

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION.,
Petitioner,
v.

X1 Discovery, Inc.
Patent Owner

Case IPR2025-00254
U.S. Patent No. 10,552,490

Case No. IPR2025-00254

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

	Page
I. The '490 Patent.....	1
II. Technology Background.....	4
III. Claim Construction.....	5
IV. GROUND 1: Petitioner Has Failed to Show the Challenged Claims Are Obvious Over the Combination of Wilcox (EX1059), Londergan (EX1060), Raskin (EX1021), and Wu (EX1019).	6
A. A POSITA Would Not Have Been Motivated To Combine Wilcox And Londergan Because Lotus Notes 3.0 And Lotus Notes R5 Have Distinct Features And Are Not The Same System.	8
1. Query Builder (v. 3.0) vs. Search Builder (R5).....	11
2. Search Bar	13
3. Petitioner Improperly Combined Wilcox and Londergan.	13
4. Petitioner's Proposed Wilcox-Londergan Combination Represents Improper Hindsight Bias.	16
B. A POSITA Would Not Have Been Motivated To Combine Wilcox-Londergan With Raskin.	18
C. A POSITA Would Not Have Been Motivated To Combine Lotus Notes With Wu.....	25
D. The Prior Art Does Not Disclose All Limitations of the '490 Patent's Claims.....	29
V. GROUND 2: Microsoft Has Not Shown That Any Challenged Claim Is Obvious in View of the Combination of Schwartz (EX1004), Sebastian (EX1005), Barber (EX1006), True (EX1009), and Baeza- Yates (EX1023).	36
A. Petitioner Mischaracterizes Search in Entourage as described in Schwartz, Sebastian, and Barber.....	37
1. The Entourage References Describe an Email System.....	37
2. Entourage's List Filtering Function.....	38

3.	Entourage’s Find and Advanced Find Commands.	39
4.	Petitioner’s Mischaracterization of Entourage’s Search Functionality.	42
B.	A POSITA Would Not Have Been Motivated to Combine the Entourage References with True.	45
1.	True Does Not Teach Incrementally Searching a Set of Files for Multiple Search Strings.	46
2.	A POSITA Would Not Have Looked To True’s “Interest Level” Method To Modify the Entourage References.	50
3.	Petitioner Fails To Explain How A POSITA Would Modify Entourage To Include True’s Disclosure.	53
C.	A POSITA Would Not Have Looked to Baeza-Yates to Improve Entourage.	56
D.	Entourage In View Of True And Baeza-Yates Does Not Disclose All Limitations of the ’490 Patent’s Claims.	59
1.	The Combination Of Entourage, True, And Baeza-Yates Does Not Teach updating a set of matching files of the plurality of files as each successive character of the first partial search string and the second partial search string is received respectively in the first search field (Limitation [1.13]).	59
2.	The Combination of Entourage, True, and Baeza-Yates Does Not Disclose Searching On Multiple Partial Strings.	61
3.	The Combination Of Entourage, True, And Baeza-Yates Does Two Partial Search Strings Separated By A String Separator Character In The Same Search Field (Limitations [1.8]-[1.10])	64
4.	The Combination Of Entourage, True, And Baeza-Yates Does Not Teach Highlighting A Plurality Of Occurrences Of The First And Second Partial Search Strings In The First File (Claim 11).	66
VI.	CONCLUSION.	67

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Adidas AG v. Nike, Inc.</i> , 963 F.3d 1355 (Fed. Cir. 2020)	21
<i>Hewlett Packard Enterprise Co. v. K.Mizra LLC</i> , No. IPR2022-00843, 2022 WL 16556406 (P.T.A.B. Oct. 31, 2022).....	22
<i>In re Klein</i> , 647 F.3d 1343 (Fed. Cir. 2011)	48
<i>In Re Magnum Oil Tools Int’l, Ltd.</i> , 829 F.3d at 1380	27, 44, 53, 56
<i>Natera, Inc. v. NeoGenomics Laboratories, Inc.</i> , 106 F.4th 1369 (Fed. Cir. 2024)	19
<i>RPX Corp. v. Rothschild Connected Devices Innovations, LLC</i> , IPR2016-00443, Paper 28 (FWD) (PTAB Jul. 6, 2017)	7
<i>RPX Corp. v. Rothschild Connected Devices Innovations, LLC</i> , IPR2016-00443, Paper 8.....	7, 8
<i>Securus Techs., Inc. v. Global Tel*Link Corp.</i> , IPR2015-00155, Paper 30 (PTAB Apr. 7, 2016)	18
<i>The Walt Disney Co., et al. v. WAG Acquisition, L.L.C.</i> , IPR2022-01346, Paper 12 (PTAB Mar. 10, 2023)	16
<i>U.S. Well Servs., LLC v. Halliburton Energy Servs., Inc.</i> , No. IPR2022-01163, 2022 WL 17822529 (P.T.A.B. Dec. 20, 2022).....	18
<i>Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.</i> , 200 F.3d 795 (Fed. Cir. 1999)	2
Other Authorities	
37 C.F.R. § 42.107	1

PATENT OWNER'S EXHIBIT LIST

Exhibit No.	Description of Document
2001	Declaration of Mr. Martin Haeberli, dated April 28, 2025
2002	CV of Mr. Martin Haeberli
2003	Revised Stipulation to Stay, <i>XI Discovery, Inc. v. Microsoft Corp.</i> , No. 8:23-cv-02415 (C.D. Cal. Feb 29, 2024), ECF No. 52
2004	Order Granting Revised Stipulation to Stay, <i>XI Discovery, Inc. v. Microsoft Corp.</i> , No. 8:23-cv-02415 (C.D. Cal. Mar. 1, 2024), ECF No. 53
2005	Joint Status Report, <i>XI Discovery, Inc. v. Microsoft Corp.</i> , No. 8:23-cv-02415 (C.D. Cal. Mar. 12, 2025), ECF No. 57
2006	Minutes of Status Conference, <i>XI Discovery, Inc. v. Microsoft Corp.</i> , No. 8:23-cv-02415 (C.D. Cal. Mar. 12, 2025), ECF No. 59
2007	Excerpt of the Initial Determination Granting-In-Part Respondents' Motion for Summary Determination, <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389, at 14 (Sept. 3, 2024)
2008	Excerpt of the Initial Determination of Violation of Section 337 and Recommended Determination on Remedy and Bond , <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389, at 31-32 (Feb. 26, 2025)
2009	Email from counsel for Microsoft to counsel for X1 regarding <i>Sotera</i> stipulation, Apr. 18, 2025
2010	File History of U.S. Patent No. 7,370,035

Pursuant to 35 U.S.C. § 313 and 37 C.F.R. § 42.107, Patent Owner X1 Discovery, Inc. (“X1” or “PO”) submits this Patent Owner Preliminary Response to Microsoft Corporation’s (“Microsoft” or “Petitioner”) Petition challenging claims 1-20 of U.S. Patent No. 10,552,490 (“the ’490 patent”). X1 requests that the Board deny institution of Microsoft’s Petition (“Petition” or “Pet.”) because Microsoft has not met its burden of establishing a reasonable likelihood of prevailing with respect to at least one challenged claim.

Microsoft’s Petition suffers two fatal flaws. First, Microsoft fails to establish that a person of ordinary skill in the art (“POSITA”) would have been motivated to either modify the identified lead references or combine those references with the secondary references in the manner alleged. Second, Microsoft fails to explain how the cited prior art discloses all elements of the challenged claims. Compounding these issues, the Petition relies heavily on the declaration of Microsoft’s expert, Dr. Loren Terveen, which itself relies on vague statements about what a POSITA would have known or found obvious. As a result, Microsoft’s Petition fails to establish that there is a reasonable likelihood that Microsoft would prevail with respect to any challenged claim, and the Board should deny institution.

I. The ’490 Patent

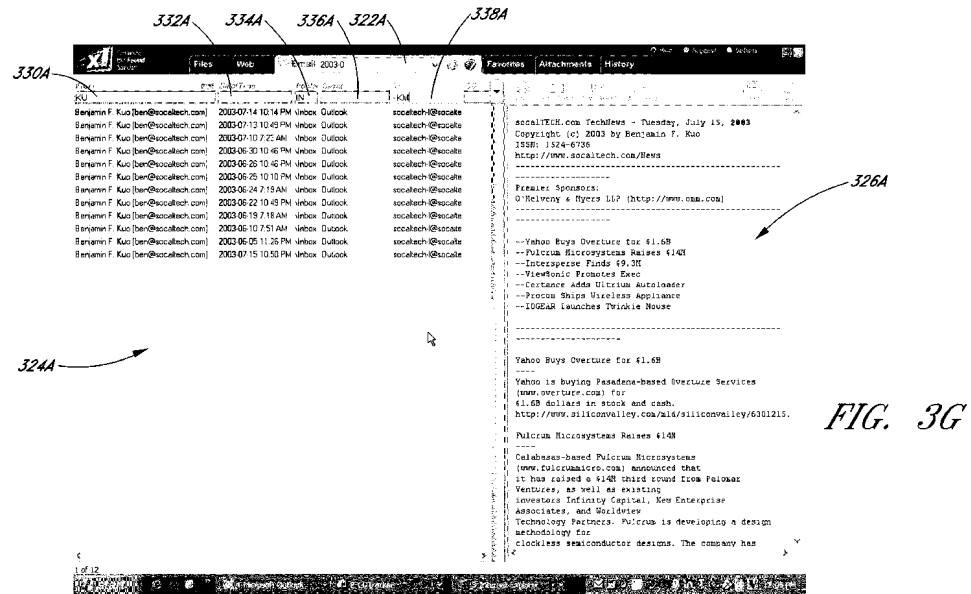
The ’490 Patent, which is titled “Methods and systems for search indexing,” issued from U.S. Application Ser. No. 15/495,507 filed on April 24, 2017.

The '490 Patent generally relates to systems for enhanced data searching and indexing, and providing faster and more efficient search results and more particularly to systems for storing an index data structure comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included. (EX1001, 1:35–36.) As the '490 Patent explains, conventional searching methods of the time “require the user to type in a search term, click on a search button, and review the results.” (EX1001, 1:43-2:5.) If the search did not return the desired results, “the user needs to edit the search terms, click on an initiate search button, and again review the results”—making the process “slow and cumbersome.” (EX1001, 1:43–2:5.)

The '490 Patent resolved these inefficiencies by allowing users to incrementally search multiple strings using just a single search field, and use an index data structure comprising associations between files and character strings to further improve searching speed and file accessibility. (EX1001, 3:43–54, 5:29-36.) As the '490 Patent explains, the results for incremental searches are “narrowed substantially immediately after each character in a search string is entered by a user.” (EX1001, 2:24–27.) Accordingly, unlike a conventional delimited search, which requires the user to execute a search by, for example, pressing a search button or hitting enter before displaying the results, the results of an incremental search display as the user types out the string. (EX1001, 1:43–50, 2:24–27.) By searching

incrementally, the “user is provided with substantially immediate feedback as the search string is being entered, and so [they] can quickly decide on the desirability of entering additional search characters, or of entering a new search string, or deleting one or more search string characters.” (EX1001, 2:27–33.)

Further improving upon the efficiency of incremental searching, the primary embodiment of the ’490 Patent allows the user to incrementally search multiple strings using a single search field. (EX1001, 2:41–47.) As such, the user can enter two strings into a single “search field; and incrementally locat[e] at least a first document that has a least a first word that begins with the first string and a second word that begins with the second string.” (EX1001, 2:41–47.) Thus, the search results immediately narrow for each portion of the string the user enters into the searching field. (EX1001, 2:41–47, 16:32–51.) Figure 3G of the ’490 Patent, showing an embodiment of the disclosed searching method, is included below:



(EX1001, Fig. 3G.).

II. Technology Background

Before the '490 Patent's invention, the general availability of information and documents had been steadily rising and the amount was expected to grow exponentially. (See EX1011, 1:23-26; EX1013, 1:14-20.) There was an ever-growing need for efficient methods of searching and accessing this unprecedented amount of stored information. (EX1023, 2-3.) Conventional search applications at the time required users to type in a search term, click on a search button, and review the results, leading to slow and cumbersome searches. (EX1001, 1:38-41.) In instances where the user was not satisfied with the results, the user would need to edit the search terms and go through the process again. (EX1001, 1:41-45.) Users needed a unified interface and application that allowed effective searches on various types of data and on multiple criteria. (EX2001,38.)

In the years leading to the invention of the '490 Patent, a variety of methods and systems were proposed to fill this need. In describing the alleged “State of the Art,” Petitioner and its expert describe such attempts, including attempts to implement incremental search. But the disclosures Petitioner points to all fall short of the '490 Patent’s method of using a unified search that allowed for incremental searching over multiple search terms. (EX2001, ¶¶38-47.)

III. Claim Construction

The Board need not expressly construe any term for purposes of this proceeding. *See Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999). In United States International Trade Commission (“ITC”) investigation 337-TA-1389 involving related patents, the Administrative Law Judge (“ALJ”) construed the term “incremental search” to mean “a search performed at each increment of the entry of the search string.” (EX2007, 32-33.) The ALJ also construed the term “non-adjacent in at least one of the identified documents” to mean “non-adjacent within the content of at least one of the identified documents.” (EX2008, 14.) The Board need not construe this term because the Petitioner fails to make its case regardless of the construction applied.

Petitioner does not propose any terms for construction (Pet., 11) but improperly applies implicit constructions unsupported by any argument or intrinsic

evidence, including at least for the terms “file” (Pet., 69; EX1003, ¶165) and “string separator character” (Pet., 24 n.7; EX1003, ¶156.)

Namely, in Ground 2 Petitioner equates a portion of a file (the “transcript window” in True) with a file, without any citations to intrinsic evidence, and without explanation or justification. (Pet., 69.) Similarly, in Ground 1, Petitioner equates a Boolean operator such as “AND” with the claimed “string separator character” as required by all claims of the ’490 patent, again without citing to intrinsic evidence or providing any justification for this implicit construction. (Pet., 24 n.7.) As discussed below (*Supra* §V.B.1.), Petitioner never sought construction of these terms and the intrinsic evidence does not support Petitioner’s implicit claim construction. Thus, both Grounds are based on implicit claim constructions of claim terms for which Petitioner failed to properly seek construction. Even if the Board considers Petitioner’s constructions, however, Petitioner cannot establish obviousness of any challenged claim..

IV. GROUND 1: Petitioner Has Failed to Show the Challenged Claims Are Obvious Over the Combination of Wilcox (EX1059), Londergan (EX1060), Raskin (EX1021), and Wu (EX1019).

The challenged claims are patentable over Petitioner’s Ground 1 challenge. In its Ground 1 combination, Petitioner cobbles together disparate disclosures from four different references: Wilcox, Londergan, Raskin, and Wu. However, this 4-way combination fails for one fundamental reason—Petitioner did not meet its

burden of showing a motivation to combine these teachings. Moreover, even if such a motivation exists (which it does not), the resulting system does not provide each and every limitation of the claims.

Petitioner's lead references are Wilcox and Londergan. Wilcox describes the functionality of Lotus Notes v 3.0 whereas Londergan discusses Lotus Notes R5. The mere fact that the references relate to a product under the same brand name is insufficient to motivate combining them and Petitioner has not adequately articulated a reason why a POSITA would have been motivated to modify Wilcox with Londergan (or vice versa). Similarly, Petitioner fails to explain why a POSITA would have been motivated to further combine the Wilcox-Londergan combination with both Raskin and Wu. Finally, Petitioner has wholly failed to even explain which specific components of the prior art references a POSITA would have combined, or why there would be a motivation to combine those specific components, to arrive at a system it alleges invalidates the challenged claims.

Even if Petitioner could establish motivation to combine, Petitioner's proposed combination does not disclose all limitations in the challenged claims. Petitioner's Ground 1 fails for all of these reasons.

A. A POSITA Would Not Have Been Motivated To Combine Wilcox And Londergan Because Lotus Notes 3.0 And Lotus Notes R5 Have Distinct Features And Are Not The Same System.

Petitioner asserts that Wilcox and Londergan are “complementary” and can be combined in part because both references disclose “entering the search strings in a single search field of a simple Search bar or a more advanced Query Builder/Search Builder.” (Pet., 13, 18.) Petitioner’s motivation to combine Wilcox and Londergan, however, spans a single paragraph and hinges on the implicit and incorrect assumption that the two references allegedly describe the same system:

A POSITA would have been motivated to combine the complementary teachings on Lotus Notes from Londergan (EX1061) and Wilcox (EX1059) to obtain a more complete and comprehensive understanding of the Lotus Notes system, its features, and its capabilities. *See RPX Corp. v. Rothschild Connected Devices Innovations, LLC*, IPR2016-00443, Paper 8 (institution decision) at 21 (PTAB Jul. 7, 2016), Paper 28 (final written decision) at 35 (PTAB Jul. 6, 2017) (finding a motivation to combine two references because, *inter alia*, they described the same system). For example, Wilcox provides additional details on the core functionality of Lotus Notes’ user interface and search functionality (EX1059, 84-85, 159-168) while Londergan provides details on new features of Lotus Notes R5, including its calendaring and auto-complete features (EX1061, 45-74, 146-147). A POSITA would

have looked to Wilcox to gain a solid understanding of Lotus Notes' core functionality, and would have looked to Londergan for details on the new features in Lotus Notes 5 that were added to this core functionality. EX1003, ¶76.

(Pet., 18.) Petitioner's vague statement that Wilcox discloses "Lotus Notes' core functionality" and Londergan describes "new features in Lotus Notes 5," implies (incorrectly) that the two references describe the same system with the same essential features. (Pet., 19.) This is false. As discussed below, Lotus Notes 3.0 described by Wilcox and Lotus Notes R5 described by Londergan are materially different with respect to the functionality Petitioner relies on. Petitioner fails to acknowledge this indisputable fact.

Worse, Petitioner never explains what the Wilcox-Londergan combination looks like or what features of Londergan Petitioner proposes to implement in Wilcox's system. Instead, Petitioner appears to imply, incorrectly, that Wilcox and Londergan describe the very same system. Petitioner's citation to *RPX Corp. v. Rothschild* is as telling as it is inapplicable. (Pet., 18.) In *RPX*, the Board found a motivation to combine two separate publications about "a localized personal communication environment known as 'Omnisphere,'" *RPX Corp. v. Rothschild Connected Devices Innovations, LLC*, IPR2016-00443, Paper 8 (institution decision) at 21 (PTAB Jul. 7, 2016), in part because Petitioner's expert credibly

opined that the “references have the same authors, describe the same system, and have similar descriptions of the functionality of the Omnisphere system.” *RPX Corp. v. Rothschild Connected Devices Innovations, LLC*, IPR2016-00443, Paper 28 (FWD) at 35 (PTAB Jul. 6, 2017). Here, Wilcox and Londergan: (1) have different authors; (2) do not describe the same system; and (3) do not have similar descriptions of the relevant functionality.

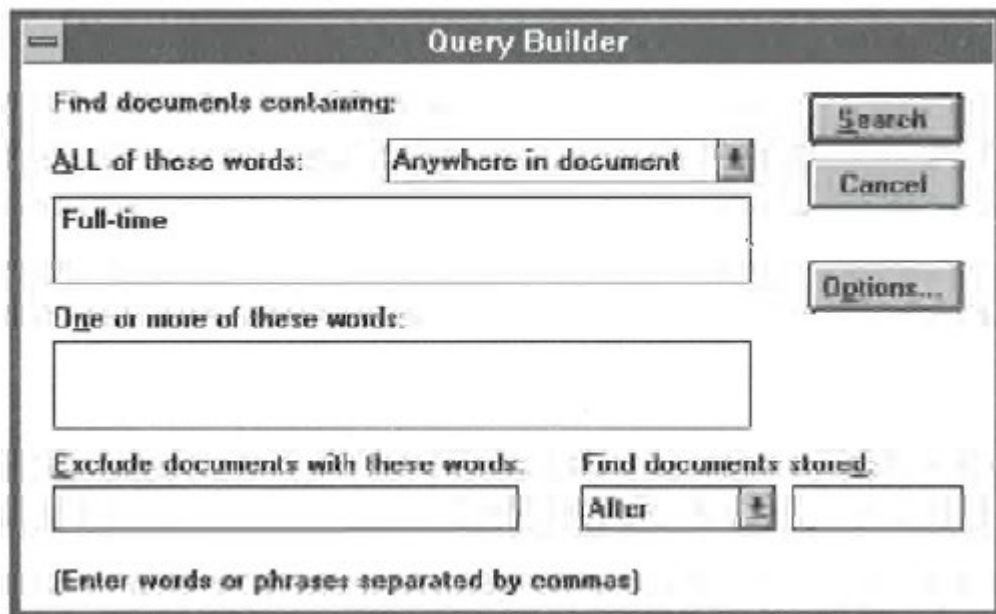
As Petitioner concedes, Wilcox and Londergan describe different versions of Lotus Notes. (Pet., 13.) Wilcox describes Lotus Notes 3.0, released in 1993, and Londergan describes Lotus Notes R5, released in 1999. (*Id.*) Not only were these two versions released six-years apart, there was at least one other major release (v. 4.0) of Lotus Notes between 1993 and 1999. (EX2001, 77.) Petitioner concedes Wilcox and Londergan are not directed to the same system by calling the references “complementary.” (Pet., 18.)

As discussed below, the relevant functionality in Lotus Notes 3.0, as described by Wilcox, and Lotus Notes R5, as described by Londergan, are starkly different, particularly with respect to the functionality Petitioner points to as allegedly rendering the challenged claims obvious. Thus, Petitioner fails to satisfy any of the requirements of *RPX* that the references have the same authors, describe the same system, or have similar descriptions of the functionalities of the system. Petitioner fails to point to any reason why a POSITA would combine teachings

related to different versions of the Lotus Notes system in which the relevant functionality is materially different. Accordingly, Petitioner fails to demonstrate that a POSITA would be motivated to combine Wilcox and Londergan.

1. Query Builder (v. 3.0) vs. Search Builder (R5).

As explained above, Petitioner suggests that Wilcox and Londergan can be combined simply because both references describe Lotus Notes. Petitioner ignores the material differences in functionality disclosed in these references, however. Specifically, the 1993 version of Lotus Notes (v. 3.0) includes a feature called Query Builder. (EX1059, 162-64.) Query Builder, pictured below, provides multiple fields where, Petitioner alleges, a user can type multi-string search queries:



(EX1059, 163; *see also* EX1003 at ¶90.)

The 1999 version of Lotus Notes (R5) eliminated the Query Builder feature and instead included a different feature with different functionality called Search Builder. (EX1061, 141-42.) Search Builder provides multiple fields, each of which can only be used for a single search string, as shown below:



(EX1061, 142.) Petitioner relies on the Query Builder functionality described in Wilcox in an attempt to show that the Wilcox-Londergan combination discloses searching on multiple strings entered into a single search field. (Pet., 13.) Petitioner separately relies on the Search Builder functionality described in Londergan for allegedly disclosing other features, such as highlighting search strings in documents (once a user opens the document from the search result window and

does another search) (Pet., 35) and searching for appointment information (Pet., 36.)

2. Search Bar

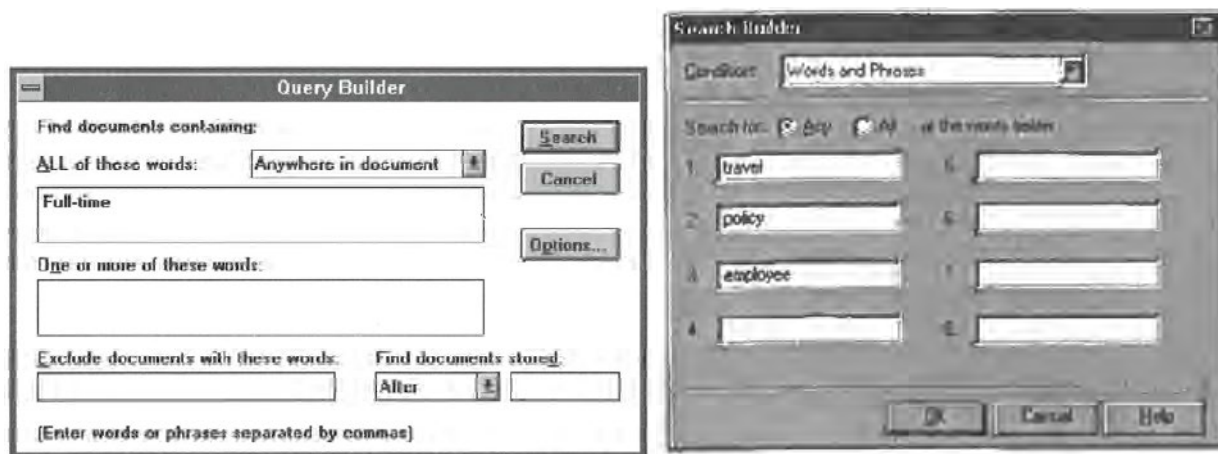
Both versions of Lotus Notes also include a separate search feature called the Search Bar, which Petitioner relies on to allegedly disclose searching on a first and second search string entered into a first search field, where the first and second strings are separated by a string separator character. (EX1059, 161; EX1061, 490-41; Pet., 13-14.) The Search bar is a search field that can perform multi-word searches only under two conditions: first, the Search bar can perform searches for exact multiword phrases if the phrase is enclosed in quotation marks (EX1059, 162; EX1061, 140); second, the Search bar can perform searches on multiple search strings that are connected with a Boolean operator, such as “AND” or “OR” (EX1059, 162; EX1061, 140-41.) In neither circumstance can the Search Bar search multiple, separate strings with a string separator character.

3. Petitioner Improperly Combined Wilcox and Londergan.

Petitioner’s proposed Wilcox-Londergan combination, to the extent it is described and can be understood, is improper. Petitioner’s expert, parroting the Petition, treats the Query Builder in Wilcox (Lotus Notes 3.0) and the Search Builder in Londergan (Lotus Notes R5) as though they have the same functionality but different names. (EX1003 at ¶64 (“For the Query Builder/Search Builder, the

user similarly enters the multiple strings ...”); Pet., 13.) As explained above, Petitioner and its expert are wrong. Tellingly, the Petition and Dr. Terveen’s declaration only include a screenshot from Wilcox of Lotus Notes 3.0’s “Query Builder” and omit any figures from Londergan showing Lotus Notes R5’s “Search Builder.” (See Pet., 12-15; EX1003 at ¶¶58-66.)

As shown below, a simple visual comparison of the Query Builder from Wilcox (Lotus Notes v. 3.0) and the Search Builder from Londergan (Lotus Notes R5) shows that the relevant functionality changed materially between versions: Query Builder (v. 3.0) allows for multiple search strings to be entered into a single search field (*Supra* §IV.A.1.) while Search Builder (R5) only allows for a single search string to be entered into each separate search field. (*Supra* §IV.A.1.)



Query Builder/Lotus Notes 3.0 (left), Search Builder/Lotus Notes R5 (right)
(EX1059, 163; EX1061, 142.)

Thus, a POSITA would not view Wilcox as providing “additional details” on Lotus Notes’ “user interface and search functionality,” as Petitioner argues (Pet., 18); rather, a POSITA would view Wilcox as being directed to an earlier, phased-out version of the program when it comes to how searches could be done. (EX2001, ¶84.) Indeed, over the years Lotus Notes moved away from a unified search field and the more modern iteration used in Petitioner’s combination, R5 (Londergan), used individual boxes for individual search terms when formatting more complicated queries. (EX1061, 142.) In contrast, the ’490 Patent teaches “a first instruction configured to read at least a first search string and a second search string entered into a *same first search field*.” (EX1001, 2:49-54 (emphasis added).) Thus, a POSITA would understand the Entourage references, as a whole, to be moving away from the claimed search technology.

That Petitioner’s motivation arguments are premised on an improper lumping together of these two different versions as if they were a single prior art reference is evident from Petitioner’s motivation to combine arguments. Petitioner asserts that “[a] POSITA would have been motivated to combine the Lotus Notes references’ teachings on the *Lotus Notes multi-string search* incrementally for the reasons stated by Raskin: speed, effectiveness, and ease of use.” ((Pet., 19) (emphasis added).) Petitioner never explains which “Lotus Notes search tool” it is referring to and therefore how, if at all, Raskin’s alleged disclosures could be

implemented into it. This also again ignores the fact that Londergan's Lotus Notes R5, the version of Lotus Notes in place closest to the priority date at issue here, had abandoned the use of a single search field in its "Search Builder" tool.

(EX1061, 142 ("Instead of entering in the text box words or phrases that you want to find, as we describe in the previous section, click Multiple Words and enter one word or phrase in the numbered boxes in the Search Builder dialog box."); *see also* EX2001, ¶86.)

In sum, a POSITA reading Londergan would not have been motivated to halt or reverse the progress of the developers of Lotus Notes R5 by reverting some of the changes that had been made to Lotus Notes R5's "Search Builder" to make it more similar to Lotus Notes earlier 3.0's "Query Builder." (EX2001, ¶86.)

Petitioner fails to provide any reason why a POSITA would considering reverting the changes to Lotus Notes R5's "Search Builder" other than hindsight bias that the "Query Builder" is a better fit for Petitioner's argument. (*Id.*)

4. Petitioner's Proposed Wilcox-Londergan Combination Represents Improper Hindsight Bias.

The only rational explanation for Petitioner's selection of disparate disclosures from Wilcox and Londergan is that Petitioner is trying to identify aspects of the claimed invention in the prior art using hindsight. Petitioner cherry-picks aspects of one version of Lotus Notes to combine with aspects of a different

version of Lotus Notes without rhyme or reason. For example, Petitioner implies that a POSITA would modify Wilcox (Lotus Notes 3.0) by including Londergan's newer calendaring and auto-complete features (from Lotus Notes R5), but would choose to keep Wilcox's phased-out Search Builder functionality (from Lotus Notes v. 3.0). (Pet., 18.) This is nothing more than impermissible hindsight used to cherry-pick different features without regard for the fact that Londergan's disclosure of the Query Builder (in Lotus Notes R5) is the more up-to-date version of advanced searching. (EX2001, ¶87.) In fact, the newer version of Lotus Notes (R5) as discussed in Londergan is more advanced in a number of other ways that would lead a POSITA to prefer the newer version when building a search user interface. (EX2001, ¶88.) But Petitioner and its expert utterly fail to address why, aside from hindsight, a POSITA would have done this. Instead Dr. Terveen merely parrots, verbatim, the Petition's conclusory statement that a POSITA would have been motivated to combine these two references directed to two separate systems. (Pet., 22; EX1003, ¶¶75-76.) Such unsupported expert testimony is not sufficient to demonstrate motivation to combine. *TQ Delta, LLC v. Cisco Sys., Inc.*, 942 F.3d 1352, 1358–59 (Fed. Cir. 2019) (“[c]onclusory expert testimony does not qualify as substantial evidence”).

Accordingly, a POSITA would not have been motivated to combine Wilcox and Londergan as required by Petitioner's Ground 1.

B. A POSITA Would Not Have Been Motivated To Combine Wilcox-Londergan With Raskin.

Petitioner acknowledges that the proposed (and improper) Wilcox-Londergan combination fails to disclose incremental searching. (Pet., 15.) For this, Petitioner turns to Raskin, which Petitioner alleges discloses incremental searching. (Pet., 15-17.) As with its proposed Wilcox-Londergan combination, Petitioner fails to demonstrate that a POSITA would have been motivated to combine Wilcox-Londergan¹ with Raskin.

¹ As discussed above (*Supra* §IV.A.4.) in reference to the improper combination of Wilcox-Londergan, Petitioner’s motivation to combine arguments of those references with Raskin fail to identify which search tool a POSITA would be modifying. (Pet., 26) (“a POSITA would have been motivated to combine the Lotus Notes references’ teachings on *the Lotus Notes search tool* with Raskin’s teachings to locate files incrementally for the reasons stated by Raskin: speed, effectiveness, and ease of use.”)(emphasis added). Petitioner has not identified exactly which search tool of Lotus Notes a POSITA would modify with the teachings of Raskin to allegedly arrive at the claimed invention. Because Petitioner improperly groups the disparate search systems together, Petitioner necessarily fails to show a motivation to combine it with the teachings of Raskin. *The Walt Disney Co., et al. v. WAG Acquisition, L.L.C.*, IPR2022-01346, Paper 12 at 24-26 (PTAB Mar. 10, 2023) (denying institution where a petitioner did not “set forth sufficient details on what aspects from [the secondary reference] would be combined with [the primary reference] and how such aspects would work together”).

Petitioner is wrong that a POSITA would look to Raskin to implement incremental search into the claimed invention, which is directed to receiving multiple, partial search strings in a single search field to search across “a set of documents.” (EX1001, cl. 1.) This is because Raskin only discusses problems with, and solutions for, searching in single file and not across multiple files. Raskin is a textbook spanning a broad list of topics related to philosophies and approaches to designing an optimal user interface, or a humane interface. (EX1021, xi-xii, 6-7.)

The pages that Petitioner cites to in Raskin to support a motivation to combine it with Lotus Notes search functionalities are in a chapter titled “String Searches and Find Mechanisms.” (EX1021, 124-27.) This entire section of Raskin discusses ways to use string searches *in a single document*. For example, Raskin discusses searching for the phrase “little Tatsu” in a “long letter” and demonstrates how the “Find and Replace” function works when a single document is being searched. (*Id.* at 124-25 (discussing that “searches usually start from the current cursor location [in the document] and continue forward through the text [of the document]”).) Indeed, the only specific search system that Raskin mentions in the pages cited by Petitioner is EMACS, which Raskin describes as “an editor used with the UNIX operating systems.” (*Id.* at 125.) A POSITA would understand that EMACS was a text editor that only allowed for search within a single document, and not across multiple documents. (EX2001, ¶45.) Indeed, Raskin’s description of

a search being conducted in EMACS where the system searches “for the first instance of that character in the chosen search direction” further confirms that such a search is being done in a single document. (EX1021, 125.)

Even Raskin’s discussion of incremental search makes clear that he is discussing it only in the context of a single document. Raskin discusses that incremental search is a “less common strategy” and is implemented in EMACS, which only searches single document. When discussing how to use incremental search, Raskin states that the “found instance” (search hit) “should appear in the middle of a display area, not at the top or the bottom. Thus, material both before and after the instance is visible; that is, the found instance is displayed in context.” (*Id.* at 127.) A POSITA reading Raskin’s discussion of implementing incremental search would understand he was referring to here, as throughout the cited pages, a search of a single document where the system can move to where there is a first hit and show the user the context around that hit. (EX2001, ¶93.) Raskin never discusses using incremental across multiple documents.

To the extent Petitioner alleges there would be a motivation to combine Wilcox-Londergan with Raskin simply because they both generally relate to searching, such a “cursory assertion that the references are drawn from the same broadly characterized field” is insufficient to meet Petitioner’s burden of identifying a motivation to combine. *Securus Techs., Inc. v. Global Tel*Link*

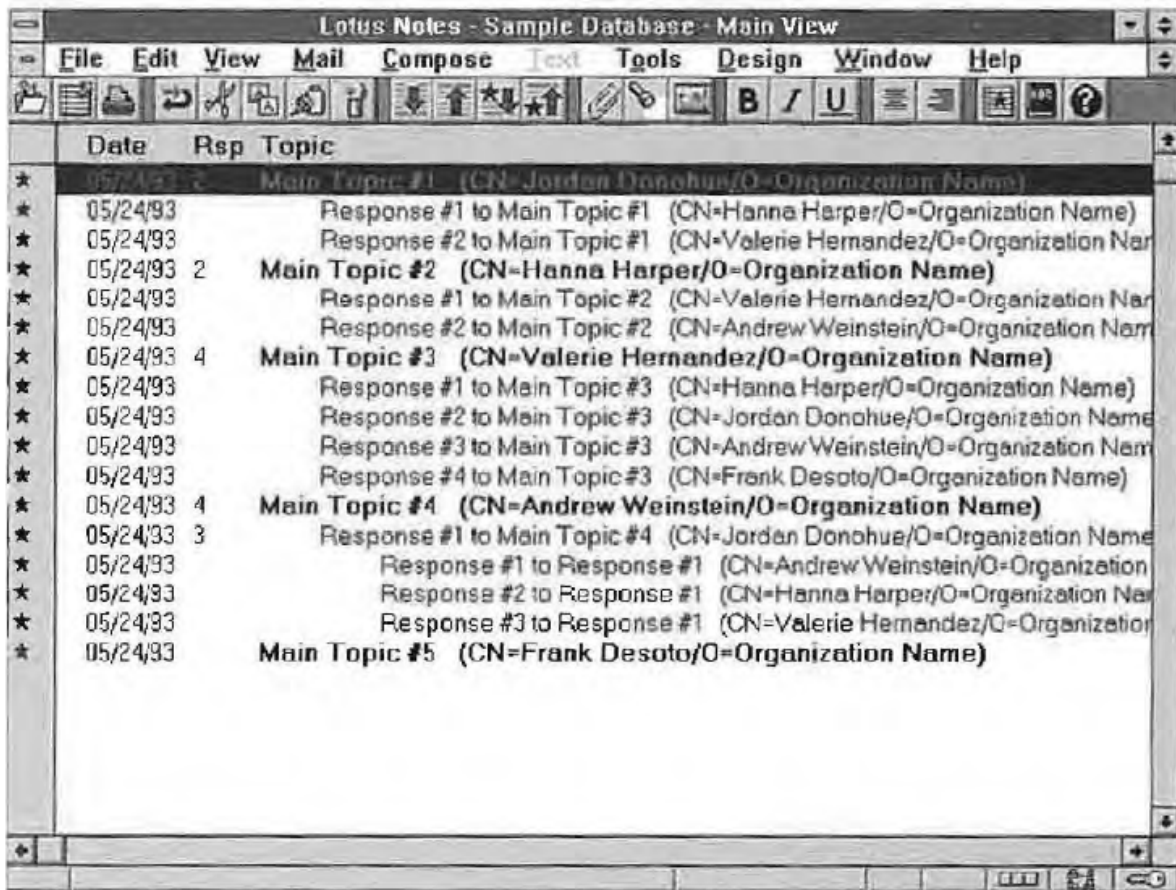
Corp., IPR2015-00155, Paper 30 at 17 (PTAB Apr. 7, 2016); *see also U.S. Well Servs., LLC v. Halliburton Energy Servs., Inc.*, No. IPR2022-01163, 2022 WL 17822529, at *8 (P.T.A.B. Dec. 20, 2022) (“The fact that the references share commonalities fails to identify what would have prompted the skilled artisan to arrive at the combination in the first place.”).

Furthermore, even if a POSITA understood the teachings of Raskin to apply to a search of multiple documents (which it does not), a POSITA would not be motivated to implement incremental search due to the serious technical difficulties with doing so. Raskin himself concludes that even if incremental searching is advantageous, “almost all interface-building tools make it easy to implement delimited searches and ***difficult or impossible*** to implement incremental searches.” (*Id.* at 126 (emphasis added).) A POSITA is not motivated to undertake a modification of the prior art with something that the art teaches would be difficult or impossible to do. *Natera, Inc. v. NeoGenomics Laboratories, Inc.*, 106 F.4th 1369, 1377-78 (Fed. Cir. 2024) (affirming a finding that there was no motivation to modify an existing array system to test for cancer because the existing knowledge in the field “put forth significant evidence of obstacles” in modifying the array system to test for cancer).

A POSITA would also not be motivated to use the teachings of Raskin with the various search systems disclosed in Lotus Notes because Raskin’s proposals

are incompatible with Lotus Notes. Lotus Notes teaches that a user can create a document, save it to a central database, have another user/employee read the original document and then respond to that document. (EX1059, 2-3.) Lotus Notes is designed around databases and views to facilitate users finding and viewing “main topic” documents and creating “response” documents through views, as shown below. (EX1059, 24-25; *see also* EX1061, 8 (“In Notes, databases contain views and folders. Both contain documents), 36 (“you use views and folders to sort through and organize your incoming and outgoing e-mail messages. Your e-mail database comes ready-made with eight views and folders.”); *see also* (EX2001, ¶96.)

The Main View of the Sample Database



(EX1059, 26.)

In other words, Lotus Notes is all about centralized collaboration around documents, files, and file structures. Raskin goes in a completely different direction, espousing the elimination of filenames and “hierarchical file structures” including directories and folders which effectively renders Lotus Notes’ functionality of hierarchical viewing of and responding to documents unusable. (See, e.g., EX1021, 119 (“With the removal of file names go all the mechanisms for dealing with file names, such as directories of file names, rules for editing file

names, ...”), 121 (“Eliminating hierarchical file structures does not mean that you have to give up structuring your stored information.”); (EX2001, ¶97.) A POSITA would understand that the lack of a server-wide file management hierarchy or file management systems taught by Raskin would severely hamper the discussions and responses envisioned by Lotus Notes and thus a POSITA would not look to Raskin to improve Lotus Notes. (EX2001, ¶97.) *See Adidas AG v. Nike, Inc.*, 963 F.3d 1355, 1360 (Fed. Cir. 2020) (affirming the Board’s conclusion that there was no motivation to combine a reference requiring stitching with a reference that has the purpose of “reduc[ing] the need for stitching.”).

Finally, to the extent that Petitioner asserts that a POSITA would be motivated to apply the incremental search of Raskin to the search tools discussed in Wilcox-Londergan on the simple basis that incremental search was known², this

²As explained by PO’s expert, Mr. Haeberli, Dr. Terveen’s discussion of incremental search as allegedly being “well-known” in the prior art is fraught with mischaracterizations. (EX2001, ¶¶39-47.) For example, the Gabbe reference (EX1014), does not provide any actual search results, instead “the user is fed back information on the number of possible matches.” (EX1014, Abstract.) Similarly, Dr. Terveen discusses Chew (EX1015) as describing “the incremental search method” even though that reference is directed specifically towards searching lists and not multiple files. (EX1015, Abstract; EX2001, ¶43.) None of the references

is insufficient and the Board should reject Petitioner’s effort to gloss over the motivation to combine these references. *Hewlett Packard Enterprise Co. v. K.Mizra LLC*, No. IPR2022-00843, 2022 WL 16556406, at *7 (P.T.A.B. Oct. 31, 2022) (denying institution where petition relied on application of “known technique,” but did not sufficiently explain why a POSITA would have made the alleged combination).

A POSITA would not have been motivated to combine one of the search tools in Wilcox-Londergan with Raskin’s incremental search because Raskin was directed to searching a single document, Raskin’s entire philosophy to user interfaces was incompatible with Lotus Notes, and Raskin noted the extreme technical difficulties with even using incremental searches.

C. A POSITA Would Not Have Been Motivated To Combine Lotus Notes With Wu.

Acknowledging that neither the Wilcox-Londergan combination nor Raskin disclose *identifying one or more documents included in both the first one or more documents and the second one or more documents*, which Petitioner refers to as an “intersection approach,” Petitioner relies on Wu to supply this missing limitation. (Pet., 11.)

cited in Dr. Terveen’s declaration disclose or render obvious incremental searching on multiple partial strings. (EX2001, ¶42.)

Wu is titled “Information retrieval from hierarchical compound documents” and is directed to a system for improving searches on large repositories, such as the internet. (EX1019, 3:24-36, 4:50-53; 12:54-58.) Wu’s search method is applied to hierarchical document repositories and the search queries in Wu are applied to that hierarchy. (*Id.*, 2:51-59.) Wu accomplishes improved search results by returning (1) documents that directly match one of the search terms; and (2) documents that indirectly match a search term by virtue of being a “child document in the hierarchy from a parent document matching all or part of the query term.” (*Id.*, 2:53-69, 3:40-43;7:24-36.) In other words, a query according to Wu’s method will return results that may not include **any** of the terms making up the query.

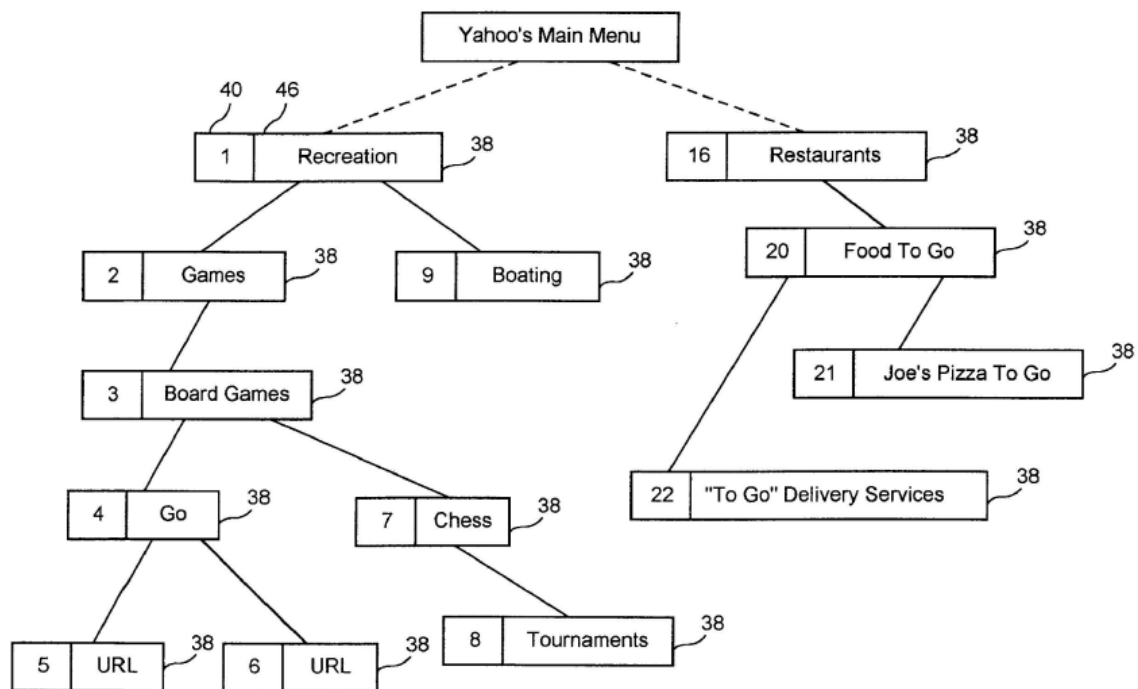
Petitioner argues a POSITA would have been motivated to combine the alleged “intersection approach” from Wu’s hierarchical search method with the Wilcox-Londergan-Raskin combination because Wu’s method “was one of a limited number of well-known predictable approaches, was simple and easy to implement, and is especially well suited for incremental searches.” (Pet., 11, 19.) Petitioner’s articulation is inadequate to explain why a POSITA would have picked Wu and combined it with Wilcox, Londergan, and Raskin to arrive at the ’490 Patent’s claimed invention. *See Personal Web Techs., LLC*, 848 F.3d at 993-994.

First, Petitioner’s motivation to combine with Wu suffers from the same fatal flaws as its other motivation to combine arguments: Petitioner has failed to

articular what search tool in which version of Lotus Notes in Wilcox or Londergan a POSITA would modify to implement the teachings of Wu. Thus, the combination necessarily fails. Furthermore, even if one could surmise which search tool in Wilcox-Londergan Petitioner is referring to, a POSITA would not have looked to Wu to improve Lotus Notes or to implement the alleged “intersection approach” described by Petitioner. Searching the Internet, as contemplated by Wu, is materially different from searching a user’s computer or an organization’s central document database as taught by Lotus Notes. (EX2001, ¶103.) Petitioner’s own cited art highlights the problems presented when searching the internet, including issues with the data itself, such as: “[d]istributed data,” “high percentage of volatile data,” “[l]arge volume,” and “unstructured and redundant data.” (EX1023, 368.) A POSITA reading Wilcox-Londergan would not need to consider such problems as the Lotus Notes program is designed to search a limited set of documents using a “full-text index” of the user’s files. (EX1061, 135.) In fact, Londergan explicitly directs the user to specify a preferred web search engine, which Lotus Notes uses to conduct web searches; that is, according to Londergan, Lotus Notes itself is not designed for web searching and uses existing external web browsers to accomplish web searches. (EX1061, 135; EX2001, ¶103.) A POSITA would not have been motivated to incorporate features from a web search system like Wu when Londergan explicitly directs users to an outside search engine designed specifically

for Internet searches, which are materially different than searching for documents on a computer or computer network. (EX2001, ¶105.)

Additionally, Wu's search method is, in part, predicated on using a specific file structure/hierarchy as it describes searching for specific nodes of the structure. (EX1019, 4:50-5:6; Fig. 2 (below).)



Such a file structure is not contemplated by Lotus Notes. (EX2001, ¶104.)

Thus, a POSITA would have no reason to look to Wu to improve searching in Lotus Notes.

Petitioner fails to describe with particularity how a POSITA would modify any of the search systems in Wilcox-Londergan to incorporate Wu's intersectional

search teachings. (EX2001, ¶105.) Dr. Terveen parrots the Petition’s conclusory argument that it “would have required only simple and straightforward software modifications” but never describes what modifications are made and how those modifications would work within any particular search system in Wilcox-Londergan. (See EX1003 at ¶79.) Focusing primarily on the alleged benefits of implementing Wu’s intersection search paradigm without explaining how the combination would work in the first place, as Dr. Terveen does, is not sufficient to satisfy Petitioner’s burden of proving obviousness. *In Re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d at 1380.

Accordingly, Petitioner failed to provide a motivation to combine Wilcox-Londergan with Wu’s alleged disclosure of intersectional search.

D. The Prior Art Does Not Disclose All Limitations of the ’490 Patent’s Claims

Petitioner does not assert that any claims of the ’490 Patent are anticipated by any single prior art reference and relies on the combination of Wilcox-Londergan, Raskin, and Wu to show that each of the challenged claims is obvious. However, the combination of these four prior art references fails to disclose certain claim limitations altogether. Because the Petition has not shown that all claim limitations are taught by the Ground 1 combinations, Ground 1 of the Petition fails.

1. The Ground 1 References in Combination Do Not Teach Two Partial Search Strings Separated By A String Separator Character In The Same Search Field (Limitation [1.10])

Petitioner relies on the combination of Wilcox-Londergan to satisfy this claim limitation with passing reference to vague disclosures from Raskin. (Pet., 25-27.) Petitioner's combination fails to disclose (1) two *partial* search strings in the same search field (2) separated by a string separator character.

Specifically, Petitioner relies on, and groups together, the Lotus Notes Search Bar functionality (*Supra* §IV.A.2.) and the “Query Builder/Search Builder” functionality (*Supra* §IV.A.1). (Pet., 23-25 (equating the “single search field” with “a blank box of a Search bar and/or blank box of a Query Builder/Search Builder dialog box”). As explained above, the Query Builder and Search Builder functionalities are meaningfully different and Search Builder (found in Lotus Notes R5) does not allow for multiple strings to be entered into a single search field. (*Supra* §IV.A.1.) Because Petitioner improperly groups together the disparate search systems and relies on them collectively (as if they are one system), when at least some of the underlying systems do not even disclose this claim limitation, Petitioner's argument that this limitation is disclosed by Wilcox-Londergan fails.

Furthermore, Petitioner does not point to any disclosure of a *partial* search string in Wilcox-Londergan; rather, Petitioner, acknowledging that there is no

such disclosure, vaguely references Raskin and states that “a POSITA would have understood or at least found it obvious that the received search strings are partial search strings” in the Wilcox-Londergan-Raskin combination because Raskin allegedly teaches incremental search. (Pet., 25-27.) But, as explained below, Raskin only teaches searching over a single string and adding characters to that one string to narrow search results in a single document. (*See, e.g.*, EX1021, 124, 126; EX2001, ¶111.) Petitioner has not cited to any suggestion or examples in Raskin of searching over two separate strings, let alone two separate partial strings.

For the string separator character portion of these limitations, Petitioner points to an example query disclosed in Londergan, which allegedly teaches that two search strings may be entered into the Search bar if they are separated by a Boolean operator such as “AND.” (Pet., 27; EX1061, 140 (disclosing search example “gold and silver”).) Petitioner contends, without citing to any intrinsic evidence for support, that in this example “AND” is a string separator and “gold” and “silver” are the two partial strings. (Pet., 15; EX1003, ¶81.) As an initial matter, Petitioner failed to request construction of the term “string separator character.” (*See* Pet., 12 (“Unless otherwise noted, Petitioner applies the plain and ordinary meaning of all claim terms below.”; EX1003, ¶52).) Furthermore, the intrinsic evidence does not support Petitioner’s implicit claim construction. First,

the '490 patent distinguishes between string separator *characters* (such as space, comma, etc.) and “modifiers” (such as AND, OR, NOT, etc.). (*Compare* EX1001, cl. 6 with cl. 22.) Second, a POSITA would not understand Boolean operators such as “AND” to function as string separator *characters*. (*See, e.g.* EX1059, 162 (discussing the use of “logical operators AND, OR, NOT, and ACCRUE to find documents,” which a POSITA would understand to be referring to a Boolean operator); EX2001 ¶110.) Indeed, a POSITA would not have understood Boolean operators like AND to be a string separator as used in the claims because when that word is used in a search, it would be read (and not ignored like the word “the” in a phrase) and the system would need to perform an action that modifies how the query is parsed. (EX2001, ¶110.) In other words, a POSITA would have known that the operator AND is a search syntax instruction rather than a string separator.³ (*Id.*)

Conceding that a Boolean operator won’t satisfy the claim elements, Petitioner asserts that it would have been obvious to a POSITA “when implementing Raskin’s teachings to perform the search incrementally to use a single-character string separator” and uses the plus sign and a comma as an

³ This understanding is consistent with the ALJ’s holding in the related ITC Investigation, 337-TA-1389, that Boolean operators such as “NOT” are commands (or, instructions for computer action). (EX2008, 31-32.)

example for separating two strings. (Pet., 25; EX1003, ¶ 81.) But Petitioner hasn't identified any mention in Raskin of searching multiple strings; therefore it would not have been obvious in view of Raskin to use a character like a plus sign or a comma and perform a search on two, separate, partial strings. (*See, e.g.*, EX1021, 124-27.) According to Raskin, a "string is a sequence of characters; ordinary English words and sentences are examples of strings" and "[s]tring searches look through a (usually lengthy) string, called the text, for an instance of a (usually brief) string that the user specifies, called the pattern." (EX1021, 124.) Petitioner has not cited any examples in Raskin of a "pattern" being two separate strings. Instead, Raskin suggests that if a user continues adding characters to a single string or pattern they will get more accurate results. (*See* EX1021, 124 (searching for the single string "little Tatsu," the "briefer string *Tatsu*," and the word "pattern."); *see also* EX1021, 126 ("The user can then add to the pattern, and the search continues without his losing the search already accomplished on the partial pattern.")) Thus, Raskin does render obvious to a POSITA using two partial search strings and separating them with a single character.

Petitioner alternatively relies on the combined Lotus Notes "Query Builder/Search Builder" to meet this claim limitation. Specifically, Petitioner argues that Wilcox-Londergan disclose this limitation by separating "two search strings" with commas in the "Query Builder/Search Builder." (Pet., 14-15;

EX1003, ¶82.) First, as discussed above, Petitioner’s combination of Wilcox and Londergan is improper because these two references are directed to different versions of the software with materially different search capabilities. (*See supra* §IV.A.1; *compare* EX1059 at 163 *with* EX1061 at 142.) To the extent Petitioner relies on Search Builder as disclosed in Londergan, this does not disclose this limitation as Londergan’s Search Builder has multiple fields, not a single field, and only a single word or phrase may be entered in each box:



(EX1061, 142 (users should “enter one word or phrase in the numbered boxes in the Search Builder dialog box.”).)

Because none of Petitioner’s cited evidence discloses these claim limitations, Petitioner’s Ground 1 fails..

2. The Combination Of Wilcox, Londergan, Raskin, And Wu Does Not Teach Searching the Index Data Structure for a [First/Second] One or More of the Plurality of Files Each Including at Least One String that Corresponds to the [First/Second] Partial Search String and Updating a Set of Matching Files of the Plurality of Files as Each Successive Character of the First Partial Search String and the Second Partial Search String is Received Respectively in the First Search Field (Limitations [1.11-1.13])

As explained above, the Wilcox-Londergan-Raskin-Wu combination does not disclose the incremental searching limitations at least because the proposed combination fails to disclose receiving two separate partial strings in a first search field wherein the two partial strings are separated by a string separator character. (*See supra* §VI.D.1.) Further, the Wilcox-Londergan-Raskin-Wu combination does not disclose incrementally searching a set of documents using two partial search strings.

Specifically, Petitioner does not explain how the Wilcox-Londergan-Raskin-Wu combination discloses incrementally searching on two partial strings. Petitioner argues that a POSITA would have understood that implementing Raskin's incremental search would have involved "treating the search strings as partial strings" because a POSITA would have known that "this is fundamentally how incremental searching works." (Pet., 30; EX1003, ¶87.) Petitioner relies on the conclusory testimony of its expert, Dr. Terveen to fill this gap in the prior art. (Pet., 30.) But Dr. Terveen's analysis fails to provide any support for the conclusion that

a POSITA would have extended the teachings of incremental search on a first partial search string to a second partial search string in the same search field. In fact, Dr. Terveen's explanation only describes obtaining results starting with a single character, and "then as the user types additional characters, the search tool narrows the results to only documents containing words that start with those two characters, and so on." (EX1003, ¶87.) Petitioner and its expert fail to identify any disclosure directed at using a second partial search string to search incrementally, or a specific teaching that would render doing so obvious. Because none of Petitioner's cited evidence discloses the claim limitations *updating the search results as each successive character of the first partial search string and the second partial string is received*, Petitioner's Ground 1 fails.

V. GROUND 2: Microsoft Has Not Shown That Any Challenged Claim Is Obvious in View of the Combination of Schwartz (EX1004), Sebastian (EX1005), Barber (EX1006), True (EX1009), and Baeza-Yates (EX1023).

The challenged claims are patentable over Petitioner's Ground 2 challenge. In its Ground 2 combination, Petitioner cobbles together even more references, arguing that the challenged claims are obvious in view of a combination of the three Entourage references (Schwartz, Sebastian, and Barber), True, and Baeza-Yates. But Petitioner has failed to show that a POSITA would be motivated to combine the teachings of these five references to arrive at the claimed invention.

Moreover, even if such a motivation exists (which it does not), the resulting system does not provide each and every limitation of the claims.

A. Petitioner Mischaracterizes Search in Entourage as described in Schwartz, Sebastian, and Barber.

Petitioner's lead references for Ground 2 are three references describing the Entourage program. Petitioner misleadingly picks and chooses among distinct features of *different types of unrelated search functionality* described in the three Entourage references (Schwartz, Sebastian, and Barber) and implicitly combines those features into a different, non-existent search system that it uses as its starting point for its obviousness arguments. Specifically, Petitioner relies on, and implicitly combines, two distinct search functionalities described in the three Entourage references: (1) list filtering; and (2) the Find and Advanced Find Commands. As discussed below, Petitioner misleadingly describes these separate functionalities as if they occur in a single search tool in Entourage.

1. The Entourage References Describe an Email System.

The Entourage References (Schwartz (EX1004), Sebastian (EX1005), and Barber (EX1006)) describe the Entourage for Macintosh 2001 email software. (EX1004, (i), 1; EX1005, (a)-(c), 171; EX1006, (ii), 281.) As Schwartz explains, "Entourage has six tightly integrated components," an email client, newsgroup reader, address book, calendar, tasks, and notes. (EX1004, 2.) The Entourage references do not describe searching generally for documents on a user's computer;

rather, the search functionality within Entourage allows a user to filter lists (e.g., emails within folders or Address Book contacts) or do more targeted searching for Entourage items (e.g., emails, contacts, notes, etc.). (EX1004, 8; EX1005, 183; EX1006, 384.)

2. Entourage's List Filtering Function.

Schwartz explains that Entourage includes a feature that allows the user to “filter most lists to show items that match simple search criteria (such as the presence of a given text string within an item title).” (EX1004, 8; *see also* EX1005, 183; EX1006, 358, 384.) Filtering is available in Entourage windows displaying lists of Entourage items, such as email messages and address book entries. For example, as shown below, a user can open an Address Book window, choose an option from the pop-up menu (“Name contains” in the example below), and then start typing a name in the text box:

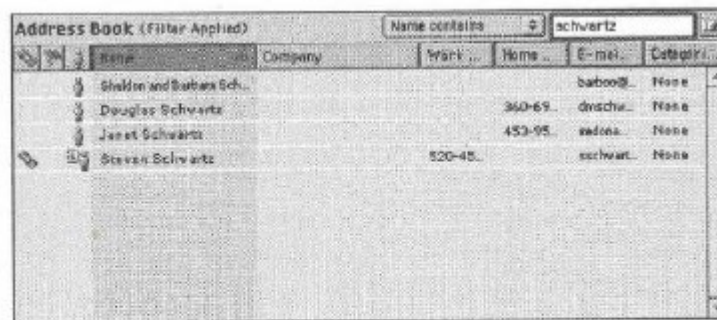


Figure 1.12 As you type the filter text, only matches are shown in the list.

Schwartz Fig. 1.12 (EX1004, 9)

The list is filtered to include only Address Book entries that include the name entered in the text box as the user types. (EX1004, 9.) None of the three Entourage references disclose filtering on more than one string; in the two Entourage references that discuss the filtering function (Schwartz and Barber), only a single string is entered into the filter text box. (*See, e.g.*, EX1004, 9 (filtering Address Book entries), 161 (filtering a message list); EX1005; EX1006, 384.)

3. Entourage's Find and Advanced Find Commands.

Entourage also includes a “slightly more powerful” search function called the Find command. (EX1004, 9; *see also* EX1006, 384-86.) To access the Find command, a user must press Apple-F or select “Find” from the Edit menu in Entourage. (EX1004, 9; EX1006, 384.) This opens a “Find dialog box” with a text entry field and a pull-down menu with options for selecting which types of Entourage items to search for (e.g., “Messages” in the example shown below). (EX1004, 9; *see also* EX1006, 385.)

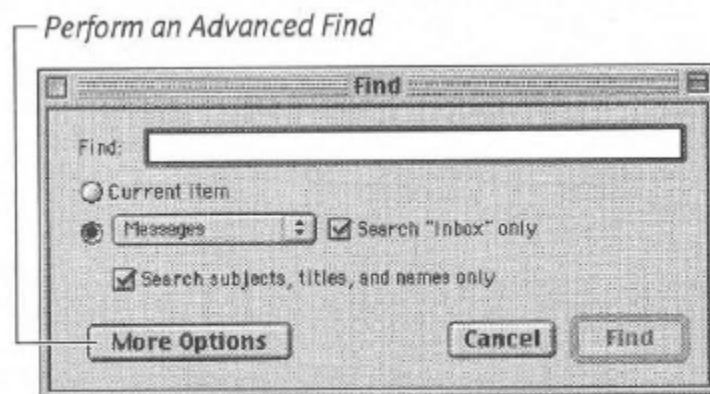


Figure 1.13 You can conduct simple searches by specifying criteria in the Find dialog box.

Schwartz Fig. 1.13 (EX1004, 9)

By pressing “More Options,” a user can access the Advanced Find command, which “allows [a user] to specify multiple search criteria; search multiple item types (messages *and* events, for example); and search any folder, regardless of the one that is currently selected.” (EX1004, 11; *see also* 385-86.) The Find and Advanced Find commands require a user to click the “Find” button after entering a search string. (EX1004, 10-11; EX1006, 385.) In other words, the Find and Advanced Find commands perform delimited searches and not incremental searches. (EX2001, ¶121.)

After clicking the Find button, “[t]he headers of all matching items are displayed in the Search Results window,” as shown below. (EX1004, 10.) The Search Results window is a separate window that pops up to display search results. (EX1004, 10.)

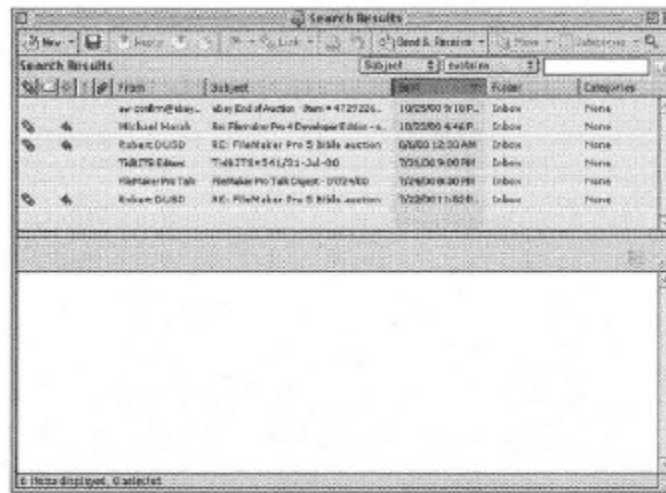


Figure 1.14 When you conduct a Find or Advanced Find, search results are displayed in a separate window. In this example, I searched for messages in the Inbox that contained the word *FileMaker*.

Schwartz Fig. 1.14 (EX1004, 10.)

The Find command can also be used to search for text *within a single item* in a list (e.g., an email message). (EX1004, 10.) To do so, either “[s]elect the item in the list (in the case of an email or newsgroup message) or open the item in its own window (in the case of a message, task, note, or Calendar event).” (EX1004, 10.) Similar to searching for items within a list or folder, Apple-F or Edit-Find is used to bring up the Find dialog box and then a user can type a string into the text box and click Find. (EX1004, 10.) In this type of search, instead of a Search Results window opening, the window displaying the item (i.e, the newsgroup message in the example shown below) scrolls to and highlights the first match, if there is one, as shown in the figure below. (EX1004, 11.)

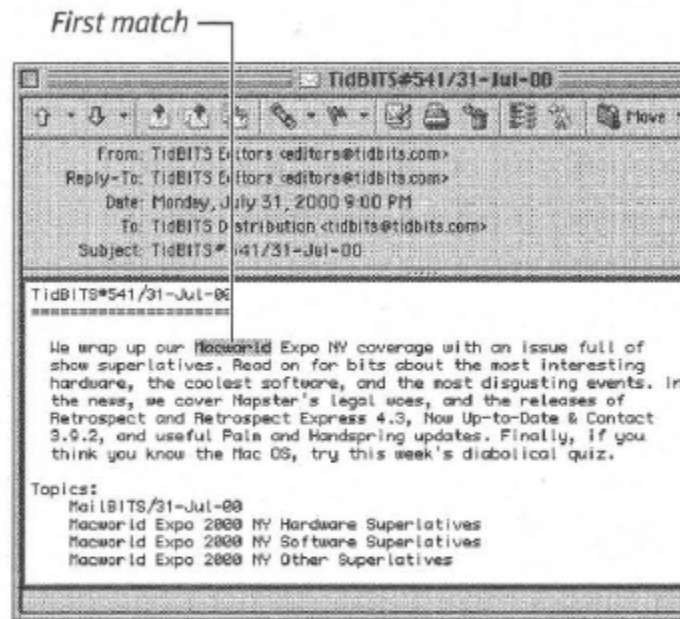


Figure 1.16 When you search for matching text within an item, Entourage highlights the first match, scrolling as necessary to display it.

Schwartz Fig. 1.16 (EX1004, 11)

The scrolling and highlighting features are only present when searching for text within a single open Entourage item, such as an email or newsgroup message. (EX2001, ¶122.)

4. Petitioner's Mischaracterization of Entourage's Search Functionality.

Petitioner mischaracterizes the Entourage references to suggest that the functionalities discussed above from filtering a list, Find and Advanced Find within a group of documents, and Find within a single document can all occur in a single search conducted in Entourage. The Petition purports to describe how a user would conduct a search in Entourage:

“Users can quickly locate files (e.g., emails, news articles, tasks, contacts) containing specific text using an incremental, quick-find search, accessible at the top of a main window.... Results are displayed ‘[a]s you type the filter text....’

(Pet., 44-45 (showing two screenshots of Entourage’s filter a list tool).)

After Entourage finds a file (e.g., email) containing the search string(s), Entourage scrolls to the portion of the document that contains the search string(s) and highlights the search string(s).”

(Pet., 47) (showing a screen shot of highlighting in a single document from Entourage’s Find command within a single document)

But the capabilities Petitioner cites are from different search tools in Entourage. Petitioner alleges that Entourage has an “incremental, quick find *search*” in its main window. (Pet., 45 (emphasis added).) Petitioner’s cited support for this assertion is the Filtering function described above. (Pet., 45; EX1003, ¶128; EX1004, 8-9, 39, 89, 161, 204-05, 209; EX1005, 173; EX1006, 358, 384.) Petitioner then misleadingly cites to portions of the Entourage references describing the Find command functionality to argue that “[a]fter Entourage finds a document (e.g. email) containing the search string(s), Entourage scrolls to the portion of the document that contains the search string(s) and highlights the search string(s).” (Pet., 50.) But the Entourage references cited by Petitioner show no such thing. Indeed, as explained

above, this is not how the Entourage references describe the filtering and Find command functionalities.

Petitioner elides over the differences between Filtering and the Find command, implying that the allegedly incremental Filtering function and the delimited Find Command functionality are somehow the same. According to Schwartz, Entourage allows you to “filter the message list” with the filtering functionality described above, or, separately, to “search for messages with the Find command.” Ex.1004, 161. These are under separate headings and described separately in Schwartz, and Petitioner fails to explain how a POSITA would understand “filtering” to be “search.”

Second, the scrolling and highlighting features that Petitioner relies on are not associated with the incremental Filtering functionality, but only with the Find command. (*Supra* §V.A.3.) Worse, Entourage only scrolls to and highlights the first search result *when using the Find command to search for text within a single Entourage item*, such as an email or newsgroup message; Entourage does not scroll to or highlight search results when using the Find command to search for items within a list. (*Supra* §V.A.3, EX1004, 11, 163.)

Furthermore, none of the three Entourage references include any disclosures that would even imply that one could use multiple search strings in either the Filtering function or when using the Find command; every example in each of the

three references refers to entering a single string in the text box. (*See, e.g.*, EX1004, 8 (typing in just “apple”), 9 (typing in just “scwartz”), 10 (typing in “macworld”); EX1005, 173; EX1006, 384 (describing filtering messages by typing in “humor”), 385 (typing “Rosebud” into the Find tool), 387 (Fig. 11-7 showing individual strings being typed into the “Advanced Find” tool).)

Thus, while Petitioner paints Entourage as allowing a user to do all of these things in a single search, Entourage does no such thing. Instead, a user could filter a list of contacts and receive results as they type the contacts’ name in one system. Or, in another system, Find and Advanced Find, a user can type a single string, hit enter, and receive a list of search results, but without any highlighting to the search results. If a user goes into yet a different window by opening a single item and then utilizing the Find toolbar again, the user can enter a single search string and the system will highlight the first match.

As Ground 2 is founded on Entourage as the starting system that a POSITA would modify, and Petitioner’s description of its searching capabilities is wholly divorced from what the Entourage references teach, Ground 2 should fail.

B. A POSITA Would Not Have Been Motivated to Combine the Entourage References with True.

Petitioner acknowledges that the Entourage references fail to disclose the receiving a first partial search string in a first search field; receiving a second partial

search string in the first search field; and wherein the first and second partial search strings are separated by a string separator character in the first search field limitations and relies on True (EX1009) to fill that gap. Petitioner also relies on True as allegedly teaching a POSITA to “incrementally locate (i.e., locate as the user types or modifies their search query) files.” (Pet., 53.) But Petitioner fails to demonstrate why a POSITA would be motivated to look to True because (1) True fails to search for the presence of any search string in any “files” as required by the claims, let alone incremental search of files; and (2) True is directed to a wholly different purpose than Entourage and is inherently imprecise. In addition, Petitioner fails to prove motivation to combine True with the Entourage references because it fails to identify any specific search tool within Entourage that would be the starting point for modification to incorporate the teachings of True. Thus, Petitioner has not provided “specific reasoning, based on evidence of record” (*Magnum Oil Tools*, 829 F.3d at 1380) including a reasoned “explanation as to how or why the references would be combined to produce the claimed invention” (*Trivascular*, 812 F.3d at 1066), and Petitioner’s combination of the Entourage References and True fails.

1. True Does Not Teach Incrementally Searching a Set of Files for Multiple Search Strings.

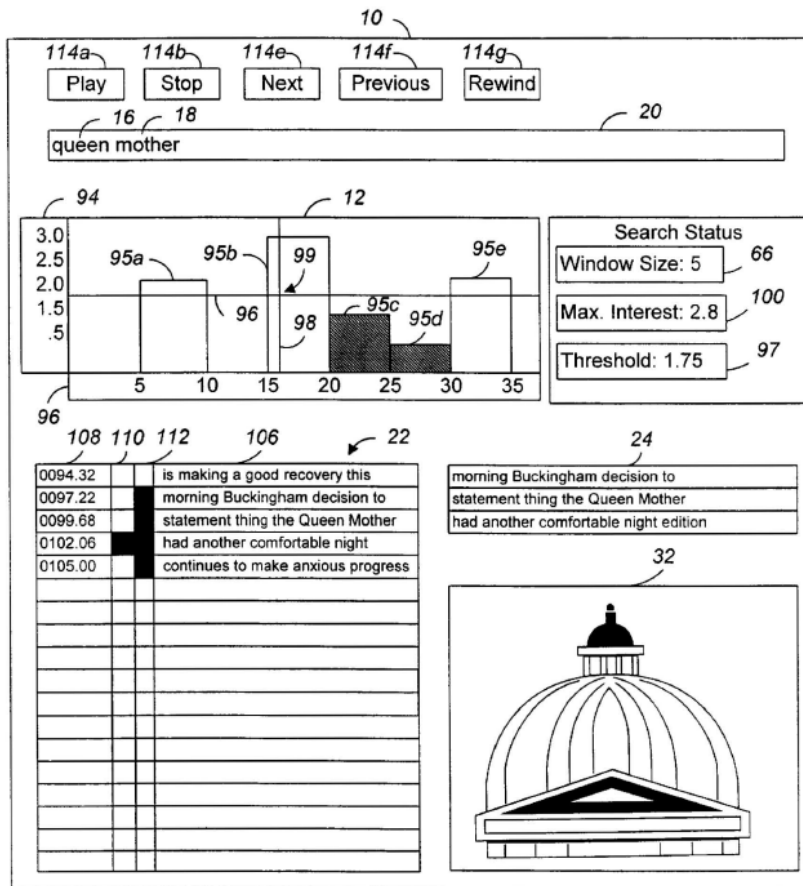
True states that it is directed to “search[ing] a file that represents underlying raw information ... to find parts of the file that represent material of interest.”

(EX1009, 1:39-41.) True seeks “to make it easier for a user to navigate through the results of a search made on a large file” by providing a graphical user interface that “allows the user to understand and control the relationship between hits and levels of confidence that the hits are representative of underlying information that is of interest to the user.” (EX1009, 3:20-27.) True is specifically directed to searching for words within a computer transcription of an audio file, taking into account the confidence level of the accuracy of the speech-to-text transcription for each word in the transcript. (EX1009, 1:55-60, 5:40-64.)

In yet another example of gap-filling, Petitioner does not identify any express disclosure that True separately searches on multiple words entered into the search box; rather, the Petition simply states this as though it is a fact, citing to the Terveen declaration. (Petition, 50; citing EX1003, ¶135.) But nor does Dr. Terveen point to any explicit disclosure of searching separately for each search string; rather he simply states, without support, that a “POSITA would have understood” that True discloses this. (EX1003, ¶135.) But true does not disclose this at all.

The output of True’s “search” method are not a list of traditional search results showing where the search string or strings appears, but rather a “result window 12 [which] indicates the degree to which portions of the transcript 14 match the query particularly well.” (EX1009, 3:37-40.) Each recognized word in the transcript includes a confidence level, “a number between zero and one indicating an estimated

likelihood that the recognized word is correct.” (EX1009, 4:21-33.) To perform a search, the user specifies a window size in seconds to define “transcript windows” which “correspond[] to a segment in the digitized audio” and then enters a query. (EX1009, 4:39-65.) True’s query results consist of a graphical display showing the “interest level” of transcript windows, calculated “as the sum of confidences of each word in the transcript window matching a word in the query.” (EX1009, 5:29-50.) True displays a “bar chart graph of the query results in result window 12,” as shown in True’s Figure 2 below. (EX1009, 5:52-54.)



True Figure 2 (EX1009)

Because True's method calculates the interest level as the sum of confidences for each word, it is possible for transcript windows to meet the threshold for "interest" even if those windows do not include any instance of one of the words in the search query. For example, True explains that "multiple occurrences of a search word are all given weight in the interest level computation." (EX1009, 5:40-50.) True's bar chart in the result window, therefore, provides a visual indication of which transcript windows *might* contain *at least one* of the words of interest entered into the search query but does not provide a list of search results pointing to portions of the transcript that necessarily include all words in the search query. (EX2001, ¶132.) Thus, True is not searching for the presence of multiple search strings in documents and locating documents that contain both partial search strings, as required by the claims.

Further, to the extent True describes searching at all, it is searching within a single document, not a set of documents. Petitioner, relying solely on its expert's conclusory testimony, contends that True discloses searching for documents because True allegedly teaches that searching within a transcript (i.e., a single document) of an audio file returns as results *portions of the transcript*, referred to as "transcript windows." (Pet., 50; EX1003, ¶138.) In fact, Dr. Terveen equates "documents" with "transcript windows" without providing any citations to intrinsic evidence, and without explanation or justification for his implicit construction. (EX1003, ¶138.)

Nor does Dr. Terveen even explain what a “transcript window” is. As explained in above, True’s transcript windows are “portions of the transcript” defined by a window size in seconds (e.g., five seconds). (EX1009, 3:40-43, 4:55-65.) In other words, these transcript windows are not themselves documents, but rather portions of a document. Dr. Terveen does not even attempt to justify his implicit construction of “document” to correspond to only a *portion* of a document.

2. A POSITA Would Not Have Looked To True’s “Interest Level” Method To Modify the Entourage References.

Petitioner’s only argument regarding motivation to combine Entourage with True is conclusory and is actually contradicted by the teachings of True:

A POSITA would have been motivated to combine the Entourage references’ teachings to incrementally search at least one search string entered in a single search field ... with True’s express teachings to do so for multiple search strings entered in the search field ... at least because a POSITA knew that such a multi-string search would return more relevant results than a single-string query.”

Pet., 56 (internal citations omitted).

But Petitioner’s stated motivation for combining the three Entourage references with True, that “a multi-string search would return more relevant results than a single-string query,” fails to explain why a POSITA would look to True specifically to improve Entourage in this way. (Pet., 56; EX2001, ¶135.) Indeed,

Entourage and True are directed to completely different purposes and POSITA would not have considered True when seeking to modify Entourage as stated. *See In re Klein*, 647 F.3d 1343, 1350-51 (Fed. Cir. 2011) (“If [a reference] is directed to a different purpose, the inventor would accordingly have had less motivation or occasion to consider it.”) (citing *In re Clay*, 966 F.2d 656, 659 (Fed. Cir. 1992)).

As explained above, Entourage, as described by Schwartz, Sebastian, and Barber is an email system (EX1004, 2; EX1006, 281) with a Filtering function and a Find command. (*See supra* §V.A.1; *see also* EX1004, 8-14; EX1005, 173; EX1006, 384-387.) True’s interest level method has a completely different purpose than the Entourage references. Rather than filtering or searching for email messages, address book contacts, or text within a single message, True provides a “graphical display of query results” that allows a user to have a “visual overview of the extent of which segments of [a] transcript relate to topics of interest.” (EX1009, 2:10-13; 1:38-42 (stating invention is directed to “interactive searching” of a file “that represents underlying raw information” and finding “parts of the file that represent material of interest in the raw information.”); EX2001, ¶137.) In other words, True is directed to searching a single text file, which itself is a computer transcription of an audio transcript produced by a speech recognizer, and then using the search results to produce a graph showing which “parts of the file are of particular interest.” (EX1009, 1:55-60; 3:20-33; EX2001, ¶137.) Using this information the user can then

read that portion of the transcript or play the underlying audio. (*Id.*, 3:43-46.) True never describes searching multiple files, or any type of file other than a transcript from an audio file. Furthermore, True is directed to solving the problem of how to locate information you are seeking when the underlying data that you are searching (i.e. a transcript of an audio file created by a speech recognizer) does not accurately represent what was actually said in the audio file. (*Id.*, 1:18-27 (“the searchable information 1 may not always accurately represent portions of the underlying information 5 that the user is trying to find. This is the case, for example, if the searchable information 1 is a text file generated by a speech recognizer from underlying information 5 representing speech, because the words produced by a speech recognizer are sometimes not the words that were spoken.”).) There is no suggestion that Entourage had such a problem. It would not make sense for a POSITA to look to True to modify Entourage simply given the respective corpus of data on which searches are performed. (EX2001, ¶138.)

To the extent Petitioner is suggesting that a POSITA would look to True’s interest level method to provide more “precise searches” or to “return more relevant results than a single-string query,” Petitioner is wrong. (*See* Pet., 56; EX1003, ¶136.) A POSITA would have understood that True, by the nature of its invention, is imprecise. (EX2001, ¶138.) As discussed above, True is designed to search a text file that is only a representation of a separate underlying file (e.g. an audio file) and

this representation has potentially dubious accuracy. (EX1009, 1:18-27.) The search performed by True does not provide definitive results of individual hits like Entourage and the emails it returns. Instead True provides *potential* hits with “levels of confidence that the hits are representative of underlying information that is of interest to the user.” (EX1009, 3:19-27; EX2001, ¶138.) A POSITA would have understood the confidence levels to refer to the confidence that the word the user searched in the text file was actually captured correctly by the speech recognizer and actually corresponds to that word in the audio file. (EX1009 at 1:61-63 (“Confidence levels may be descriptive of degrees to which parts of the file correspond to parts of the raw information.”); EX2001, ¶138.) Searching on this kind of data is inherently imprecise, especially in comparison to Entourage which searches written text in actual emails and returns results where the query text directly matches the email text. (EX2001, ¶138.) If a POSITA wanted to make the search functionality in Entourage more precise, it would not have made any sense to turn to True’s inherently imprecise interest level method. (EX2001, ¶138.).

3. Petitioner Fails To Explain How A POSITA Would Modify Entourage To Include True’s Disclosure.

Similar to the deficiencies in Ground 1, in Ground 2 Petitioner fails to describe with particularity how a POSITA would modify a specific search feature in

Entourage to incorporate True's disclosure of allegedly receiving two "search strings" in a single field. (EX2001, ¶490.)

Specifically, Petitioner never describes what specific modifications a POSITA would make other than saying Entourage's search field would receive two partial search strings, thereby leaving unexplained how the resulting interface would work. (*Id.*)

For example, Dr. Terveen describes three different "search" approaches as allegedly disclosed in the Entourage references, including the "quick-find" filtering of lists, the Find command, and the Advanced Find command. (EX2001, ¶140; EX1003, ¶¶126-27.) While Petitioner relies on the quick-find filtering as allegedly performing "incremental" searches, and Dr. Terveen admits that the Find and Advanced Find command features perform only delimited searches. (Pet., 48-50; EX1003, ¶¶126-27.) However, the quick-find filtering, as shown in Petitioner's examples, only filters on fields such as subject lines and names, and does not search on the text of email messages or notes, for example; therefore, Petitioner's proposed combination necessarily relies on the Find or Advanced Find commands to meet the challenged claim limitations. (*See, e.g.*, EX1004, 8, 9 (results are shown as "you type filter text"), EX1005, 183 (figure showing quick-find as "Address Book (filter applied)"); EX1006, 383-87; *see also* EX2001, ¶140.) In Entourage's delimited search tools Find and Advanced Find, results are only displayed after a user selects

a “Find” button and “search results are displayed in a separate window.” (See EX1003, ¶127; EX1004, 10, Fig. 1.14.) Petitioner does not even attempt to explain how an incremental search would be implemented in the context of the Find or Advanced Find commands where these features include a separate Find command window, a separate search result window, and also require the user to click “Find” to initiate the search. (EX2001, ¶140.)

Further, a POSITA would not consider using True’s alleged incremental search with Entourage’s Find and Advanced Find features, given that searching through all files and text in Entourage is known to take a long time even with the Find and Advanced Find tools. (EX2001, ¶141; EX1006, 385 (“you can choose All Items—but be prepared to rent a video during the time Entourage will take to search its entire multimegabyte database.”) Petitioner’s analysis does not include any explanation as to why a POSITA would consider implementing True’s confidence/interest level search method in such conditions or any explanation as to how a POSITA would combine the references to avoid such a major obstacle. This is insufficient to satisfy Petitioner’s burden of proving obviousness. *In Re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d at 1380.

Accordingly, Petitioner fails to provide a motivation to combine Entourage with True’s disclosure of searching on two strings in one search field.

C. A POSITA Would Not Have Looked to Baeza-Yates to Improve Entourage.

Acknowledging that neither the Entourage references nor True disclose *identifying one or more documents included in both the first one or more documents and the second one or more documents*, which Petitioner refers to as an “intersection approach,” Petitioner relies on Baeza-Yates to supply this missing limitation. (Pet., 56.)

Baeza-Yates is a textbook intended for information retrieval students with an emphasis on retrieving information from the Internet. (EX1023, v (“Information retrieval (IR) has changed considerably in recent years with the expansion of the World Wide Web”), 2 (“IR was seen as a narrow area of interest mainly to librarians and information experts ... In the beginning of the 1990s, a single fact changed once and for all these perceptions—the introduction of the World Wide Web.”).) According to Baeza-Yates, information retrieval is using “natural language text” to “retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible.” (EX1023, 2.) Baeza-Yates distinguishes IR from data retrieval, which is “determining which documents of a collection contain the keywords in the user query.” (EX1023, 1; EX2001, ¶143.) Thus, IR is directed to finding *relevant* information about a *topic* whereas data retrieval, like that used in the Entourage References search tools, is directed to

finding specific instances of a search term in documents like emails, address books, and notes.

Petitioner fails to demonstrate why a POSITA seeking to improve a search tool in Entourage would look to Baeza-Yates for how to implement multiple partial string searches incrementally using the “intersection approach.” Petitioner’s only argument of why a POSITA would be motivated to combine Baeza-Yates with Entourage is that the intersection approach allegedly disclosed in Baeza-Yates is “one of only a limited number (a few at most) of known ways to locate documents containing multiple search strings, was simple and easy to implement, and was especially well suited for an incremental search.” (Pet., 57.) But Petitioner fails to explain why a POSITA would look to Baeza-Yates in the first place. As discussed above, Baeza-Yates is directed to IR which the reference itself distinguishes from data retrieval (used in traditional searches of documents like those in Entourage). (EX2001, ¶144.) Thus, A POSITA seeking to modify or improve a data retrieval system like Entourage would not be motivated to look to an information retrieval textbook for teachings or suggestions. (EX2001, ¶144.) Furthermore, as Petitioner’s own cited art describes, searching the Internet and performing information retrieval is on a completely different scale and comes with its own challenges when compared to searching a local computer or an email client as taught by Entourage. (EX1023, 373 (“...in the Web, all queries must be answered without accessing the text (that is,

only indices are available). Otherwise, that would require either sorting locally a copy of the Web pages (too expensive) or accessing remote pages through the network at query time (too slow).”; EX1023, 373.) Given the difference in scope and purpose of data retrieval versus information retrieval, a POSITA would not have been motivated to incorporate features from an Internet-focused information retrieval textbook like Baeza-Yates when looking to modify a data retrieval system like Entourage. (EX2001, ¶144.)

As with True, because Petitioner failed to identify exactly which Entourage search tool would be the starting point for any modifications Petitioner’s motivation to combine with this reference is also deficient. Dr. Terveen parrots Petitioner’s argument “[a] POSITA would have reasonably expected to succeed in incorporating this functionality from Baeza-Yates in the Entourage search tool at least because it would have required only simple and straightforward software modifications.” (Pet., 60; EX1003 at ¶490.) However, Petitioner does not even say which “search tool” in Entourage it is referring to. Petitioner does not provide any implementation details or suggest which functionalities would be changed in the combination of Entourage and Baeza-Yates. This is insufficient to satisfy Petitioner’s burden of proving obviousness. *In Re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d at 1380.

Accordingly, Petitioner fails to provide a motivation to combine Entourage with Baeza-Yates’ disclosure of intersectional search.

D. Entourage In View Of True And Baeza-Yates Does Not Disclose All Limitations of the '490 Patent's Claims

Petitioner does not assert that any claims of the '490 Patent are anticipated by any single prior art reference and relies on the combination of the three Entourage References, True, and Baeza-Yates to show that each of the challenged claims is obvious. However, the combination of these five prior art references fails to disclose certain claim limitations altogether. In particular, Petitioner fails to establish that the combination discloses *updating a set of matching files of the plurality of files as each successive character of the first partial search string and the second partial search string is received respectively in the first search field [1.13] and receive a second partial search string in the first search field [1.9]* Because the Petition has not shown that all claim limitations are taught by the combination of the Ground 2 references, Ground 2 necessarily fails.

1. The Combination Of Entourage, True, And Baeza-Yates Does Not Teach *updating a set of matching files of the plurality of files as each successive character of the first partial search string and the second partial search string is received respectively in the first search field (Limitation [1.13])*.

Neither the Entourage references, True, nor the proposed combination discloses performing incremental searching on a set of documents.

Petitioner appears to rely solely on the Entourage references for this claim limitation. Under the heading “Updating Search Results as Each Character Is

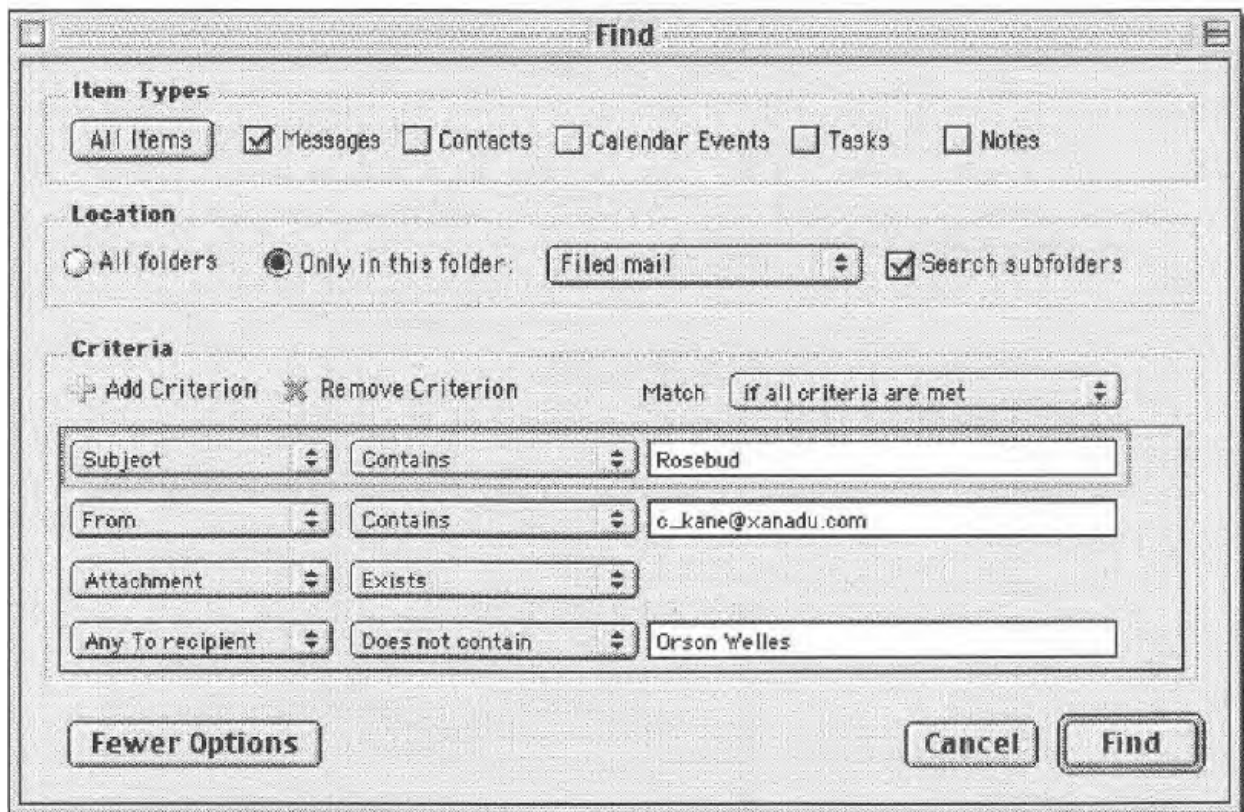
Received (Performing An Incremental Search)”, Petitioner states that “The Entourage references teach what was generally well known in the art: incrementally locating and displaying documents (e.g., contacts, emails, news articles, tasks, etc.) containing partial search string(s) as each successive character of the partial search strings is received.” (Pet., 63, 88, 93.) The Entourage references disclose two tools to find information: (1) filtering lists; and (2) searching for Entourage items or for words within a single Entourage item using the Find or Advance Find commands. (*Supra* §V.A.3.) As discussed above, the latter functions of Entourage do not search incrementally and instead a user must hit a “search” button to initiate a search. Therefore, the Find and Advanced Find functions of Entourage do not supply these claim elements.

While Petitioner fails to identify what the “lead” search tool is for Ground 2, to the extent the Petitioner is implying (without expressly stating) that the filtering function performs incremental *searches* for documents, this is inconsistent with the teachings of the Entourage references that distinguish filtering from searching. (*Supra* §§V.A.1-2.) Because Entourage’s filter a list function is not “searching” as described in the Entourage references, it does not teach this limitation.

2. The Combination of Entourage, True, and Baeza-Yates Does Not Disclose Searching On Multiple Partial Strings.

Neither the Entourage references, True, nor the proposed combination discloses performing *searching on multiple*, partial *strings*. Petitioner primarily relies on the Entourage references to disclose searching for multiple strings, (Pet., 66.) but the only examples Petitioner points to as support for its argument in Entourage do not disclose this limitation. Specifically, Petitioner relies on (1) filtering of “Orson Welles” in the Entourage filtering function and (2) a misleading citation to the index in Entourage. Neither examples disclose a search on multiple, partial strings. (Pet., 61.)

According to Petitioner, the Entourage references “at least suggest[]” incrementally searching on multiple strings by using the Find command to search on the string “Orson Welles.” (Pet., 61.)



Barber Figure 11-7

(EX1006, 387.) First, the Find command performs only delimited searches, not incremental searches as required by the claims. (*Supra* §V.A.3; *see* Pet., 66 (citing EX1006, 387 as “at least suggest[ing] incrementally searching for files containing two partial search strings entered in a single search field”).) Further, there is no evidence that when a user types “Orson Welles” into this Find command text box that a search is performed on “Orson” and “Welles” separately as opposed to searching on the full text string “Orson Welles.” Therefore, Petitioner fails to establish that the Entourage references alone disclose this limitation.

Petitioner then argues that the references “expressly disclose ‘searching for text strings’” thereby “at least suggesting to a POSITA to receive multiple search strings in the single search field of Entourage’s incremental Quick Find search.” (Pet., 53 (emphasis in original).) The alleged disclosure of “text strings” that Petitioner cites to is an entry in the index of Schwartz that refers the reader back to pages 10 and 11 of Schwartz, which discusses the filtering a list and Find commands. (Ex. 1004, 267 (under “Find Command”, “searching for text strings”), 269 (under “items”, “searching for text strings in”).) On pages 10 and 11, there are multiple discussions of searching a single text string, but no disclosure of searching multiple text strings. (*Id.*, 10-11 (“To search the current item for a text string... The first instances of the text string (if there is one) is highlighted”).) A POSITA would understand that the index in Schwartz was not disclosing searching for multiple separate partial strings and instead was referring to the fact that the referenced pages discussed multiple instances of searching a single search string. (EX2001, 152.)

Petitioner’s improper fallback argument is that it would have been obvious to search on multiple partial search strings “at least because it was within a POSITA’s knowledge to do so.” (Pet., 66.) Petitioner provides no explanation as to why this would have been obvious and thus its conclusory arguments must fail.

Finally, Petitioner argues that the combination of the Entourage references with True discloses this limitation. (Pet., 66.) As discussed above (Supra §V.B.1),

True does not disclose incrementally *searching* for *documents* containing multiple partial strings. (EX2001, ¶154.) Indeed, True does not search for documents, nor does it provide “search results.” Instead, True looks at a single text file that is created by a speech recognizer receiving an audio file. (See EX1009, 3:34-37, 4:5-20.) True’s results are not a list of documents that contain the partial strings in the inputted query; rather, True provides results in the form of transcript windows that may or may not contain all strings in the query and an “interest level” that indicates the sum of the “confidence levels” for each word in the query. (See EX1009, 3:37-49, 4:34-38.) Therefore, even to the extent True could be understood to describe performing an incremental search for documents (which PO does not concede), it does not provide search results that necessarily include all search strings in the query. Thus, neither Entourage nor True, to the extent such a combination is even described in the Petition, discloses *receiv[e/ing] a second partial search string in the first search field*. (EX2001, 155.)

3. The Combination Of Entourage, True, And Baeza-Yates Does Two Partial Search Strings Separated By A String

**Separator Character In The Same Search Field
(Limitations [1.8]-[1.10])**

Petitioner has not established that the Ground 2 combination discloses searching two *partial* search strings separated by a string separator character in the same search field.

Petitioner relies on Entourage References for the “two partial strings” claim limitation. (Pet., 66.) Specifically, Petitioner asserts that the combination of (1) the list filtering functionality being incremental and therefore inherently partial and (2) the alleged disclosures in Entourage of a plurality of search strings would render this claim element obvious. But filtering a list is distinguished from searching in Entourage, and Entourage does not disclose searching multiple partial search strings in one field. (*Supra* §V.A.1-3.). Then, acknowledging that there is no disclosure of a *partial* search string in Entourage, Petitioner vaguely references that this claim element “would have been obvious” to a POSITA “at least because it was within a POSITA’s knowledge to do so.” (*Id.*) Petitioner cannot rely on the general state of the art without further explanation to satisfy this missing claim element.

4. The Combination Of Entourage, True, And Baeza-Yates Does Not Teach Highlighting A Plurality Of Occurrences Of The First And Second Partial Search Strings In The First File (Claim 11)

Petitioner has not established that the proposed combination discloses *highlighting a plurality of occurrences of the first and second partial search strings*. Petitioner relies solely on one of the Entourage references, Schwartz, for these claims, but that reference does not disclose the claim element. (Pet., 75-79.) The disclosure that Petitioner cites for this limitation is using the Find and Advanced Find command functionality to search for a single string within a single Entourage item. (Pet., 75-78; EX1003, ¶171; *see also* EX1004, 10-11 (describing searching the “current item” using the “Find” feature), 162-63 (describing searching the “current message” using the Advanced Find feature); *see supra* §V.A.3..) However, as explained above, the “Find” and “Advanced Find” features perform only delimited searches and not incremental searches as required by these claims. (*Supra* §V.A.3; *see also* EX1003, ¶127).) And the filtering functionality described in the Entourage references does not highlight the occurrences of the first and second search strings. (*Supra* §V.A.2.)

Therefore, the combination of the Entourage references, True, and Baeza-Yates does not disclose highlighting a plurality of occurrences of the first and second partial strings in the selected document.

VI. CONCLUSION

In light of the above, Microsoft has not shown that there is a reasonable likelihood that at least one challenged claim is unpatentable, and Microsoft's petition should be denied.

Date: April 28, 2025

By: /Christie L. Larochelle/
Christie L. Larochelle (Reg. No. 75,020)
clarochelle@goodwinlaw.com
GOODWIN PROCTER LLP
100 Northern Avenue
Boston, MA 02210
Tel: (617) 570-1000
Fax: (617) 523-1231

Attorney for Patent Owner

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** complies with the type-volume limits of 37 C.F.R. § 42.24(b)(1) because it contains 13,415 words according to the word-processing system used to prepare this Response, excluding the words in the Table of Contents, Table of Authorities, List of Exhibits, Certification Under §42.24(d), and Certificate of Service, as set forth in 37 C.F.R. § 42.24(a)(1).

Date: April 28, 2025

/Christie L. Larochelle/
Christie L. Larochelle (Reg. No. 75,020)
clarochelle@goodwinlaw.com
GOODWIN PROCTER LLP
100 Northern Avenue
Boston, MA 02210
Tel: (617) 570-1000
Fax: (617) 523-1231

Attorney for Patent Owner

CERTIFICATE OF SERVICE

A copy of this Patent Owner's Preliminary Response has been served on
Petitioner's attorneys of record as follows via electronic mail:

Todd M. Siegel, Reg. No. 73,232
todd.siegel@klarquist.com

Amy C. Haspel, Reg. No. 78,385
amy.haspel@klarquist.com

Andrew M. Mason, Reg. No. 64,034
andrew.mason@klarquist.com

Samuel B. Thacker, Reg No. 78,633
samuel.thacker@klarquist.com

KLARQUIST SPARKMAN, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon, 97204
503-595-5300 (phone)
503-595-5301 (fax)

Msft-X1-IPRs@klarquist.com

Attorneys for Microsoft Corporation

Dated: April 28, 2025

By: /Christie L. Larochelle/
Christie L. Larochelle (Reg. No. 75,020)
clarochelle@goodwinlaw.com
GOODWIN PROCTER LLP
100 Northern Avenue
Boston, MA 02210
Tel: (617) 570-1000
Fax: (617) 523-1231

Attorney for Patent Owner