. Even if the user does not perform a specific log off operation, as described above in Section IV.A., it was well known for registrations to only be provided as a lease, which expires after a designated time or is only valid for a specified time interval (and thereafter removing the registration of the device). EX1011, 257 (“all devices must be registered with the DHCP server(s) in order to be granted a lease.”), 263 (describing lease times, e.g., “Our DHCP servers are configured to grant ‘real’ address leases for 24 hours and ‘registration’ leases for five minutes. Once the leases have been gathered from the DHCP servers into the database, entries for real addresses are maintained for two weeks while entries for registration addresses are purged after only three days.”); EX1019, 354 (“Another problem that had to be solved is what to do about impolite mobile hosts that leave without saying goodbye. The solution is to make registration valid only for a fixed time interval. If it is not refreshed periodically, it times out, so the foreign

host can clear its tables.”). Thus, one of ordinary skill would have an expectation of success in performing the removal of the device from registration since there were already known lease mechanisms that would be usable in systems such as Belfiore’s platform. *See supra* ¶ 31.

87. Belfiore further describes “terminating a relationship between an entity and the directory” using the event component 155. Belfiore, [0184]. It would have been at least obvious to implement Belfiore such that the termination of the relationship resulted in the entity no longer being authenticated with the platform, at least because terminating a relationship between the entity and the directory would remove access of the entity to the directory. Belfiore, [0184]. Belfiore describes the term “entity” as including machines and hardware and/or software components, thus encompassing mobile devices such as the client devices 110. Belfiore, [0200]. Thus, one of ordinary skill would recognize the termination of the relationship between the entity and the directory in Belfiore as removing one or more of the plurality of mobile devices from the plurality of mobile devices registered with the context manager. Accordingly, for this additional reason, one of ordinary skill would recognize Belfiore as rendering obvious removing one or more of the plurality of mobile devices from the plurality of mobile devices registered with the context manager.

**XI. GROUND 2: CLAIMS 11, 12, 15, 17, AND 18 ARE
OBVIOUS OVER KENNEWICK (EX1005) AND BELFIORE (EX1004)**

88. As explained below, claims 11, 12, 15, 17, and 18 are obvious over Kennewick and Belfiore.

A. Kennewick

89. Kennewick (EX1005) published March 4, 2004, from an application filed June 3, 2003. I understand that this is prior art under at least Sections 102(a)-(b). Kennewick and the '367 patent share a common named inventor, Robert A. Kennewick, and a common title "Systems and Methods for Responding to Natural Language Speech Utterance." *Compare* EX1001, 1 *with* EX1005, 1. Furthermore, the '367 patent includes many of Kennewick's disclosures nearly verbatim. *See* EX1006. As a result, overlapping disclosures and Figs. 1-6 of Kennewick and the '367 patent describe substantially the same subject matter.

90. Kennewick discloses a natural language processing system (*e.g.*, EX1005, [0010]) that users interact with "on one or more devices," such as "a mobile or portable device" (*e.g.*, *id.*, [0043]).

91. It further discloses synchronizing user and session context across a user's multiple devices: "history and profile information for the user may be synchronized between the multiple devices on a periodic basis or other basis." *Id.*, [0043].

92. Kennewick also discloses recognizing words and phrases using information in “dictionary and phrase tables” (*e.g.*, *id.*, [0147]) and use of a “context stack” in “scoring” contexts, where the “weight of each context for the scoring is based on the relevance of one context to another and the age of the contexts” (*e.g.*, *id.*, [0166]).

B. Combining The Teachings Of Kennewick And Belfiore

93. One of ordinary skill implementing the teachings of Kennewick would have looked to Belfiore for its teachings on (i) the synchronization of context and other user information across a user’s different devices and (ii) multi-modal user interfaces. In particular, Kennewick is directed toward responding to spoken language utterances. As I have explained *supra* ¶¶ 27-31 the internet in its first years empowered a large number of new applications, but between 1994 and 2004 some advances were required in order to support secure, multi-modal mobile apps. Because Belfiore is directed to this type of innovation, it would have been natural for one of ordinary skill during this period to consider Belfiore for some of the details in how to make use of Kennewick’s claimed natural language technology.

1. Device Synchronization

94. Persons of ordinary skill would have been motivated to include Belfiore’s teachings on device authentication and event subscriptions, to facilitate the synchronization described in Kennewick. For example, Kennewick teaches

synchronizing a user’s “history and profile information” between the user’s many devices. *E.g.*, EX1005, [0043] (“[H]istory and profile information for the user may be synchronized between the multiple devices.”). Given Kennewick’s lack of express detail, those of ordinary skill would have turned to other references teaching methods for synchronizing information on devices used in NLP systems, and naturally would have encountered Belfiore. One of ordinary skill would have been motivated to incorporate Belfiore’s teachings on events in order to synchronize “state, session and preference status of a user” across devices. *E.g.*, EX1004, [0019]; *see also id.*, [0020] (“Events within the platform are used to synchronize.”); *supra* Section X.C.1.e). This would achieve Kennewick’s goal of device synchronization and provide benefits taught by Belfiore, including that a user could “switch from one device to another mid-session with all state and preference information remaining consistent.” *E.g.*, EX1004, [0019]; *see also id.*, [0097-98]; *supra* Section X.A, Section X.B.

95. Moreover, one of ordinary skill would have been motivated to use Belfiore’s authentication to ensure that information was only synchronized between authorized devices of a given user and not synchronized indiscriminately or exposed to “unscrupulous individuals.” EX1004, [0197]; *see also id.*, [0198] (identifying “authentication” as a security tool), [0200] (explaining how the “authentication

module” controls access to “hardware and/or software modules and components” and prevents entities from “hiding” their identity).

96. Those of ordinary skill would have reasonably expected success in using Belfiore’s event component and related functionality to synchronize context information within a Kennewick-like system. Belfiore teaches that events are “used to synchronize” ([0020]) and that its “event system enables software components and architectures to have access to continually updated information about their context.” EX1004, [0119]. As both Kennewick and Belfiore teach distributed NLP systems, one of ordinary skill would have recognized that incorporating Belfiore’s event-based synchronization would have required only basic software modifications such that authenticated mobile devices both provided and received events indicating changes to a given user’s session status and preferences. *Id.*, [0119]; *see also id.*, [0097-98] (describing use of MyState directory to facilitate providing “preference and session information” to a user’s authenticated devices). Additionally, Belfiore explains that the different components of its platform may be implemented on “general purpose computing devices, including various hardware components, such as personal computers, servers, laptops, hand-held devices, cell phones or the like.” *Id.*, [0041].

2. Multi-Modal Interfaces

97. Persons of ordinary skill also would have been motivated to incorporate Belfiore's teachings on multi-modal user interfaces, to beneficially allow users to interact using not only "traditional" methods such as keyboard entries" but also "audio/voice input" and stylus/touch input." EX1004, [0019]. Belfiore expressly teaches that these multi-modal inputs "allow[] flexibility in receiving input from a user and providing output to the user" (*id.*, [0118]) and provide different types of interactions "based on the user's state and proximity (*id.*, [0095]). *See also id.*, [0091].

98. Those of ordinary skill would have had a reasonable expectation of success in providing multi-modal inputs such as text and speech, at least because Kennewick teaches interfacing to its NLP system using devices with multiple input/output modalities, such as "a PDA or other portable computer device" and "a wireless telephone." EX1005, [0012], [0043]. Persons of ordinary skill would understand that these devices allowed for at least both voice or speech input/output, as well as text-based input/output. And Belfiore's teachings on how to implement multi-modal natural language communication were compatible with Kennewick's processing of natural language speech inputs. *E.g.*, EX1004, [0019], [0096] (discussing *inter alia* "analysis of an input string," "matching via logical form matching," and "generation of an appropriate language-like sentence along with the results."); EX1005, [0185] (describing "high level process 300 for receiving natural

language speech-based queries and/or commands and generating a response according to an embodiment of the invention”).

99. Further, one of ordinary skill would have found it obvious to incorporate the speech user interface device of Kennewick to allow for multi-modal input as taught by Belfiore because, similarly to the mobile devices described in Kennewick as including speech interface devices for processing user speech input, the multi-modal client devices of Belfiore are described as having a user interface that allows for speech input to enable a user to interact with the client device to ask questions or provide commands using natural language. EX1004, [0091], [0104]; EX1005, [0003], [0043] (“If the invention is used with a mobile or portable device that has position location capability, the location data can be used by the invention as part of the context for user questions. A user may use the system on one or more devices.”). Incorporating the multi-modal functionality of Belfiore would involve the straightforward application of conventional techniques and technologies. For example, Belfiore discloses natural language speech input processing techniques such as analyzing an input string and performing logical form matching, which are described in more detail in Kennewick. EX1004, [0092], [0098], [0106]; EX1005, [0010], [0088], [0149-56].

3. The Combined Kennewick-Belfiore System

100. One obvious implementation of this Kennewick-Belfiore system (hereinafter, “Kennewick-Belfiore”) would incorporate:

- Kennewick’s teachings on a “speech-based information query, retrieval, presentation and command environment” (EX1005, [0009]), adapted to provide **multi-modal natural language processing** of not just speech but also text- or touch-based inputs, as taught by Belfiore (EX1004, [0091-92], FIG. 4).
- **Context-aware natural language speech processing** that “determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile 110, and dialog history” (EX1005, [0152]), e.g., through *inter alia*:
 - “Dialog history” that is “maintained in a context stack.” *Id.*, [0166].
 - Use of grammar expressions “for evaluation of a context.” *Id.*, [0108].
 - Use of “a scoring system” to determine a “most likely context or domain” *Id.*, [0153].
 - Use of updated “probabilities or possibilities” to score “the possible contexts.” *Id.*, [0152].
- Multiple user **mobile devices** that:

- Are **synchronized** with one another, as taught by both Kennewick and Belfiore, upon each being **authenticated** by the same user, as taught by Belfiore. *E.g.*, EX1005, [0043-44]; EX1004, [0019-20], [0097-98], [0168] (enumerating client devices), claims 21, cl. 22-23 (authenticating), & claims 34-35 (first and second mobile phones).
- Provide **multi-modal user interfaces**, as taught by Belfiore. *E.g.*, EX1004, [0091-92], FIG. 4.
- An adaptive directory, such as a **MyState directory**, as taught by Belfiore, that stores information about a given “person or entity” (EX1004, [0182]), including “**session status information**” that “is regularly updated” as “a user works on a client device” (*id.*, [0098]). *See also id.*, [0109].
- An **event component**, through which a user’s mobile devices **subscribe to event notifications** in order to “synchronize” information and context across a user’s authenticated devices, as taught by Belfiore. EX1004, [0020]; *see also id.*, [0019]. “The event system enables software components and architectures to have access to continually updated information about their context.” *Id.*, [0119]. The event component may accomplish this synchronization, for example, by “notifying other entities of changes and updates to the MyState

directory.” *Id.*, [0184]. The event component may be incorporated with or separate from an event manager as taught by Kennewick. *E.g.*, EX1005, [0084] (“The event manager 100 may mediate interactions between other components of the main unit 98.”).

101. As explained below, such an implementation of Kennewick-Belfiore, and obvious variants thereof, render all challenged claims obvious.

C. Ground 2 Claim Mappings

1. Claim 11

a) Element [11.1] (Preamble)

[11.1] A method for processing multi-modal natural language inputs, comprising:

102. If the preamble is limiting, Kennewick-Belfiore satisfies it. As explained above, it would have been obvious to implement Kennewick-Belfiore so as to process multi-modal natural language inputs, as expressly taught by Belfiore. *Supra* Section X.B. Kennewick discloses a “a complete speech-based natural language query and response environment.” *E.g.*, EX1005, [0009-10]. Belfiore teaches a “user interface (UI)” that “is multi-modal, intelligent and responsive.” EX1004, [0019]; *see also id.*, [0104-06]; *see also supra* Section X.C.1.a.

b) Element [11.2]

[11.2] registering a plurality of mobile devices with a context manager in response to a registration module associated with the context manager receiving a communication from the plurality of mobile devices;

103. Kennewick-Belfiore satisfies this element, as it utilizes Belfiore's teachings on device authentication and communications, which either satisfy this element or render it obvious. *Supra* Section XI.B; *see also supra* Section X.C.1.b. Furthermore, Kennewick teaches that its system may be deployed in multiple devices that are all connected to a network. EX1005, [0044].

c) Element [11.3]

[11.3] subscribing the plurality of mobile devices registered with the context manager to one or more context events;

104. Kennewick-Belfiore satisfies this element, as it utilizes Belfiore's teachings on subscriptions to event notifications relating to updates to an adaptive directory, which either satisfy this element or render it obvious. *Supra* Section XI.B; *see also supra* Section X.C.1.c.

d) Element [11.4]

[11.4] receiving, at the context manager, a context input from one or more of the plurality of mobile devices registered with the context manager, wherein the context input includes a context change event; and

105. Kennewick-Belfiore satisfies this element, as it utilizes Belfiore's teachings on receiving updates to an adaptive directory, which either satisfy this

element or would have rendered it obvious. *Supra* Section XI.B; *see also supra* Section X.C.1.d. When implemented within a Kennewick-like system, Belfiore's teachings result in receiving a context input from one or more of the plurality of mobile devices registered with the context manager, wherein the context input includes a context change event. *Id.*

e) **Element [11.5]**

[11.5] informing the plurality of mobile devices registered with the context manager of the context change event, wherein informing the plurality of mobile devices registered with the context manager of the context change event synchronizes a context across the plurality of mobile devices.

106. Kennewick-Belfiore satisfies this element, as it utilizes Belfiore's teachings on event notifications that provide mobile devices with continually updated information about their context and enable a user interface to adapt and/or change based on the user's state and context across multiple devices, which either satisfy this element or render it obvious. *Supra* Section XI.B; *see also supra* Section X.C.1.e. Furthermore, Kennewick discloses synchronizing history and profile information between multiple devices, where the history and profile information are updated automatically as a user uses the system. EX1005, [0029], [0043]. The combined Kennewick-Belfiore implementation informs the plurality of mobile devices registered with the context manager of the context change event, thereby synchronizing a context across the plurality of mobile devices.

2. Claim 12

12. The method of claim 11, wherein a context tracking module associated with the context manager informs the plurality of mobile devices registered with the context manager of the context change event.

107. Kennewick-Belfiore satisfies this element, as it utilizes Belfiore's teachings on event notifications with logging mechanisms, which either satisfy this element or render it obvious. *Supra* Section XI.B; *see also supra* Section X.C.2. Furthermore, Kennewick discloses maintaining a context stack, which one of ordinary skill would understand as tracking the contexts because the context stack includes contexts that are associated with ages, meaning the system is aware of respective timing of multiple contexts. EX1005, [0166]. The Kennewick-Belfiore implementation thus includes a context tracking module associated with the context manager informing the plurality of mobile devices registered with the context manager of the context change event.

3. Claim 15

[15] The method of claim 11, further comprising removing, by the context manager, one or more of the plurality of mobile devices from the plurality of mobile devices registered with the context manager.

108. Kennewick-Belfiore satisfies this element, as it utilizes Belfiore's teachings on device log-on, authentication, and event subscription and notification, which at least render obvious removing mobile devices that a user has logged out of. *Supra* Section XI.B; *see also supra* Section X.C.3. So implemented, the Kennewick-

Belfiore implementation's context manager (platform) removes one or more of the plurality of mobile devices from the plurality of registered mobile devices. *Id.*

4. Claim 17

[17.1] The method of claim 11, further comprising: receiving a natural language utterance at a multi-modal voice user interface associated with at least one of the plurality of mobile devices registered with the context manager;

109. Kennewick-Belfiore satisfies this element. As explained above, the combination utilizes Kennewick's teachings and disclosures on natural language speech processing, accessed via mobile devices such as mobile telephones that are registered with the platform for event-based synchronization, as taught in more detail by Belfiore. *Supra* Section XI.B. Kennewick discloses receiving "a natural language utterance" from a user at a system comprising "a speech unit interface device that receives spoken natural language queries, commands and/or other utterances from a user." EX1005, [0010]; *see also id.*, [0012] (the system comprises "a PDA or other portable computer device"). Thus, in the Kennewick-Belfiore system, natural language utterances would be received at the multi-modal voice user interfaces of the mobile devices associated with / authorized by a given user. *See also supra* Section XI.C.1.b.

[17.2] identifying, using a knowledge-enhanced speech recognition engine associated with the at least one mobile device, one or more contexts that completely or partially match one or more text combinations contained in text transcribed from the natural language utterance, wherein identifying the matching contexts includes comparing the text combinations against one or more

grammar expression entries in a context description grammar and one or more expected contexts stored in a context stack synchronized through the context manager;

110. Kennewick-Belfiore satisfies this element. As explained above, the combination utilizes Kennewick's teachings and disclosures on natural language speech processing. *Supra* Section XI.B. Kennewick's parser corresponds to the claimed "knowledge-enhanced speech recognition engine."

111. The written description of the '367 patent describes a "knowledge-enhanced speech recognition engine" in terms of the functionality it provides, and largely tracks the language of claim 17. EX1001, 14:6-19. The '367 patent states that the knowledge-enhanced speech recognition engine "may be used to determine the intent of the request and/or to correct false recognitions," that it "may access a set of expected contexts that are stored in a context stack to determine a most likely context" and that it "may use context specific matchers that are able to identify context such as time, location, numbers, dates, categories (e.g., music, movies, television, addresses, etc.) and other context." EX1001, 14:9-17. These descriptions are consistent with the term itself, which suggests software that uses additional/contextual information in connection with performing speech recognition/understanding—for example, to recognize the intent of a user's request (and thus determine that the speech was likely recognized correctly) or to determine that the recognized speech may be incorrect. As shown in the passages just cited, the

'367 patent uses permissive language (e.g., “may”) when describing the knowledge-enhanced speech recognition engine’s functionality and does not require it to be able to perform all, or even any, of the specific functions described. In any event, as explained below, Kennewick’s parser provides the same functionality.

112. Kennewick’s parser is used to determine the intent of a request in the transcribed natural language utterance. It does this by examining the words of the transcribed utterance, as well as by using user profile information and dialog history, to determine a “context.” EX1005, [0152] (“The parser 118 determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile 110, and dialog history.”) The context in Kennewick is described as “what is the subject the query and/or command directed to,” among “other parameters used in defining the query and/or command.” EX1005, [0185]. “For example,” Kennewick explains, “a question with the keywords ‘temperature’ [sic] implies a context value of weather for the question.” *Id.*, [0152]. In other words, in this example, the parser determines that the intent of a user’s question using the keyword “temperature” is to ask about the weather. By determining the user’s intent, the parser can identify the appropriate application and query to execute in order to try to provide the user with the requested information. EX1005, [0152] (“The context of a question or command determines the domain and thereby, the domain agent 156, if any, to be evoked.”), [0092] (“Based on keywords in the questions and commands

and the structures of the questions and commands, the parser invokes the required agent[s].”) And by being able to identify the context, the parser also provides an indication that the speech was recognized correctly. And these disclosures show that Kennewick’s contexts are the subject matter to which a particular user input is directed and are used to determine the meaning of the user input. *Supra* Section XI.A.

113. In further alignment with the knowledge-enhanced speech recognition engine described in the ‘367 patent, Kennewick’s parser is also used to correct incorrect recognitions. For example, Kennewick discloses that the parser “uses a scoring system to determine the mostly likely context or domain for a user’s question and/or command.” EX1005, [0153]. But “[i]f the confidence level of the score is not high enough to ensure a reliable response, the system 90 may ask the user to verify whether the question and/or command is correctly understood.” EX1005. Kennewick further explains: “If the confidence level of the domain or context score is not high enough to ensure a reliable response, the system can request that the user verify the question or command is correctly understood. In general, the question may be phrased to indicate the context of the question including all criteria or parameters. If the user confirms that the question is correct the system may proceed to produce a response. Otherwise, either the user can rephrase the original question, perhaps adding additional information to remove ambiguity, or the system may ask one or

more questions to attempt to resolve the ambiguity or other actions may taken.” *Id.*, [0031]. By causing the system to prompt the user to verify whether the input was correctly understood, the parser provides the ability to correct false recognitions.

114. One of ordinary skill would have understood and found it obvious that Kennewick’s parser would access a set of expected contexts that are stored in a context stack to determine a most likely context. Kennewick discloses the “set of expected contexts” in a few different ways. First, Kennewick states that, as part of determining the most likely context, multiple “possible contexts are scored and the top one or few are used for further processing.” EX1005, [0152]. One of ordinary skill would recognize that the possible contexts of Kennewick establish an expectation of being the context of the user’s input, since the “top one or few [that] are used for further processing” are selected from the set of possible contexts. *Id.* Therefore, because these “possible contexts” are recognized as possibly matching the user’s intent, and thus reflect an expectation that at least one or some will be contexts of the user’s input, this collection of “possible contexts” is the first way that Kennewick discloses the “set of expected contexts.” Kennewick further discloses that the possible contexts are stored in a context stack, describing an example of how “the contents of the context stack,” such as keywords and criteria corresponding to past contexts, may be used in determining the context of a user’s next request (e.g., “a following-on question”): “[A voice query language] provides a grammar to

clearly specify the keyword used to determine the context and a set of one or [sic] criteria or parameters....The voice query language may be sensitive to the contents of the context stack. Thus, a following-on question can be asked using an abbreviated grammar, since key words and criteria can be inherited from the stack. For example, the user can simply asked [sic] about another keyword if the criteria of the question remain constant.” EX1005, [0144]-[0145]; *see also id.*, [0166] (“Part of the dialog history is maintained in a context stack.”).) In the excerpt above, the context keyword(s) stored in the context stack and associated with an earlier question is an example of an “expected context.” One of ordinary skill would understand the context stack to be stored in a data structure, as Kennewick describes storing “[u]ser specific data, parameters, and session and history information that may determine the behavior of agents 106” in “one or more user profiles 110,” which are further described as being stored in a database. *Id.*, [0085], [0186] (“These queries may then be sent to one or more agent domains, such as a domain which may access a database containing the user's profile...”). One of ordinary skill would further recognize that the contexts in the context stack of Kennewick can be ordered because the contexts are described as having weights or “other scoring variables that can be associated through the context stack.” *Id.*, [0166]. The set of contexts associated with earlier question(s) is another way that Kennewick discloses and renders obvious the “set of expected contexts.”

115. Kennewick’s parser, like the ’367 patent’s “knowledge-enhanced speech recognition” engine, also uses “context specific matchers that are able to identify context such as time, location, numbers, dates, categories (e.g., music, movies, television, addresses, etc.) and other context.” EX1001,14:13-17. For example, Kennewick describes “specific parameters or values (criteria) used to determine context for questions and commands.” EX1005, [0093]. These criteria, like the “context specific matchers” of the ’367 patent, include time, location, dates and other information. *Id.*, [0154] (“For a weather context, examples of criteria include, location, date and time. Other criteria can include command criteria (i.e., yes/no, on/off, pause, stop), and spelling.”); *see also id.*, [0189] (describing parameters and criteria).

116. Therefore, because Kennewick’s parser includes the specific functionality ascribed to a “knowledge-enhanced speech recognition engine” in the ’367 patent’s written description—and more generally is software that uses additional/contextual information in connection with performing speech recognition/understanding—Kennewick’s parser discloses and renders obvious the claimed “knowledge-enhanced speech recognition engine.”

117. Kennewick discloses **“identifying ... one or more contexts that completely or partially match...text , transcribed from the natural language**

utterance,” describing matching tokens (e.g., words) in a recognized utterance to contexts—such as the text “temperature” to the context “weather”:

Once the words and phrases have been recognized by the speech recognition engine **120**, the tokens and user identification is passed to the parser **118**. The parser **118** examines the tokens for the questions or commands, context and criteria. The parser **118** determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history. The context of a question or command determines the domain and thereby, the domain agent **156**, if any, to be evoked. For example, a question with the keywords “temperature” implies a context value of weather for the question. The parser dynamically receives keyword and associated prior probability or fuzzy possibility updates from the system agent 150 or an already active domain agent 156. Based on these probabilities or possibilities the possible contexts are scored and the top one or few are used for further processing.

EX1005, [0152].

118. Note that, although in this example, the keyword “temperature” does not literally match the word “weather” for a weather context, nothing in claim 17 requires this sort of narrow, verbatim matching. Claim 17 merely recites “one or

more contexts that completely or partially match one or more text combinations”, (EX1001, 31:20-21) which would be satisfied by recognizing that a text combination is associated with a context, and therefore matches the context.

119. In fact, both Kennewick, as shown in the excerpt above, and the '367 patent state how “[t]he parser **118** determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history.” EX1005, [0152]; EX1001, 21:9-11. One of ordinary skill would have therefore understood that the text “temperature” “matches” the “weather” context in the above example. Even if claim 17 required a verbatim match, one of ordinary skill would have understood and found it obvious, based on Kennewick’s teachings, to identify a context that matches verbatim a text combination in the transcription. For example, it would have been obvious that a keyword “weather” would identify a “weather” context because the keyword completely overlaps with the context descriptor. It would not make sense, for example, for a user’s question, “what’s the weather?” to not match a weather context for processing the question. One of ordinary skill would have been motivated to include verbatim matching in the system of Kennewick for at least the reason that verbatim matching represents the strongest probability of keyword matching and thereby improves the confidence and reliability of the parser by enabling a default or baseline case of context determination. The skilled artisan would have had a reasonable expectation of

success in implementing verbatim matching in the system of Kennewick because it would require no additional hardware or software modification to detect verbatim matches using the prior or fuzzy possibility algorithms already taught by Kennewick.

120. With respect to “text combinations,” in addition to matching a single word, such as keyword “temperature” mentioned in the excerpt above, Kennewick also makes clear that “text combinations” are matched, in at least two distinct ways.

121. First, Kennewick discloses examples of not only single-word keywords, but also multi-word keywords, such as “get the value” and “my stock portfolio” in the user utterance, “please get the value of my stock portfolio.” EX1005, [0186] (“For instance, suppose a user is interested in retrieving the value of her stock portfolio. The user may utter ‘please get the value of my stock portfolio.’ The system **90** may review this request together with stored data such as the user’s profile and determine keywords such as ‘get the value’ and ‘my stock portfolio.’”); *see also id.*, [0187] (“The user may utter ‘please record my favorite TV program.’ The system **90** may review this request together with stored data such as the user’s profile and determine keywords such as ‘record’ and ‘my favorite TV program.’”), [0098] (“new keywords for a domain, which can include names of politicians, athletes, entertainers, names of new movies or songs, etc. who have achieved recent prominence.”). One of ordinary skill would have understood and found it obvious to identify context(s) by matching multi-word keywords (i.e., “text combinations”) in

a recognized utterance—for example, to identify a stock portfolio context (and ultimately invoke a stock portfolio agent) by matching the text combinations “get the value” and/or “my stock portfolio.” Kennewick, e.g., [0186] (“a domain [agent] which accesses stock pricing sources to determine the answers to these questions”).

122. Second, Kennewick makes clear that the evaluation of a context can involve matching not only keyword(s) in the transcription, but criteria or parameters in the transcription as well. Kennewick, [0031]. For example, Kennewick discloses that “[i]n general, the question may be phrased to indicate the context of the question including all criteria or parameters.” *Id.* The criteria or parameters can be “used to determine the context for questions and commands.” Kennewick, [0093] (“The system agent **150** manages the criteria handlers **152**, which handle specific parameters or values (criteria) used to determine context for questions and commands.”).

123. Kennewick further discloses identifying a context by comparing the text combinations against grammar expression entries (i.e., grammar expressions) in a context description grammar (i.e., regular grammar). Kennewick, [0108], [0152]. For example, Kennewick explains that the matching process described above uses a grammar. Kennewick, [0108] (explaining that an agent “passes a regular grammar expression to the parser **118** for evaluation of a context or question.”). Kennewick further describes how, in one example, a grammar is used “to clearly specify the

keyword used to determine the context and a set of one or [sic] criteria or parameters.” Kennewick, [0144] (“[T]he voice query language helps users clearly specify the keywords or contexts of the question or command along with the parameters or criteria. The language provides a grammar to clearly specify the keyword used to determine the context and a set of one or criteria or parameters.”).

124. Kennewick also discloses that the grammar would include multiple grammar expressions. Kennewick, [0108]. For example, Kennewick explains how a “content package” for an agent “include[s] questions or commands” and that each question or command has an associated regular grammar expression:

Content packages include questions or commands. Each command or question or group of commands or questions includes contexts used for creation of one or more queries. The agent **156** passes a regular grammar expression to the parser **118** for evaluation of a context or question. An initial or default context is typically supplied for each command or question. The command or question includes a grammar for the management and evaluation of the context stack.

Id.

125. Therefore, a collection of grammar expressions used to identify a context in Kennewick discloses “grammar expression entries in a context description grammar” as claimed. The collection of Kennewick discloses “a context description

grammar” because it comprises a grammar that describes context(s). *Id.* This is consistent with the descriptions of a “context description grammar” in the written description of the ’367 patent. EX1001, 13:58-60 (“The text combination may be compared against entries in a context description grammar that is associated with each agent 106.”), 16:55-63 (“Content packages include questions or commands. Each command or question or group of commands or questions includes contexts used for creation of one or more requests. The domain agent 156 passes a context description grammar expression to the parser 118 for evaluation of a context or question. An initial or default context is typically supplied for each command or question. The command or question may include a grammar for the management and evaluation of the context stack.”). One of ordinary skill would have understood the collection of grammar expressions in Kennewick to be stored as entries in a data structure, as Kennewick describes the collection as being a category of data used by agents, the parser, etc., where such data is further described as being stored in a database. EX1005, [0107-8], [0085], [0186]. One of ordinary skill would further recognize that the grammar expressions in Kennewick constitute or reference sets of rules describing the structure of natural language in a particular context, as Kennewick describes using grammar expressions “for evaluation of a context or question,” where the grammar expressions describe structures (e.g., syntax, tokens) of natural language for a given context. *Id.*, [0108], [0155-56].

126. Thus, the “regular grammar expression” in Kennewick serves the same purpose as the “context description grammar expression” in the ’367 patent: it is passed to the parser for evaluation of a context or question. Kennewick, [0108]. With respect to evaluating a context with the context description grammar, the written description of the ’367 patent largely tracks the language of claim 17, stating that “[t]he text combination [of a text message] may be compared against entries in a context description grammar that is associated with each agent **106**. If a match is identified between an active grammar in the context description grammar and the command and/or request, then the match may be scored.” EX1001, 13:58-62. Although Kennewick does not also include a nearly-verbatim passage to this one, Kennewick does make clear, and it would have been obvious to a person of ordinary skill in the art, that using a “regular grammar” for evaluation of a context would involve comparing grammar expression entries in the regular grammar to text combinations in a transcribed utterance. This is because, as explained above, Kennewick describes matching text in a transcription, such as keywords and criteria, to grammar expressions to identify context(s) for scoring. Kennewick, e.g., [0144] (“[T]he voice query language helps users clearly specify the keywords or contexts of the question or command along with the parameters or criteria. The language provides a grammar to clearly specify the keyword used to determine the context and a set of one or criteria or parameters.”), [0152] (“The parser **118** determines a

context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history.”).

127. Thus, based on Kennewick’s descriptions a person of ordinary skill in the art would have found it obvious, for example, that matching “temperature” to identify a “weather” context or matching “stock portfolio” to identify a “stock portfolio” context would involve matching grammar expressions in a grammar.

128. One of ordinary skill would have understood and found it obvious that Kennewick’s parser would access one or more expected contexts that are stored in a context stack to determine a most likely context. Kennewick discloses the “one or more expected contexts” in a few different ways. First, Kennewick states that, as part of determining the most likely context, multiple “possible contexts are scored and the top one or few are used for further processing.” Kennewick, [0152]. Therefore, because these “possible contexts” are recognized as possibly matching the user’s intent, the “possible contexts” are a first way that Kennewick discloses the “one or more expected contexts.” Kennewick further discloses that possible contexts matching a transcription are stored in a context stack, describing an example of how “the contents of the context stack,” such as keywords and criteria corresponding to past contexts, may be used in determining the context of a user’s next request (e.g., “a following-on question”):

[A voice query language] provides a grammar to clearly specify the keyword used to determine the context and a set of one or criteria or parameters....The voice query language may be sensitive to the contents of the context stack. Thus, a following-on question can be asked using an abbreviated grammar, since key words and criteria can be inherited from the stack. For example, the user can simply asked [sic] about another keyword if the criteria of the question remain constant.

129. Kennewick, [0144]-[0145]; *see also id.*, [0166] (“Part of the dialog history is maintained in a context stack.”). In the excerpt above, the context keyword(s) stored in the context stack and associated with an earlier question is an example of an “expected context.” The one or more contexts associated with earlier question(s) is another way that Kennewick discloses and renders obvious the “one or more expected contexts.” As Kennewick makes clear, the transcribed text of a user’s utterance (e.g., text of “a following-on question”) (“text combinations”) can be compared to context keywords in the context stack (“against one or more expected contexts stored in a context stack”).

130. One of ordinary skill would have found it obvious to synchronize the context stack through the context manager (i.e., platform) of Belfiore, because the platform of Belfiore is described as being used to adapt and synchronize a user interface across client devices based on context. Belfiore, [0098]. Accordingly, one

of ordinary skill would have been motivated to look to Belfiore to provide more detailed teachings relating to the configuration and communications involved in maintaining the synchronization of the context across mobile devices.

[17.3] scoring each of the identified matching contexts; and

131. Kennewick-Belfiore satisfies this element. As explained above, the combination utilizes Kennewick's teachings and disclosures on natural language speech processing, including its teachings on context scoring. *Supra* Section XI.B.

132. Kennewick discloses "scoring each of the identified matching contexts." As discussed above, Kennewick describes identifying possible contexts that match a user's utterance "by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history." Kennewick, [0152]. "Based on these probabilities or possibilities the possible contexts are scored and the top one or few are used for further processing." *Id.*; *see also id.*, [0153] ("The parser **118** uses a scoring system to determine the mostly likely context or domain for a user's question and/or command."), [0030] ("The system may determine the mostly likely context or domain for a user's question or command, for example, by using a real-time scoring system or other technique.").

[17.4] selecting the matching context having a highest score as the most likely context for the natural language utterance.

133. Kennewick-Belfiore satisfies this element. As explained above, the combination utilizes Kennewick’s teachings and disclosures on natural language speech processing, including on context scoring and selection. *Supra* Section XI.B.

134. Kennewick discloses selecting (i.e., using for further processing) the matching context having a highest score (i.e., “top one or few”) as the most likely context for the natural language utterance. Kennewick, [0152]-[0153]. For example, as discussed above, Kennewick discloses that “the possible contexts are scored and the top one or few are used for further processing.” Kennewick, [0152]. Kennewick further describes how “[t]he parser **118** uses a scoring system to determine the mostly likely context or domain for a user’s question and/or command” and that “[b]ased on this scoring, the system **90** invokes the correct agent.” Kennewick, [0153]. One of ordinary skill would have therefore understood and found it obvious that the matching context “having a highest score” would determine the “most likely context”—first, because Kennewick discloses selecting only the context with the “top” score as one example, and second, because it would be quite sensible to select the context having a highest score as a mostly likely context even where “the top...few are used for further processing.” Kennewick, [0152].

5. Claim 18

[18] The method of claim 17, wherein the context change event received at the context manager includes a communication from the at least one mobile device that indicates a change in context to the most likely context.

135. Kennewick-Belfiore satisfies this element. As explained above, the combination utilizes Belfiore's teachings and disclosures on use of event subscriptions to synchronize context across a user's devices. *Supra* Section XI.B.

136. As explained above, Belfiore discloses receiving a context input from a mobile device, wherein the context input includes a context change event including changes to the "session status" and context of a given user on given device. *Supra* Sections XI.C.1.d and X.C.1.d; *see also* Belfiore, [0020], [0068], [0097], [0109], [0119-20], [0123], [0182], [0184], [0203]. For example, Belfiore teaches that its event component "enables software components and architectures to have access to continually updated information about their context." Belfiore, [0119]. Kennewick's "most likely context" (e.g., "top one or few" scored contexts) relates to the user's session status and context at least because it reflects the course and state of the user's NLP speech session. Kennewick, [0151-52]. Thus, within the Kennewick-Belfiore system, the session status / context change of identifying a "most likely context" would be communicated by the event component to the platform (e.g., for storing in the user's MyState directory, as taught by Belfiore) and then also to the user's other authenticated devices, including mobile devices. *Supra* X.C.1.c-e; *see also* Belfiore, [0097-98], [0123], [0184].

**XII. GROUND 3: CLAIMS 17 AND 18 ARE OBVIOUS
OVER KENNEWICK, BELFIORE, AND ROSS (EX1007)**

137. To the extent Patent Owner argues that Kennewick somehow does not teach verbatim matching and thus does not satisfy element 17.2's "identifying...one or more contexts that completely or partially match one or more text combinations contained in text transcribed from the natural language utterance," Ross provides more express teachings that confirm the obviousness of claims 17-18.

A. Ross

138. Ross (EX1007) is a published patent application that published September 19, 2002, so I understand that it qualifies as prior art at least under 35 U.S.C. § 102(b). Ross is analogous art to the '367 patent at least because it describes techniques for using a grammar to identify a context for processing a user command or question and techniques for choosing an application to invoke to process a command or question in a user's transcribed utterance. EX1001, 21:21-26; Ross, e.g., [0010], [0013].

139. Ross describes a system for determining the appropriate speech-enabled application to handle a user's spoken utterance in a multi-context environment. Ross, [0010] ("determining a speech-enabled application to receive a spoken utterance in a multi-context speech enabled environment."). The system uses a "context manager" that maintains a list of application contexts, each associated with a grammar. *Id.*, [0033] ("The context manager 50 includes a context list 62. The

context list includes contexts 70 (e.g., 70-1, 70-1, 70-3, etc.) for speech enabled applications 26, which represent the grammars for the applications 26.”). When a user speaks, the system receives a recognition message and compares the utterance against the grammars in the context list to determine which application can process the utterance. *Id.*, [0037] (“the context manager 50 receives the recognition message 68 and selects a context 70 for a speech enabled application 26 by checking the speech representation against contexts 70, starting at top of context list 62 to determine the highest priority context 70 that can process the speech representation.”). Ross provides examples where words or phrases in utterances are matched verbatim against words or phrases in grammars of different contexts. *E.g.*, *Id.*, [0040-53] (describing matching an utterance of “print it” to an electronic mail application or calendar application grammar that allows phrases including “print it”).

B. The Combined Kennewick-Belfiore-Ross Implementation

140. One of ordinary skill implementing Kennewick-Belfiore as described above would have been motivated with a reasonable expectation of success to include Ross’s teachings on comparing transcribed user utterances to grammar expressions in a grammar to identify a context. Kennewick generally teaches and discloses comparing text combinations against grammar expression entries in a context description grammar. *Supra* Section XI.A, XI.C.1. Ross, in turn, describes a remarkably similar technique but in much more detail. A skilled artisan would have

thus naturally turned to these more detailed teachings in Ross—and been motivated to apply those detailed teachings—when implementing Kennewick’s teachings and disclosures. Those of ordinary skill would have had a reasonable expectation of success in doing so given that Ross merely provides more details on a “comparing” technique highly similar to that taught and disclosed in Kennewick. Skilled artisans would have thus understood these Ross teachings as highly compatible with Kennewick and seen no issues in applying those teachings when implementing Kennewick. This would have involved the straightforward application of conventional techniques and technologies. Ross, [0060] (“In a preferred embodiment, the grammar used for each speech-enabled application 26 may be a Backus Naur Form (BNF) grammar as shown above.”).

141. The combination would have predictably resulted in Kennewick’s parser identifying context(s) matching text combination(s) contained in a transcribed user utterance by “comparing the text combinations against the [sic] grammar expression entries in the [sic; a] context description grammar” as claimed.

C. Ground 3 Claim Mappings

142. As explained above in Ground 2, Kennewick-Belfiore renders obvious claims 17-18. Sections XI.B.3, XI.B.7-8. To the extent Patent Owner argues that element 17.2 requires a verbatim match and that Kennewick somehow does not render obvious a method that involves such matching, Ross confirms the

obviousness of the claims. Therefore, claims 17 and 18 would have been obvious in view of Kennewick-Belfiore-Ross for at least the reasons outlined herein.

143. Ross describes techniques for determining a context associated with a user's spoken command or question in order to determine an application to invoke to process the command or question, just like Kennewick and the '367 patent. Ross, [0010], [0013]. For example, Ross describes a technique where determining a context involves evaluating grammars associated with potential contexts against a user's recognized speech input—Figure 4 of Ross illustrates a general overview:

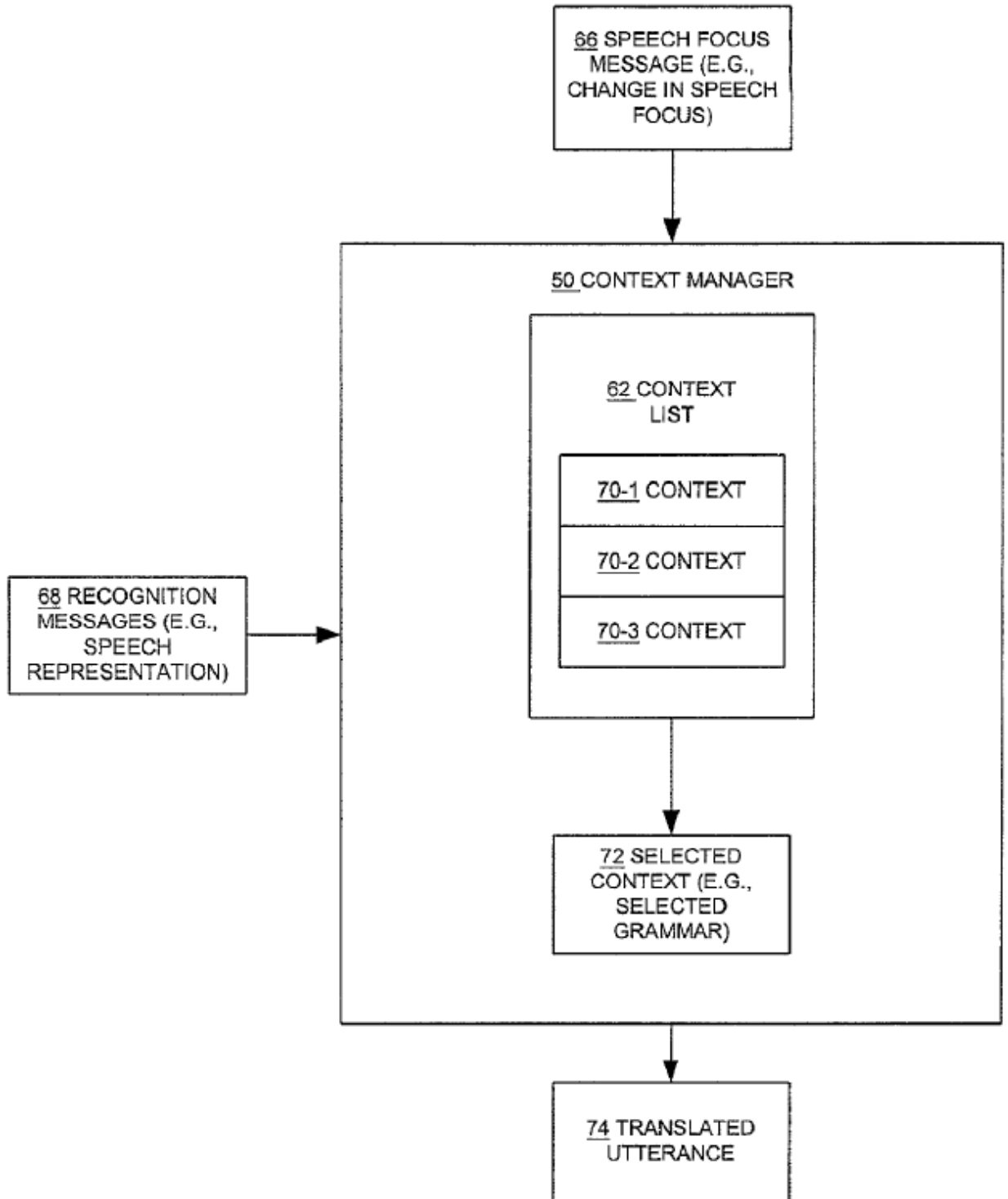


FIG. 4

Ross, Fig. 4. As shown above, Ross includes a context manager 50 that includes a context list 62. Ross, [0033]. Context list 62, in turn, “includes contexts 70 (e.g., 70-1, 70-2, 70-3, etc.) for speech-enabled applications 26, which represent grammars for the applications 26.” Ross, [0033]. A user’s transcribed utterance (a “recognition message”) is tested against the grammars in the context list in order to determine an application to invoke:

The context manager 50 maintains the priority and state of the various grammars in the context list 62 in the system.... Recognition messages 68 from the speech engine interface 30 are tested by the context manager 50 against the active grammars in the context list 62 in priority order. When a successful match is found, the corresponding translation 74 is dispatched to the script engine 38 for execution, and the priority of the matching grammar (i.e., the selected context 72) is raised.

Ross, [0034].

144. In the following descriptions, Ross further makes clear that “test[ing]...against the active grammars” and finding a successful “match” involves matching text combinations in a user’s transcribed utterance to grammar expressions in a grammar. Ross explains that “a grammar is defined for each application 26” and describes two example grammars, one for an electronic mail application:

```

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

```

(Ross, [0035], [0040]) and one for a calendar application:

```

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

```

Ross, [0046]. A grammar expression can generally comprise word tokens, rule references, and grammar operators. For example, in the grammar for the electronic mail application shown above, the line “<message> = the? <nth> message | it | this,” defines a rule named “message.” The rule specifies a grammar expression that includes word tokens (e.g., “the,” “message”), a reference to another rule (“<nth>”) and a grammar operator (“?”). *Id.* As shown above, each grammar includes a number of grammar expressions.

145. Therefore, as Ross explains, the grammar for the electronic mail application allows spoken phrases such as “open the first message,” “create a message,” “send this,” and “print it” to be matched, while the grammar for the calendar application matches phrases such as “open the first appointment,” “create an appointment,” “print the fourth appointment,” and “print it.” Ross, [0041]-[0051]. The phrases that can be matched can be easily recognized by substituting the term in angle brackets with text (or expression) to which it can correspond. For example, in the grammar for the electronic mail application, “<message>” can correspond to “the? <nth> message,” “it,” or “this.” Therefore, the grammar expression “print <message>” can match “print it” (as well as “print this” or “print the first message” and so on). Ross discloses, therefore, that a text combination in a user’s transcribed utterance, such as “print it,” matches the “print <message>” expression defined by the “mail” rule as well as the “it” expression defined by the “message” rule in the grammar for the electronic mail application. With respect to the calendar application grammar, “print it” would also match the “print <appointment>” and “it” expressions in the “appointment” rules. *Id.*

146. Using the example electronic mail and calendar application grammars shown above, Ross further describes how identifying context(s) involves matching transcribed text to the grammar expressions in the grammars:

If both of these grammars were loaded into the context manager 50, (ignoring foreground/background issues in this example, assuming that the grammars are the same for both foreground and background), the speech center system 20 is listening for any of the phrases accepted by either grammar. If the speech center 20 hears a phrase such as “print the first message” or “print the first appointment,” the context manager 50 can readily figure out the intended target application 26 for the uttered sentence. Only one grammar will accept the phrase, which thus indicates the selected context 72 for that phrase and that associated application 26 is the one that should be targeted to receive the corresponding command.

If the sentence is “print it” however, both grammars are capable of accepting the utterance. The context manager 50 therefore has to make a choice by referring to the context list 62 of applications in order of recency of access. The context manager 50 tests the utterance against these grammars (indicated by the contexts 70 in the context list 62) in priority order, and passes the commands on to the first application 26 that has a grammar that will accept the phrase.

Ross, [0052-53].

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

Dated: 7/18, 2025



Paul Jacobs

APPENDIX A

Paul S. Jacobs

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

4105 Faith Ct.
Alexandria VA 22311

Education

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

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

Work History

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

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

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

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

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

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

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

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

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

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

APPENDIX A

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

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

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

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

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

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

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

Professional Activities

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

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

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

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

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

Member of Intellectual Property Committee of the USACM

Executive Committee, Association for Computational Linguistics, 1997-9

Other Activities

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

APPENDIX A

PAUL S. JACOBS

PUBLICATIONS, PROFESSIONAL AND SCHOLARLY ACTIVITIES

Refereed Journal Articles

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

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

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

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

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

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

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

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

Refereed Conference Proceedings

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

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

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

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

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

APPENDIX A

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

APPENDIX A

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

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

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

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

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

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

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

Books and Book Chapters

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

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

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

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

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

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

APPENDIX A

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

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

Other Published Articles

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

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

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

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

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

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

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

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

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

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

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

APPENDIX A

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

Technical Reports Not Published Elsewhere

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

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

Research Grants

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

Professional Societies and Committees

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

Program committees:

- AAAI National Conference on Artificial Intelligence

- ACL Annual Conference

- ACL Conference on Applied Natural Language Processing

- ACM SIGIR Conference on Research and Development in Information Retrieval

- ARPA Speech and Natural Language Conference

- Others

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

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

Executive Committee of the ACL, 1995-1998

USACM Public Policy Committee, Intellectual Property Subcommittee, 2005 -

U.S. Patents

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

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