Trials@uspto.gov
571-272-7822

Paper 47
Date: June 18, 2026

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

X1 DISCOVERY, INC.,
Patent Owner.

IPR2025-00253
Patent 7,370,035 B2

Before WILLIAM V. SAINDON, SCOTT C. MOORE, and LISA A. MURRAY, *Administrative Patent Judges.*

MOORE, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining No Challenged Claims Unpatentable
35 U.S.C. § 318(a)

I. INTRODUCTION

A. Background and Summary

Petitioner Microsoft Corporation filed a Petition requesting an *inter partes* review of claims 1–10 of U.S. Patent No. 7,370,035 B2 (Ex. 1001, "the '035 patent"). Paper 1 ("Pet."). The Board instituted an *inter partes* review under 35 U.S.C. § 314. Paper 14 ("D.I."). Thereafter, Patent Owner X1 Discovery, Inc. filed a Response (Paper 21, "PO Resp."), Petitioner filed a Reply (Paper 27, "Pet. Reply"), and Patent Owner filed a Sur-reply (Paper 37, "PO Sur-reply"). An oral hearing was held and the transcript is in the record. Paper 46.

We have authority to enter this final written decision ("Decision") under 35 U.S.C. § 318(a). The standard for review is set forth in 35 U.S.C. § 316(e), which provides that "the petitioner shall have the burden of proving a proposition of unpatentability by a preponderance of the evidence." For the reasons that follow, we determine that Petitioner has not met this burden of establishing unpatentability of any of the challenged claims by a preponderance of the evidence.

B. Real Parties in Interest

Petitioner identifies Microsoft Corporation, Acer America Corporation, Acer Inc., ASUS Computer International, ASUSTeK Computer, Inc., Dell (Chengdu) Company Limited, Dell Products LP, and Dell Technologies Inc. as real parties in interest. Pet. xiii. Patent Owner identifies X1 Discovery, Inc. as the real party in interest. Paper 3, 2.

C. Related Matters

The parties identify *X1 Discovery, Inc. v. Microsoft Corp.*, No. 8:23-cv-02415 (C.D. Cal.) ("*X1 v. Microsoft* litigation") as involving the '035 patent and related U.S. Patent Nos. 9,633,139 ("the '139 patent") and

10,552,490 ("the '490 patent"). Pet. xiii; Paper 3, 2 (Patent Owner's Mandatory Notices); Paper 10, 2 (Patent Owner's Updated Mandatory Notices). The parties indicate that Petitioner is concurrently challenging the '139 and '490 patents in IPR2025-00255 and IPR2025-00254, respectively. Pet. xiv; Paper 3, 3.

Patent Owner's Updated Mandatory Notices further identify an additional patent, U.S. Patent No. 8,498,977 ("the '977 patent"), as being asserted against Petitioner in the *X1 v. Microsoft* litigation. Paper 10, 2. Patent Owner reports that the '977 patent and still another patent, U.S. Patent No. 8,856,093 ("the '093 patent"), also are asserted against Acer Inc. and Acer America Corporation; ASUSTeK Computer Inc. and ASUS Computer International; and Dell Technologies Inc. and Dell Products L.P. in *Certain Computing Devices Utilizing Indexed Search Systems & Components Thereof*, Inv. No. 337-TA-1389 (ITC), and in the following district court proceedings: *X1 Discovery, Inc. v. Acer Inc.*, No. 2:23-cv-10674 (C.D. Cal.); *X1 Discovery, Inc. v. ASUSTeK Computer Inc.*, No. 2:23-cv10672 (C.D. Cal.); and *X1 Discovery, Inc. v. Dell Technologies Inc.*, No. 2:24-cv-04109 (C.D. Cal.). *Id.* at 2–3.

D. The '035 patent

The '035 patent, titled "Methods and Systems for Search Indexing," issued May 6, 2008, from an application filed September 3, 2003, and claims the benefit of five provisional applications filed between September 23, 2002, and June 13, 2003. Ex. 1001, codes (22), (45), (54), (60). The '035 patent "relates generally to data processing, and in particular to systems and methods for locating data." *Id.* at 1:29–30.

The '035 patent identifies various drawbacks of conventional search application programs. Ex. 1001, 1:32–2:5. For example, according to the

’035 patent, such applications were “often . . . slow and cumbersome to use,” and many “require[d] the user to type in a search term, click on a search button, and review the results,” in some cases repeatedly. *Id.* at 1:32–39, 1:51–2:5. In addition, according to the ’035 patent, many search application programs only provided for limited search filtering, and some did not include search features directed to specific types of search targets, necessitating the use of separate tools to search for email, Web pages, and files. *Id.* at 1:39–50. In view of such drawbacks, the ’035 patent in one exemplary embodiment “provides for incremental or reactive searching of a variety of search targets,” including “files, emails, email attachments, Web pages, specific databases, and/or the like.” *Id.* at 2:9–13. Because the search is performed incrementally, according to the ’035 patent, the search results are “provided or narrowed substantially immediately after each character in a search string is entered” by the user, and, “[t]hus, the user is provided with substantially immediate feedback as the search string is being entered, and so can quickly decide on the desirability of entering additional search characters, entering a new search string, or deleting one or more search string characters.” *Id.* at 2:13–21. The ’035 patent also describes, for example, an embodiment that includes receiving at least two search strings as they “are being entered into the same search field,” and “incrementally locating at least a first document that has a[t] least a first word that begins with the first search string and a second word that begins with the second search string.” *Id.* at 2:30–36.

E. Challenged Claims

Petitioner challenges claims 1–10 of the ’035 patent. Of the challenged claims, claims 1 and 10 are independent. Claim 1, reproduced

below with Petitioner's bracketed identifiers and formatting added (*see* Ex. 1040), is representative.

> [1.1] A method of performing a search comprising:
>
> [1.2] receiving a first string in a first search field;
>
> [1.3] in response to receiving the first string, incrementally locating a first group of documents that has at least a first word that begins with the first string;
>
> [1.4] receiving a second string in the first search field, wherein the first and second strings are separated by a string separator character in the first search field;
>
> [1.5] in response to receiving the second string, incrementally locating a second group of documents that has at least a second word that begins with the second string and identifying one or more documents that are included in each of the first and second groups of documents; and
>
> [1.6] displaying an indication of one or more of the identified one or more documents that are included in each of the first and second groups of documents.

Ex. 1001, 29:39–54.

F. Asserted Grounds

Petitioner asserts the following grounds of unpatentability (*see* Pet. 2):

Ground	Claims Challenged	35 U.S.C. §[1]	Reference(s)/Basis
1	1–10	103(a)	Wilcox,[2] Londergan,[3,4] Raskin,[5] Wu[6]
2	1–10	103(a)	Schwartz,[7] Sebastian,[8] Barber,[9, 10] True,[11] Baeza-Yates[12]

[1] The Leahy-Smith America Invents Act ("AIA") included revisions to 35 U.S.C. § 103 that became effective on March 16, 2013. Because the application from which the '035 patent issued was filed before this date, the pre-AIA version of § 103 applies. *See* Ex. 1001, code (22).

[2] Adam A. Wilcox, *PC Learning Labs Teaches Lotus Notes 3.0* (1993) (Ex. 1059).

[3] Stephen Londergan & Pat Freeland, *Lotus Notes R5 for Dummies* (1999) (Ex. 1061).

[4] Petitioner refers to Wilcox and Londergan collectively as "The Lotus Notes references." *See, e.g.*, Pet. 2.

[5] Jef Raskin, *The Humane Interface* (2000) (Ex. 1021).

[6] Wu, US 5,991,756, issued November 23, 1999 (Ex. 1019).

[7] Steve Schwartz, *Visual QuickStart Guide: Entourage 2001 for Macintosh* (2001) (Ex. 1004).

[8] Bonita Sebastian et al., *Microsoft Office 2001: A Hands-on Guide for Macintosh* 171–88 (2001) (Ex. 1005).

[9] Nan Barber & David Reynolds, *Office 2001 for Macintosh: The Missing Manual* (2001) (Ex. 1006).

[10] Petitioner refers to Schwartz, Sebastian, and Barber collectively as "The Entourage references." *See, e.g.*, Pet. 2.

[11] True et al., US 6,112,172, issued August 29, 2000 (Ex. 1009).

[12] Ricardo Baeza-Yates & Berthier Ribeiro-Neto, *Modern Information Retrieval* (1999) (Ex. 1023).

Petitioner and Patent Owner also rely on declaration testimony from Dr. Loren G. Terveen and Mr. Martin P. Haeberli. *See* Declaration of Loren G. Terveen, Ph.D. (Ex. 1003), Supplemental Declaration of Loren G. Terveen, PhD. (Ex. 1100), Declaration of Martin P. Haeberli (Ex. 2001), Supplemental Declaration of Martin P. Haeberli (Ex. 2020).

II. ANALYSIS

A. Legal Standards

"In an IPR, the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable." *Harmonic Inc. v. Avid Tech., Inc*., 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring *inter partes* review petitions to identify "with particularity . . . the evidence that supports the grounds for the challenge to each claim")); *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc*., 800 F.3d 1375, 1378 (Fed. Cir. 2015) (discussing the burden of proof in *inter partes* review).

A claim is unpatentable under the pre-America Invents Act ("AIA") version of 35 U.S.C. § 103,[13] if "the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious . . . to a person having ordinary skill in the art to which said subject matter pertains." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of

[13] The AIA made revisions to § 103 that apply to inventions having an effective filing date on or after March 16, 2013. *See* Pub. L. No. 112-29, § 3(n)(1), 125 Stat. 293 (2011). The pre-AIA version of § 103 applies to the '035 patent, which issued from an application filed on September 3, 2003. *See* Ex. 1001, code (22).

the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) when in evidence, objective indicia of non-obviousness.[14] *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17–18 (1966).

When evaluating a combination of teachings, we must also "determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR*, 550 U.S. at 418 (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)). It is not enough for a petitioner to show merely that the prior art includes separate references covering each separate limitation in a challenged claim. *Unigene Labs., Inc. v. Apotex, Inc*., 655 F.3d 1352, 1360 (Fed. Cir. 2011). Demonstrating obviousness requires a showing that a person of ordinary skill at the time of the invention "would have selected and combined those prior art elements in the normal course of research and development to yield the claimed invention." *Id*.

Moreover, in determining the differences between the prior art and the claims, the question under 35 U.S.C. § 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. *Litton Indus. Prods., Inc. v. Solid State Sys. Corp.*, 755 F.2d 158, 164 (Fed. Cir. 1985) ("It is elementary that the claimed invention must be considered as a whole in deciding the question of obviousness."); *see also Stratoflex, Inc. v. Aeroquip Corp*., 713 F.2d 1530, 1537 (Fed. Cir. 1983) ("[T]he question under 35 U.S.C. § 103 is not whether the differences themselves would have been obvious. Consideration of differences, like each of the findings set forth in *Graham*, is but an aid in

[14] Neither party has directed us evidence that allegedly pertains to objective indicia of non-obviousness.

reaching the ultimate determination of whether the claimed invention as a whole would have been obvious."). As a factfinder, we also must be aware "of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning." *KSR*, 550 U.S. at 421.

B. Level of Ordinary Skill in the Art

Determining whether an invention would have been obvious under 35 U.S.C. § 103 requires resolving the level of ordinary skill in the pertinent art at the time the invention was made. *Graham*, 383 U.S. at 17. The level of skill in the art is "a prism or lens" through which we view the prior art and the claimed invention. *Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001).

Factors pertinent to a determination of the level of ordinary skill in the art include: (1) educational level of the inventor; (2) type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) educational level of workers active in the field. *Env't Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 696–697 (Fed. Cir. 1983) (citing *Orthopedic Equip. Co. v. All Orthopedic Appliances, Inc.*, 707 F.2d 1376, 1381–82 (Fed. Cir. 1983)). Not all such factors may be present in every case, and one or more of these or other factors may predominate in a particular case. *Id.* Moreover, "these factors are not exhaustive but are merely a guide to determining the level of ordinary skill in the art." *Daiichi Sankyo Co. v. Apotex, Inc.*, 501 F.3d 1254, 1256 (Fed. Cir. 2007). In determining a level of ordinary skill, we also may look to the prior art, which may reflect an appropriate skill level. *Okajima*, 261 F.3d at 1355.

Petitioner contends that a person of ordinary skill in the art "for purposes of the '035 patent in September 2002 ('POSITA') would have had

a bachelor's degree in computer science, or equivalent, with two or more years of experience developing and implementing search engines and/or search interfaces." Pet. 8. Petitioner further contends, "[m]ore education, e.g., a Master's or Ph.D., would compensate for less work experience and vice versa." *Id.* (citing Ex. 1003 ¶ 30). Still further, Petitioner argues, a person of ordinary skill in the art "would have been familiar with common, then-existing search tool functionality," including "incremental (i.e., character-by-character) searching in which search results are displayed as a user types their query"; "locating documents containing two or more search strings, such as in response to a user including a Boolean AND operator in their search query"; "highlighting a searched string in the search results so that a user can easily and quickly see where the searched string appears in the located document(s)"; and "email-specific search functionality, such as the ability to search within email-specific headers (e.g., To, From, Subject, etc.)." *Id.* at 8–11 (citing Ex. 1003 ¶¶ 31–59; Ex. 1004; Exs. 1006–1041; Ex. 1043; Ex. 1044; Ex. 1052; Ex. 1053; Ex. 1057; Ex. 1059; Ex. 1061).

Patent Owner does not comment on Petitioner's formulation or offer any competing formulation in its Response. Mr. Haeberli, however, testifies as follows in his Declaration:

> 29. In my opinion, the POSITA at the time of the alleged invention of the Asserted Patents would have had a (1) a Bachelor's degree in a pertinent discipline, such as computer science, electric engineering, computer engineering, or a related field and (2) have 2–4 years of experience in working with computing devices and have an understanding of indexed search systems utilized by such computing devices.
>
> 30. I am aware that Petitioner has provided its own definition . . . .

> 31. I believe Petitioner's definition is not meaningfully different from my definition of a POSITA. My opinions would not change regardless of which definition of POSITA is applied.

Ex. 2001 ¶¶ 29–31.

In our Decision on Institution, we adopted Mr. Haeberli's formulation, finding it to be consistent with the evidence of record at this stage of the proceeding, including the asserted prior art and the '035 patent. D.I. 8. Following a review of the full record, we agree with Mr. Haeberli's statement that Petitioner's definition of one of ordinary skill in the art is "not meaningfully different" from Mr. Haeberli's for the purposes of this proceeding. We continue to view Mr. Haeberli's proposed definition as consistent with the '035 patent and the asserted prior art, and we apply it in our analysis below. However, our analysis of the issues below would have been the same had we instead adopted Petitioner's proposed definition.

C. Claim Construction

We interpret claims in the same manner used "in a civil action under 35 U.S.C. § 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent." 37 C.F.R. § 42.100(b). When applying that standard, we interpret the claim language as it would have been understood by one of ordinary skill in the art in light of the specification. *Wasica Fin. GmbH v. Cont'l Auto. Sys., Inc.*, 853 F.3d 1272, 1279–80 (Fed. Cir. 2017). Thus, we give claim terms their ordinary and customary meaning as understood by an ordinarily skilled artisan. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc). Only terms that are in controversy need to be construed, and then only to the extent necessary to resolve the controversy. *Nidec*

Motor Corp. v. Zhongshan Broad Ocean Motor Co., 868 F.3d 1013, 1017 (Fed. Cir. 2017).

In our Institution Decision, we construed the claim term "string separator character" to mean "a single-character string separator," and invited the parties to raise any issues they had with this construction in future briefing. D.I. 10. Patent Owner subsequently indicated that it agreed with this construction. *See* PO Resp. 8. Petitioner did not raise any objections to this construction (*see generally* Pet. Reply). Accordingly, and in the absence of any dispute, we construe "string separator character" to mean "a single-character string separator."

Neither Petitioner nor Patent Owner ask us to adopt explicit constructions of any other claim terms. *See* Pet. 14, PO Resp. 8–9; Pet. Reply 1–23; PO Sur-reply 2–7. We also find that the issues in dispute can be resolved without construing any other claim terms. Accordingly, we do not adopt explicit constructions of any other claim terms. *See Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019) ("The Board is required to construe 'only those terms … that are in controversy, and only to the extent necessary to resolve the controversy.'") (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)).

D. Ground 1: Alleged Obviousness of Claims 1–10 over Wilcox, Londergan, Raskin, and Wu

1. Overview of the Prior Art

a) The Lotus Notes References

Wilcox (Ex. 1059) and Londergan (Ex. 1061), which Petitioner refers to collectively as the "Lotus Notes references," are publications describing Versions 3.0 and R5, respectively, of the Lotus Notes software package. *See* Ex. 1059, (a), xvi; Ex. 1061, (i). Patent Owner does not dispute

Petitioner's contention that these publications qualify as prior art under 35 U.S.C. Sections 102(a) and 102(b) because they were published in 1993 and 1999, respectively, and were publicly available more than one year before the earliest possible priority date of the challenged patent. *See* Pet. 16.

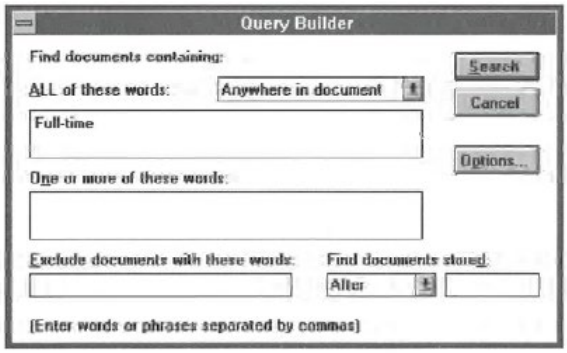
Wilcox and Londergan explain that Lotus Notes is a so-called "groupware" software product that is designed to allow groups of users to communicate and share information. Ex. 1059, 2; Ex. 1061, 6. Both versions of Lotus Notes store documents in a central database that resides on a network, rather than on the computers of individual users. Ex. 1059, 2–3; Ex. 1061, 12. Both versions of Lotus Notes contain an electronic mail software. Ex. 1059, 5, Ex. 1061, 11. These electronic mail messages are also stored in a database on a central server. Ex. 1059, 5; Ex. 1061, 11–12. Lotus Notes indexes email messages and other types of documents, and allows users to conduct searches. *See* Ex. 1059, 160–65; Ex. 1061, 139–142.

(1) Wilcox

The first Lotus Notes reference, Wilcox, explains that version 3.0 of Lotus Notes provides two tools for searching through documents (such as emails) in the indexed Lotus Notes database, a Search bar and a Query Builder, plus a Forms feature that allows users to search for text in a particular field in documents created using a specific form. Ex. 1059, 160–161, 167–168. Petitioner's obviousness arguments citing Wilcox primarily rely on the Query Builder, a dialog box that makes it easier for users to construct complex searches. *See, e.g.,* Pet. 27–29 (citing Ex. 1059, 163); Pet. Reply 17–19 (focusing on the Wilcox Query Builder).

Wilcox Figure 5.16, reproduced below, depicts the Query Builder dialog box.

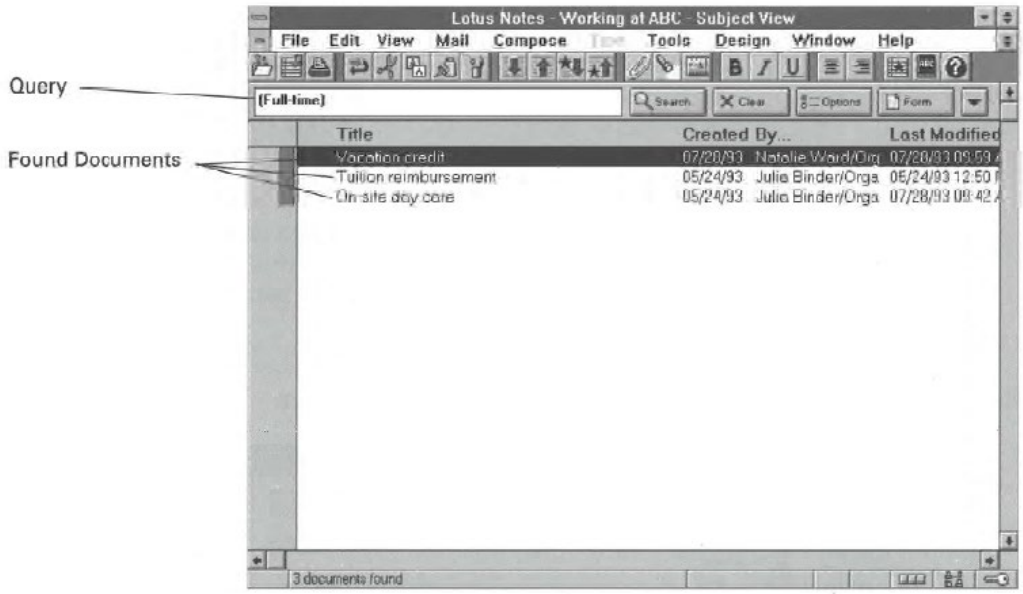
Figure 5.16 Using the Query Builder dialog box



Ex. 1059, 163. As shown above, the Query Builder dialog box includes a drop-down menu that allows a user to choose whether to conduct a search for the selected search terms "anywhere in document," or "near each other" (in which case search results are ranked by how close search terms appear to one another). *Id*. at 162–63. The Query Builder also contains an "All of these words" text box that can be used to search for documents that contain all of two or more user-specified words, and a "One or more of these words" box that can be used to search for documents that contain one or more of the specified words. *Id*. The Query Builder additionally contains a textbox that permits a user to exclude certain words from a search, and a drop-down menu and text box that a user can use to search for documents stored before or after a certain date. *Id*. The user is instructed that when entering multiple words in a textbox, the user should "[e]nter words or phrases separated by commas." *Id*. Once a query has been entered, the user must click on the "Search" button to view in a separate window a list of documents containing the search terms. *Id*.

Figure 5.17 of Wilcox, reproduced below, depicts the results of a query constructed with the Query Builder dialog box.

Figure 5.17 The results of a query



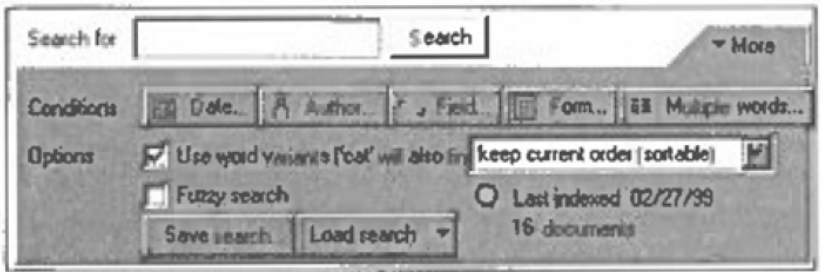
Ex. 1059, 164. Figure 5.17 depicts a search result window that displays the user's query in a search bar at the top of the window, with a list of found documents underneath. The list displays document titles, create dates, authors, and last modified dates, but does not show the documents' contents. *Id.*, Fig. 5.17. To view search results in a particular document, the user must manually open the document to view the found text, which is indicated by a red box. *Id.* at 164.

(2) Londergan

The second Lotus Notes reference, Londergan, describes the search functionality of version R5 of Lotus Notes. Ex. 1061, (i), 139–48. This 1999 version of Lotus Notes (R5) eliminated the Query Builder feature in favor of a different feature with different functionality called "Search Builder." Ex. 2001 ¶ 82. Search Builder provides multiple fields, each of which can only be used for a single search string. Like Version 3.0, Lotus

Notes R5 includes a Search bar that can be used to search for words and phrases. *See id*. at 140. The Search bar in Version R5, however, includes a "More" drop-down button that displays the "Search Builder" dialog box. *See* Ex. 1061, 140–142.

A figure from Londergan depicting the Lotus Notes R5 Search Builder dialog box is reproduced below:



Ex. 1061, 141. As shown above, the Search Builder dialog box contains several buttons and fields that permit a user to specify various types of search conditions. *Id*. at 141–45. For example, the Search Builder includes an "Author" condition button that can be used to limit a search query to documents written by a particular author, and a "Date" condition button that can be used to limit a search to documents created before, after, or between certain dates. *Id*. at 141–42. There is also a "Field" condition button that opens a separate dialog box that permits searches for text contained in particular fields of the Lotus Notes database. *Id*.

The Lotus Notes R5 Search Builder also includes a "Multiple words" condition button. *See* Ex. 1061, 142. This button opens a separate dialog box, depicted below, that permits searches for documents that contain "any" or "all" of two or more user-specified words. *Id*.

Id. Unlike in Query Builder, in Search Builder depicted above, each search term is entered into a separate textbox. *Id.* As shown above, a user must click on an "OK" button to view the search results, which then appear in a separate window. *Id.*

b) Raskin

Raskin is a textbook describing user interface design principles for interactive systems. Ex. 1021, (i), (xi)–(xii). Patent Owner does not dispute Petitioner's contention that Raskin qualifies as prior art under 35 U.S.C. Sections 102(a) and 102(b) because it was published in 2000 and was publicly available more than one year before the earliest possible priority date of the challenged patent. *See* Pet. 20–21.

Portions of Raskin discuss interfaces for conducting string searches and "Find" searches. *See* Ex. 1021, 124-32. Raskin teaches that such search interfaces typically use either or both of two specific search strategies: delimited searches and incremental searches. *Id.* at 125.

In the first and more common type of search strategy, delimited searching, the user would typically enter text in a dialog box field. Ex.

1021, 125. The user would then initiate a search by entering a delimiting character (e.g., the Return key), or clicking a button (e.g., OK, Search, Find). *Id*. This would cause the system to search for instances of the relevant text string and designate the first matching instance of the string with a cursor placed immediately at the end of the selection. *Id*.

In the less common strategy described in Raskin, incremental searching, the user might summon a dialog box and enter a search term. Ex. 1021, 125. In an exemplary incremental search, the computer would immediately search for the first character entered by the user and designate the first instance of this character in the searched document with a cursor. *Id*. After the second character is entered, the computer would begin searching for the first instance of those two characters and then highlight the first such instance with a cursor. *Id*. The computer would then repeat this pattern each time an additional character was added to the search term. *Id*.

Raskin teaches that incremental searches have a number of advantages over delimited searching. Ex. 1021, 126. For example, incremental searches waste less of a user's time, make it easy for users to identify and correct search errors, and provide constant feedback to the user. *Id*. Notwithstanding these advantages, Raskin teaches that incremental searching is less common than delimited searching because interface-building tools (e.g., JavaScript and Visual BASIC) make it "difficult or impossible" for developers to implement incremental searches. *Id*.

c) Wu

Wu is a U.S. Patent titled "Information Retrieval from Hierarchical Compound Documents." Ex. 1019 code (54). Patent Owner does not dispute Petitioner's contention that Wu qualifies as prior art under 35 U.S.C. Sections 102(a) and 102(b) because it was published in 1999 and publicly

accessible more than one year before the earliest possible priority date of the challenged patent. *See* Pet. 22.

Wu generally relates to an apparatus for searching for selected documents in a document repository that contains a large number of documents. Ex. 1019, 3:24–27. Wu discloses a search engine that utilizes a document repository and word index to perform an intersection search. *Id*. at 6:51–52. Petitioner's contentions regarding Wu are not disputed, and this reference receives little attention from the parties. Thus, no further discussion of Wu is necessary.

2. The Asserted Combination

Petitioner asserts that claims 1–10 are unpatentable as obvious over the combination of Wilcox, Londergan, Raskin, and Wu. Pet. 6–39. According to Petitioner, Wilcox and Londergan "teach every element of the challenged claims" except for "(1) locating the documents incrementally (i.e., character-by-character) and (2) doing so using an intersection approach (i.e., searching each string separately and then identifying the documents that appear in both sets of search results)." *Id.* at 15. Petitioner contends that Raskin expressly discloses incremental searching and that Wu expressly discloses using an intersection approach. *Id.* (citing Ex. 1003 ¶ 54).

Petitioner argues that a person of ordinary skill in the art would have been motivated to combine Wilcox and Londergan, which describe two different versions of Lotus Notes but contain "complementary teachings" on Lotus Notes systems, in order to "obtain a more complete and comprehensive understanding of the Lotus Notes system, its features, and its capabilities." Pet. 24; *see also* Ex. 1059; Ex. 1061.

According to Petitioner, one of ordinary skill would have been further motivated to "perform Lotus Notes' multi-string search incrementally for the

reasons stated by Raskin: speed, effectiveness, and ease of use." Pet. 25; Ex. 1021, 125–27. Petitioner argues that "Raskin specifically points to the type of search taught by Lotus Notes—a multi-string Boolean search—as a search that benefits from being performed incrementally." Pet. 25 (citing Ex. 1021, 127; Ex. 1003 ¶ 75). Petitioner contends that one of ordinary skill additionally would have been motivated to use Wu's intersection approach "at least because it 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. 25; Ex. 1003 ¶ 76. According to Petitioner, one of ordinary skill "would have reasonably expected to succeed in combining these teachings from Raskin and Wu with Lotus Notes' teachings because it would have required only simple and straightforward software modifications." Pet. 25–26; Ex. 1003 ¶ 77.

Patent Owner argues, *inter alia*, that Petitioner has not shown a reason to combine Lotus Notes with Raskin with a reasonable expectation of success. PO Resp. 17–28. In particular, Patent Owner argues that, as a technical matter, Raskin's incremental search is a different kind of search compared to the searches in Lotus Notes, and that Petitioner has not sufficiently addressed these differences. *See, e.g., id*. at 26–27. Petitioner replies that it is only taking the idea of incremental searching from Raskin, and that Patent Owner is requiring a bodily incorporation of Raskin's search into Lotus Notes' search. Pet. Reply 8–9, 15–16. Patent Owner then reiterates its points about the differences between the two types of searching in Lotus Notes and Raskin. PO Sur-reply 10–11. Patent Owner also disputes Petitioner's "bodily incorporation" characterization, pointing out that Patent Owner is not discussing bodily incorporation, but rather an incompatibility between two different types of systems. *Id*. at 14–15. We

have read and considered the arguments in the briefs, and the evidence cited therein.

3. Analysis

The Lotus Notes search features cited by Petitioner (Wilcox's Query Builder and Search bar, and Londergan's Search bar and Search Builder) permit a user to conduct searches by *entering or building* the search/query and then *subsequently* executing it (by, for example, pressing a "Search" button), and the purpose of the search features is to locate matching *documents*. *See, e.g.*, Ex. 1059, 163 (noting Figure 5.16, depicting a dialog box stating "Find documents containing" various text strings), 164 (noting Figure 5.17, showing the results of the search as a list of documents); Ex. 1061, 140 (stating that, "when you ask Notes to search for a word, Notes finds all the documents for you"); Ex. 2001 ¶ 52.

For example, in Wilcox, the Query Builder has a number of text boxes that a user fills out in order to build the search, which is then executed with all of the text boxes considered at once. Ex. 1059, 162–164; PO Resp. 19 (arguing that "[a]ll of the search tools in Lotus Notes references are delimited searches; requiring a user to click 'Search' to perform the search"). The Query Builder uses the information gathered from the user to "translate[] your request into Notes query language." Ex. 1059, 162; *see also id.* at 167 (by using the Form mode in the search bar, the user can also create queries on text in a particular field). That is, Lotus Notes takes all of the information entered by the user to build up, and then submit, a query for the user. *Id.*; *see also* Ex. 2020 ¶ 34 (Patent Owner's declarant, Mr. Haeberli, testifying that, "[i]n a delimited search interface, the user must complete their query entry and then explicitly click a 'Search' button of some kind to initiate the search operation and receive results").

The newer version of Lotus Notes discussed in Londergan uses a search bar, but it has largely the same configuration – searching over multiple fields in a single search. Ex. 1061, 140–142; *see also id.* at 143–144 (noting the query form, like in Wilcox, exists behind the search bar). In summary, in the Lotus Notes search features described in the Petition, the user enters the information needed to construct a query before the search is executed, and the search results are a list of matching documents that match the query.

The types of delimited searches that Petitioner cites from Wilcox and Londergan are very different from searching for a word within an open document and then finding and moving to that word within the document. *See* Ex. 2020 ¶ 15 (Mr. Haeberli contrasting scanning within an open document with a delimited search performed across multiple documents), ¶¶ 34–37 (further contrasting delimited and incremental searches). For example, Londergan says that "you may simply want to find some text in the document you're reading," and in such case, *instead* of using the query builder, Londergan instructs the user to use the simple "Find/Replace" function. Ex. 1061, 148. In other words, the Lotus Notes references explicitly recognize the difference between building a query to find matching documents and simply looking for a word or words within a document, and Lotus Notes provides separate functionality to search for a word within a document. *See* Ex. 2020 ¶ 31 (Mr. Haeberli testifying that the various search features in Lotus Notes are intentionally different because "each reflect different assumptions about how a user selects search terms, how results are filtered, what subset of documents is subject to the search, and how precisely a delimited search is initiated").

We agree with Patent Owner that a fundamental problem with Petitioner's combination is that Petitioner proposes to take the *idea* of incremental searching from Raskin, but divorces the idea from Raskin's actual teachings regarding implementation of incremental searching. *See* PO Resp. 20 (arguing, "a POSITA must consider all the teachings of a reference, and not pick out general 'concepts' disclosed in the art without considering the context in which they are used"); *see also id.* at 17–28 (explaining this argument in detail); Ex. 2020 ¶¶ 39–62 (Mr. Haeberli's testimony explaining why Petitioner's approach is "technically problematic"). Raskin's incremental search is tied to finding and moving to character strings within an open document. Ex. 1021, 125. For example, if a user were to type a string that was found in the text, "the instance [of that string] is selected and the cursor placed immediately after the end of the selection." *Id.* The cursor can move about the text as characters of the search string are added or deleted. *Id.* at 125–126.

Raskin continues to discuss within-document text searches even when it addresses searching through multiple documents. *Id.* at 132 ("After the local document has been completely searched the search then proceeds through the following documents until the end of the folder is reached, when the search continues from the beginning of the first document in the folder until the current, already searched document is reached. After cycling through the folder, the next larger domain is treated similarly, and so forth."). Raskin even distinguishes its text-finding search from the type of document-finding Lotus Notes searches discussed by Petitioner, stating that "[i]t is usually easy to tell in what domain you are searching, because you see the results in context; *what you see is not just a list of file names, as with many present search systems*." *Id.* (emphasis added); *see also* PO Sur-reply

17 (citing the same page and arguing that "Raskin recommends doing the *opposite* of what is claimed").

This context gives credence to Patent Owner's argument that "the search functionalities disclosed in the Lotus Notes References and Raskin are materially different in such a way as to render combining them anything but simple and straightforward." *See* PO Resp. 26–27. Indeed, Petitioner has given little or no explanation of how one of ordinary skill in the art allegedly would have taken the "idea of incremental search" and applied it to Lotus Notes. *See, e.g.*, Ex. 2021, 116:20–117:17 (Petitioner's declarant, Dr. Terveen, testifying that "Raskin is not explicitly talking about searching across a set of documents, but the teaching that I relied on in general from Raskin is the idea of incremental searching," and "I did not have to talk about implementation level details or incorporating a particular implementation. That wasn't necessary for my opinion."). Raskin simply does not teach a mode of incremental searching that applies to the query-style, document-finding searching of the type Petitioner cites from Lotus Notes. *See generally* Ex. 1021; *see also* Ex. 2020 ¶¶ 34–35 (Mr. Haeberli contrasting the Lotus Notes searches cited by Petitioner with Raskin's searches). Raskin even states that it is "difficult or impossible to implement incremental searches" using interface-building tools. Ex. 1021, 126; Ex. 2020 ¶ 60 (Mr. Haeberli's testimony that this passage of Raskin "directly undermines Dr. Terveen's suggestion that a POSITA would have been motivated to implement Raskin's incremental search approach in the Lotus Notes system").

Although Petitioner calls Raskin's statement a "red herring," we do not agree. *See* Pet. Reply 15. Even if a person of ordinary skill in the art *could* implement an incremental search using C or C++ instead of the

interface-building tools described in Raskin, the scope of our inquiry is not mere technical capability. *See* Pet. Reply 15–16 (asserting that C or C++ "allowed an ordinary programmer to build a [user interface] that supported incremental searching"); Ex. 1098, 24:29–35:6 (Mr. Haeberli agreeing that a person of ordinary skill in the art could have made a "dynamic text box" using C++); *see also KSR*, 550 U.S. at 418 ("a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art").

Petitioner does not offer evidence of incremental searching being implemented in a context similar to Lotus Notes, or otherwise explain how a person of ordinary skill in the art allegedly would take the apples of Raskin's within-a-document text search and apply it to the oranges of the Lotus Notes document-locating searches relied on by Petitioner. Ex. 2020 ¶ 65 (Mr. Haeberli testifying that Petitioner's declarant offers an opinion "without grappling with the implementation hurdles that Raskin itself identified"); *see also id.* ¶ 17 ("I am not aware of any prior art before 2002 which discloses incrementally searching for and locating documents. None of Dr. Terveen's cited references disclose such a search system."), ¶¶ 18–25 (persuasively explaining how the background references cited by Petitioner do not teach incremental document search). Petitioner effectively takes the disembodied idea of a database-wide incremental search—a type of search not disclosed in or taught by any reference—and proposes to add it to the Lotus Notes references (themselves not one but two separate and distinct disclosures), waving away any potential issues relating to how or why a skilled artisan would have done so in a manner that would have yielded the claimed invention. *See* PO Resp. 17 (arguing that the combination is faulty because "the only stated motivation to combine the teachings of the Lotus Notes

references says nothing about why a POSITA would be motivated to combine the disparate elements of the four search tools in Lotus Notes in a combination that aligns with the claimed invention"); Ex. 2020 ¶ 31 (Mr. Haeberli testifying that the various features in Lotus Notes are separate because "each reflect different assumptions about how a user selects search terms, how results are filtered, what subset of documents is subject to the search, and how precisely a delimited search is initiated").

Petitioner has not persuasively shown that Raskin teaches an incremental search of the type required by both independent claims, much less that one of ordinary skill in the art would have realized that this type of search could improve the Lotus Notes search functionalities described in the Petition. Raskin discloses and discusses a fundamentally different type of incremental search. Even assuming, *arguendo*, that we were persuaded that one of ordinary skill would have had reason to take the disembodied idea of Raskin's incremental search and combine it with Lotus Notes, we would still be left with the fundamental question of how Lotus Notes would have needed to be modified to implement an incremental search, given that the Petitioner-cited Lotus Notes search functionalities are premised on first *building* a search, which includes the ability to complete several fields, before executing the built search to locate documents. *See, e.g.*, Ex. 1059, 163 (Fig. 5.16); Ex. 1061, 141. Based on the entirety of the record before us, we agree with Patent Owner that the Lotus Notes style of searching does not appear compatible with an incremental search. PO Sur-reply 10–11; Ex. 2020 ¶¶ 78–80 (contrasting the two types of search), ¶¶ 82–83 (highlighting the open questions of what Petitioner's combination actually would be); *see also id.* at ¶¶ 63–71 (detailed explanation of the problems with the combination).

Based on the entirety of the record before us, we determine that Petitioner has not shown, by a preponderance of the evidence, that a person of ordinary skill in the art would have considered it obvious to add (or would have succeeded in adding) incremental searching to the Lotus Notes search functionalities relied on by Petitioner in the way that Petitioner alleges would have yielded the invention recited in claims 1 or 10. *See* Pet. 26–34, 39–40. Petitioner's contentions for claims 2–9 (which depend from claim 1) rely on the same arguments discussed above and are unpersuasive for the same reasons. *See* Pet. 34–39. Accordingly, Petitioner has not demonstrated by a preponderance of the evidence that claims 1–10 are unpatentable as obvious over Wilcox, Londergan, Raskin, and Wu.

E. Ground 2: Alleged Obviousness of Claims 1–10 over Schwartz, Sebastian, Barber, True, and Baeza-Yates

1. Overview of the Prior Art

a) The Entourage References

Schwartz, Sebastian, and Barber discuss the 2001 version of software called Entourage for Macintosh (hereinafter, "Entourage"). Patent Owner does not contest Petitioner's contention that the Entourage references qualify as prior art under 35 U.S.C. Sections 102(a) and 102(b). *See* Pet. 41.

Because the three Entourage references describe the same version of the same software, our overview below is organized primarily by feature, rather than by reference. Entourage is similar in many ways to modern-day Microsoft Outlook. *See, e.g.*, Ex. 1004, 1 (comparing Entourage to Outlook Express, but with more features such as address book, calendar, tasks, etc., all present in modern-day Outlook). Petitioner primarily relies on two independent features of Entourage—filter and Find. *See* Pet. 41–14; *see also* Ex. 1004, 8–12, 161–164. Find has a further sub-feature called Advanced

Find. *See, e.g.*, Ex. 1004, 11–12, 162–163. The filter and Find features are available in most Entourage components (e.g., email, address book). *Id.* at 8.

(1) Entourage's "filter" Feature

Schwartz states that "[p]erhaps the easiest way to find a particular message, contact record, note, or any other Entourage item is to filter the item list. Ex. 1004, 8. The filter feature is made available "[a]bove the list in many Entourage windows" in the form of "one or two pop-up menus and a text box." *Id.* Figure 1.12 of Schwartz depicts an example, and is reproduced below:

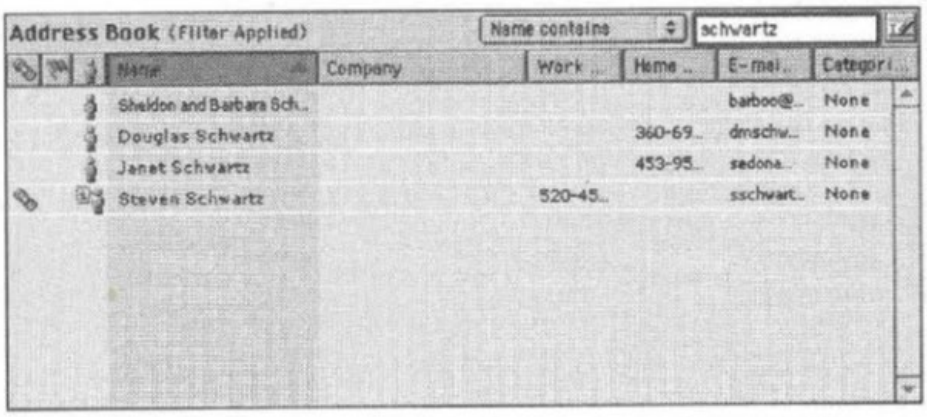


Figure 1.12 As you type the filter text, only matches are shown in the list.

Figure 1.12 of Schwartz, above, illustrates an address book window with a filter applied. Ex. 1004, 9. A button called "Name contains" is selected above an address book window, with the string "schwartz" typed into an adjacent text box. *Id.* The entries displayed in the list all appear to include the string "schwartz" in the "Name" column. *Id.* Notably, when using a filter, "[a]s you type the filter text, only matches are shown in the list," indicating an incremental search capability. *Id.* (caption of Figure 1.12).

The Entourage filter feature as applied to emails includes selection criteria limited to fields like "Subject, From, To, or Category." Ex. 1004, 161. Figure 11.15 of Schwartz is reproduced below:

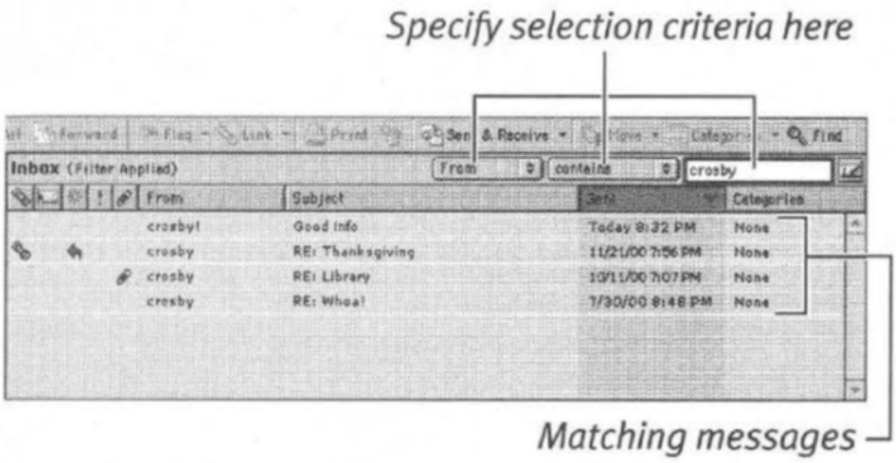


Figure 11.15 To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

Figure 11.15 of Schwartz, above, depicts an inbox with various buttons and a text field above the inbox. The filter interface is labeled with a note that says "Specify selection criteria here," and includes buttons that allow the user to select criteria for filtering. *Id.* at 161. Here, "From" and "contains" are chosen, with "crosby" typed into a text box. *Id.* Figure 11.15 contains a caption that states, "To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list." *Id.* Figure 11.15 of Schwartz, depicting an email filter, is generally similar to Figure 1.12 of Schwartz, depicting an address book filter. *Compare id.* at 161, *with id.* at 9.

Petitioner also cites Sebastian for showing a filter (Pet. 42, citing Ex. 1005, 173), but there is no further information made available at that cite; it merely shows a figure depicting user's inbox that in general appears very

similar to the filter function in Schwartz depicted in Figure 11.15 described above. *Compare* Ex. 1005, 173, *with* Ex. 1004, 161–62.

Barber briefly discusses filters. Ex. 1006, 384. In particular, Barber states that

> You can *filter* messages, notes, tasks, and addresses easily enough (that is, hide all but a certain group) by typing into the little box above the message, task, notes, or address list; for example, you might configure the companion pop-up menus to say, for example, "Subject" and "Contains," and then type *humor* into the box.

Id. This disclosure is consistent with Schwartz, discussed above.

The most notable feature of the filter function is that results are displayed "[a]s you type the filter text," i.e., incremental searching. Ex. 1004, 9. According to Schwartz, "[a]lthough though it is not as sophisticated as the Find and Advanced Find commands discussed later in this section, filtering is the fastest way to perform a simple search." *Id.* Barber adds, "when the needle still eludes you in your rummage through the Entourage haystack, you can use Entourage's slower, more powerful search," a statement that refers to the separate Find command. Ex. 1006, 384.

(2) Entourage's "Find" Feature

Separate from the filter feature is the Find feature. *See, e.g.*, Ex. 1004, 161–163. According to Schwartz, "[t]he Find command is slightly more powerful than filtering. In addition to searching for matching text in an item title, it can search for matches within the body text of all messages or only the current message." *Id.* at 9. Thus, in contrast to the filter, Find allows a user to search for text within the body of a message. *Id.* Find does not use the same interface that the filter uses; instead, Schwartz describes "Find" as

appearing in a dialog box (a separate pop-up window). *Id.* A dialog box requires the user to fill out a form to select various options and then to click a button to begin the search and see the results. *Id.* at 9–10. Figure 1.13 of Schwartz, reproduced below, depicts a Find dialog box.

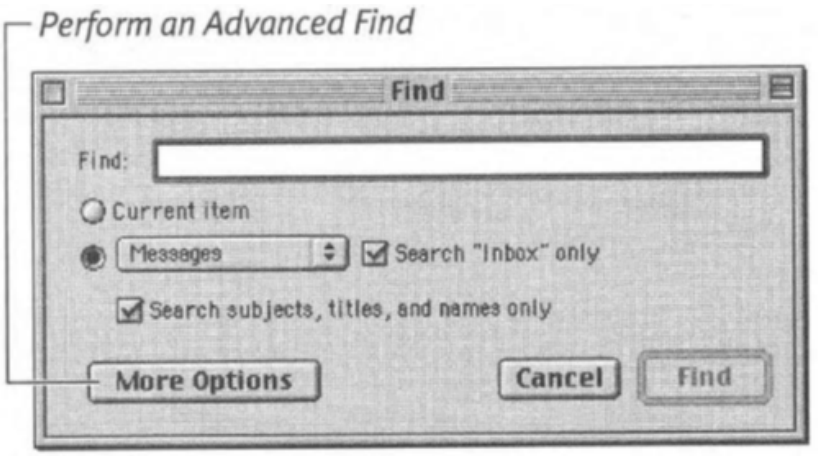


Figure 1.13 You can conduct simple searches by specifying criteria in the Find dialog box.

Id. at 9. In Figure 1.13 of Schwartz, above, the caption states that "You can conduct simple searches by specifying criteria in the Find dialog box." *Id.* Schwartz describes the various steps for using the Find dialog box, such as selecting what is being searched, defining text to find, and clicking on a button to execute the search. *Id.* at 9–10.

Once the Find search executes, "results are displayed in a separate window," as shown in Figure 1.14, reproduced below.

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*.

Ex. 1004, 10. Figure 1.14 of Schwartz, above, displays a separate window with the title "Search Results." *Id.* The caption of Figure 1.14 states that "search results are displayed in a separate window" and that the search results displayed are "for messages in the Inbox that contained the word *FileMaker*." *Id.*; *see also id.* at 162 (Fig. 11.17) (stating in the caption that "[s]earch results are displayed in a new window").

(3) Entourage's Ability to Highlight Text Search Results

Schwartz states that once the search results are displayed in the new window, "[t]o examine any of the matches, click its header and read the message in the preview pane." Ex. 1004, 162 (Fig. 11.17). That is to say, the search results provide a list of hits, but to locate and display particular text within a document, a user has to perform a separate action. *Once a*

message is selected, then a *user* can start a *separate search* for the text within that selected message. *Id.* at 10–11 (describing how "[t]o search the current item for a text string"), 163–164 (describing how "[t]o search the current message for a text string"). In particular, as depicted in the "Find" dialog box of Figure 1.13, reproduced earlier, if a user instead chooses to search within the "Current item," then the search is redirected to only look at the selected item, allowing the user "[t]o search the current item for a text string." *Id.* at 10–11.

Figure 11.19 of Schwartz, reproduced below, shows a search entered by a user who is about to search for a text string within the body of a message:

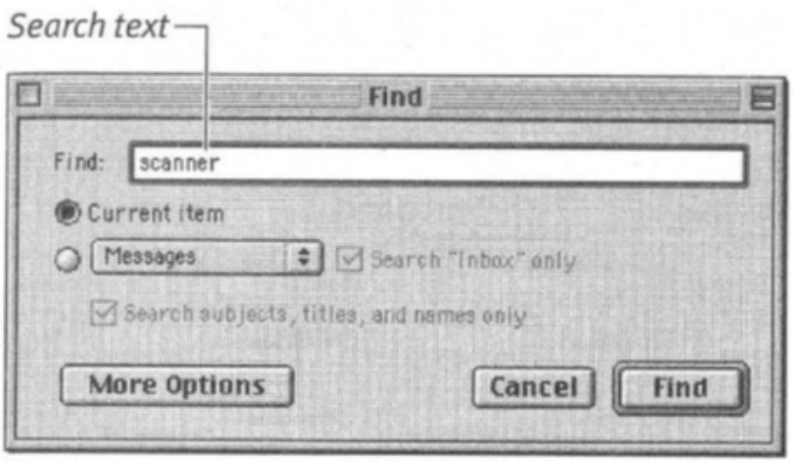


Figure 11.19 To search within the body of the current message for a particular text string, click the Current item radio button in the Find dialog box. All other options are disabled.

Ex. 1004, 163. The caption of Figure 11.19, above, states, "[t]o search within the body of the current message for a particular text string, click the Current item radio button in the Find dialog box. All other options are disabled." *Id.* In the image of Figure 11.19, the "Current item" option is selected and "scanner" is typed into a text box labeled "Find." Other options

normally available in the Find command are colored gray to indicate that they are disabled.

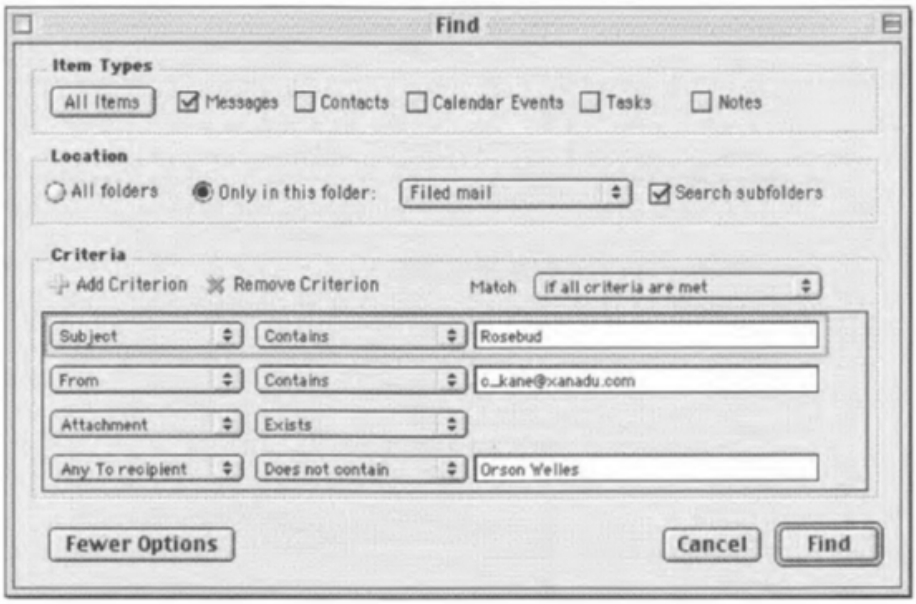
If such a search results in a hit, "[t]he first instance of the text string … is highlighted." Ex. 1004, 11. Figure 1.16, reproduced below, shows the results of such a text search within a selected email item:

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

Id.; *see also id.* at 163 (Figure 11.20, showing the results of the search in Figure 11.19 for "scanner"). The caption of Figure 1.16, above, states that "[w]hen you search for matching text within an item, Entourage highlights the first match, scrolling as necessary to display it." The Figure shows an open window that contains an email message, and in the body of the message the word "Macworld" is highlighted. The highlighted word is annotated as the "First match." Accordingly, the Find feature of Entourage provides two separate modes, one to find an item (potentially containing certain text) and a second to find and display text within a selected item.

(4) Entourage's "Advanced Find" Feature

The Advanced Find feature is accessed through the Find pop-up by clicking "More Options" and provides ways for a user to "perform a more complex search (such as one that uses multiple criteria)." *See, e.g.*, Ex. 1004 at 162–63 (caption of Figure 11.18); *see also id.* at 11–12 (providing another overview of Advanced Find); Ex. 1006, 385–86 (providing yet another overview of Advanced Find). For example, the Advanced Find allows a user to look for items "If Any Criteria Are Met" or "If All Criteria Are Met." Ex. 1004, 163; *see also id.* at 12 (similar). Figure 11-7 of Barber, reproduced below, depicts a dialog box showing Advanced Find:



Ex. 1006, 387. Figure 11-7, above, depicts a search in "Messages," only in the folder "Filed mail" and its subfolders, where all of the following criteria are met: the Subject field contains the word "Rosebud," the From field contains c_kane@xanadu.com, and any "To" recipient does not contain "Orson Welles."

(5) Summary of Entourage References

In sum, the Find and Advanced Find features (which are separate from the filter feature) require a user to open a dialog box to build up a search query by selecting multiple options and then executing a search that takes into account all of the options selected. *See, e.g.*, Ex. 1004, 9–12. Results of the search are provided in another separate window, and if the user wants to find and highlight particular words within the body of a message found in the results, the user has to click on the particular message and perform a separate text search within that message. *See, e.g.*, *id.* at 9–11. The filter feature, on the other hand, allows for searching within various defined categories though a list of items displayed. *See, e.g.*, *id.* at 8–9. Unlike the Find feature, which uses a pop-up window to build a search query and then displays results in yet another window, the filter works in the current window and updates matches as the user types. *See id.*

b) True

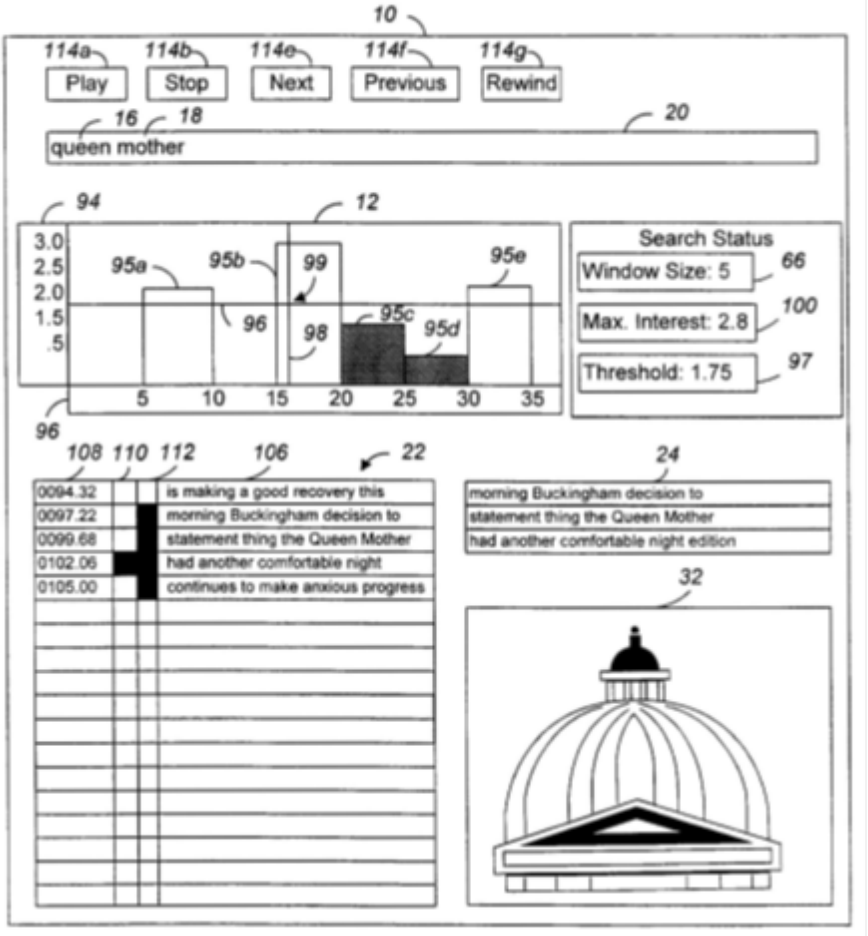
Petitioner contends that True teaches incrementally searching for multiple search strings entered in a single search field. Pet. 47. Patent Owner does not contest Petitioner's contention that True qualifies as prior art under 35 U.S.C. Sections 102(a) and 102(b). *See id*. at 46.

True provides a system for interactive searching wherein the searchable information "may be derived from an audio signal that is linked to a video signal from, e.g., a television broadcast feed." Ex. 1009, 1:31–33. Because the searchable text is derived from audio tracks, the text data "may not always accurately represent the portions of the underlying information … that the user is trying to find." *Id.* at 1:19–22. In particular, the information "is a text file generated by a speech recognizer" and "the words produced by a speech recognizer are sometimes not the words that were

spoken." *Id.* at 1:22–27. Thus, the recognized words are associated with a confidence interval that the spoken word (i.e., sound) is the recognized word (i.e., text). *Id.* at 4:21–37. If, for example, a user is looking for the combination of words "queen mother," then the system will look for "queen" and "mother" separately and combine their confidence intervals to arrive at a confidence in the results of having found "queen mother." *Id.* at 5:41–51; *see also id.* at 5:52–6:22, Fig. 2 (providing details and an implementation example).

True presents the search results graphically, indicating where in the audio track the particular words the user is looking for are likely to be found, as shown in Figure 2, reproduced below:

FIG. 2



Ex. 1009, Fig 2. Figure 2 of True, above, depicts a search in which a user is searching for the words "queen mother" using query window 20. *Id.*; *see also id.* at 3:20–33. Results are displayed in bar chart format in result window 12, with the transcript divided into time segments, and the heights of bars 95a, 95b, 95c, etc. representing True's estimate of the confidence that the searched-for words appear in that portion of the audio transcript. *Id.* at 5:29–39, 6:6–14. Estimates of where the phrase "queen mother" may appear are also listed alongside the searched text information in transcript window 22. *Id.* at 6:38–44. The user can play back the audio using navigation buttons at the top of the window (114a, 114b, etc.) and can watch the corresponding video feed 32. *Id.* at 6:45–61, 7:13–24.

True suggests an incremental search. *Id.* at 5:5–6 ("the user can view the results of a query as the query is being modified"). If a user specifically wants one of the words typed to appear in the results, they must place a plus sign before the query term. *Id.* at 5:26–28. True performs queries on the individual query words, adding together the confidence level of the accuracy of the speech recognition for each of the words appearing in a window of time. *Id.* at 5:35–50. Thus, even where a user searches for two separate words, a high-scoring search result could arise from only one of those words appearing multiple times within a given window of time.

In sum, True takes a search string in a single field that has multiple words in a single field, and looks for those words separately using a confidence interval analysis to allow the user to see if the searched-for words actually appear in the transcript and were in fact likely said.

c) Baeza-Yates

Petitioner cites to Baeza-Yates as teaching independently searching for multiple strings and taking the intersection of those results. Pet. 70.

Patent Owner does not contest Petitioner's contention that Baeza-Yates is prior art under 35 U.S.C. Sections 102(a) and 102(b). Petitioner's contentions regarding Baeza-Yates are uncontested, and this reference receives little attention from the parties. Accordingly, no further discussion of Baeza-Yates is necessary.

2. The Asserted Combination

Petitioner asserts that claims 1–10 are unpatentable as obvious over the combination of Schwartz, Sebastian, Barber, True, and Baeza-Yates. Pet. 40–82. According to Petitioner, the Entourage references (Schwartz, Sebastian, and Barber) "disclose every element of the challenged claims" except "(1) receiving two partial search strings in a single search field; and (2) locating documents containing the two partial search strings by searching each string separately and then identifying the documents that appear in both sets of search results." *Id*. at 40. Petitioner contends that these two missing features are disclosed by True and Baeza-Yates, respectively. *Id*. In particular, Petitioner asserts that the combination of the Entourage references with True and Baeza-Yates "combines the Entourage references' teachings to incrementally locate documents, including emails, containing at least one search string entered in a search field with True's teachings to do so for multiple search strings entered in the search field, and Baeza-Yates teachings to do so [using an intersection search]." *Id*. at 50; Ex. 1003 ¶ 120.

Petitioner argues that one of ordinary skill in the art would have been motivated to combine the teachings of the three Entourage references, and would have reasonably expected to succeed in combining them, "at least because they describe the same version of Entourage (Entourage 2001) for the same computing system (Macintosh)." Pet. 46; Ex. 1003 ¶ 116. One of ordinary skill allegedly "would have been motivated to look to the teachings

of all three references to obtain a more complete and comprehensive understanding of the Entourage system, its features, and its capabilities[.]" Pet. 45.

According to Petitioner, the motivation to combine True's multiple search strings with the teachings of the Entourage references would have been the knowledge that "such a multi-string search would return more relevant results than a single-string query." Pet. 50; Ex. 1003 ¶ 121; *see also* Pet. 54, 56–57, 60 (similar statements on motivation). The motivation to incorporate the intersection search taught in Baeza-Yates would have been that "this was 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. 51; Ex. 1003 ¶ 123. Petitioner asserts that there would have been a reasonable expectation of success in combining the five references "because it would have required only simple and straightforward software modifications." Pet. 51; Ex. 1003 ¶¶ 122, 124.

Patent Owner argues in response that Petitioner is mixing and matching disparate features in Entourage (PO Resp. 37–48), that there is no reason to combine the teachings of the references (*id.* at 55–61), that Entourage does not teach entering two search strings in the same field (*id.* at 62), and that True does not teach searching for documents (*id.*). Petitioner's Reply seeks to clarify its position on True (Pet. Reply 9–10), Entourage (*id.* at 19–20), and searching across documents (*id.* at 22). Patent Owner's Sur-Reply identifies Petitioner's allegedly "shifting and internally inconsistent positions" (PO Sur-reply 2) and discusses True further (*id.* at 26). We have read and considered the arguments in the briefs, and the evidence cited therein.

3. Analysis

a) Receiving two strings separated by a string separator character in the first search field

Limitations [1.2] and [1.4] of independent claim 1 recite "receiving a first string in a first search field" and "receiving a second string in the first search field, wherein the first and second strings are separated by a string separator character in the first search field." Ex. 1001, 29:39–46; Ex. 1040 (claim listing including Petitioner's bracketed claim references). Independent claim 10 limitations [10.2] and [10.4] are identical. *See id*. at 30:33–44; Ex. 1040. Rather than mapping the prior art references to these claims on an element-by-element basis, Petitioner asserts that the "Entourage references in combination with True teach these claim elements, especially in view of the knowledge of a POSITA." Pet. 52. Petitioner goes on to argue that Schwartz teaches receiving one search string (*id.*, citing Ex. 1004, 9, 38) "and at least suggest[s] receiving two search strings" (*id.*, citing Ex. 1006, 387 (Figure 11-7, providing the name "Orson Welles" in a search field (reproduced *supra*)). Petitioner also asserts that "[t]he Entourage references also expressly disclose 'searching for text strings' (plural), at least suggesting to a POSITA to receive multiple search strings in the single search field of Entourage's incremental Quick Find search." *Id.* at 53 (citing Ex. 1004, 267, 269). Lastly, Petitioner asserts that "[a person of ordinary skill in the art] would have been motivated to incorporate True's express teachings to receive two search strings in the search field." *Id.* at 54 (citing Ex. 1009, 2:26–31, 2:39–48, 4:38–43, 4:65–5:6, 5:41–50, Fig. 2). Petitioner has therefore set out three positions for how Entourage or True allegedly disclose the claimed first and second partial search strings.

Petitioner's first position, that Entourage "at least suggest[s] receiving two search strings," relies on an example screenshot showing a user's search, which includes the name "Orson Welles." Pet. 58 (citing Ex. 1006, 387). But Petitioner offers no evidence that the Entourage software described in Schwartz treats the name "Orson Welles" as two separate strings. Semantically, in a human mind, "Orson Welles" contains two words separated by a space. But there is no indication that "Orson" or "Welles" are searched separately by Entourage as required by the claims. To the contrary, it appears that Entourage treats "Orson Welles" as a single text string. *See, e.g.*, Ex. 1006, 387 (explaining how to remove hits to "Orson Welles" from search results in order to ignore emails that were sent to a person with that exact name, thus indicating that "Orson Wells" is a single search string, not two strings that are searched separately); Ex. 2001 ¶ 150 (Mr. Haeberli's testimony that "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.'").

Petitioner's second position, that the Entourage references expressly disclose searching for "text strings" (plural) relies solely on references to the term "text strings" in the index at the end of Schwartz. *See* Pet. 53; Ex. 1004, 267, 269. These portions of the index cite to pages 10–11 of Schwartz, which merely describe the filter and Find command discussed above. *See id.* at 10–11; Ex. 2001 ¶ 151. Nothing in the cited portions of the Schwartz index, or on pages 10–11 of Schwartz, states or even remotely implies that Entourage permits a user to enter two strings in a single search box, much less to do so with a string separator character. *See* Ex. 1004, 10–11, 267, 269.

Petitioner's third position, that it would have been obvious to incorporate True's teachings regarding two search strings into Entourage, has two subparts. *See* Pet. 54. The first subpart pertains to what True teaches, and the second subpart pertains to whether it would have been obvious to a person of ordinary skill in the art to modify Entourage, in some fashion, with True's teaching.

As to the first subpart, True teaches breaking up a single search string containing two words separated by a space, e.g., "queen mother," and searching for the separate words, i.e., "queen" and "mother." Ex. 1009, 5:40–50 (describing "matching a word in the query"). Thus, True teaches a search where two strings (the words "queen" and "mother") are separated by a string separator character. *See* Ex. 1098, 39:10–12 (Mr. Haeberli conceding that "there must, of necessity, be a way of distinguishing the two search strings from each other" when using a search field). However, True's searches are performed differently than Entourage's searches.

In Entourage, the user is looking for documents or records that contain certain text or whose metadata contains certain text. *See, e.g.*, Ex. 1004, 8 ("Entourage has features that make it easy to find items that meet your criteria."); Ex. 1006, 384 ("After a short time—shorter than you might think—you'll collect a lot of email messages, contacts, and other Entourage items. Trying to find a particular morsel of information just by browsing becomes impractical."). In contrast, True calculates an "interest level" which "represents the degree to which text in the window represents segments of interest in the digitized audio." Ex. 1009, 5:29–50. The reason True breaks up a string into separate words and searches for them independently to generate a score is because True does not trust the quality of the underlying text. *Id.* at 1:18–30 (stating, "the searchable information

… may not always accurately represent the portions of the underlying information … that the user is trying to find"); *see also* Ex. 2020 ¶ 114. Thus, True is breaking up the words so that if one of the two words is garbled or uncertain, but the other is clear, there is an increased chance of finding the phrase. *See* Ex. 1009, 5:41–51 (calculating the total score based on the sums of the individual words appearing in a given timeframe).

Although True is searching text, it is really searching for sounds, with text as the proxy for sounds. *Id.* at 1:19–22. In other words, True does not use its two search strings to locate documents containing the text of both strings, but rather to locate portions of a document that *might* contain at least one of the spoken words in the search query. *See* PO Resp. 50–51 (citing Ex. 2001 ¶¶ 131–32; Ex. 2020 ¶¶ 109–16); Ex. 2020 ¶ 111 (Mr. Haeberli testifying that "[i]t is possible for a transcript window [in True] to be highlighted as relevant even though the visible transcript text does not show the query word"). As Patent Owner points out, several of the results of the "queen mother" query include no hits whatsoever. PO Resp. 50 (discussing Ex. 1009, Fig. 2 (reproduced *supra*), pointing to search result list 22). Thus, although True discloses a search that can be characterized as searching for two strings, True's search is very different from Entourage's search.

As to the second subpart of Petitioner's third argument, despite the marked differences between the goals of Entourage and True and how they operate, Petitioner asserts that it would have been obvious to use True's two search strings "to provide more targeted searching, as was common sense and well known in the art." Pet. 54 (citing Pet. 10–11; Ex. 1057, 27); *see also id.* at 60–61 (similar). But we are not persuaded by a preponderance of the evidence that True provides more targeted or precise searching. True itself appears silent on the issue. Petitioner cites Exhibit 1057, but all that

exhibit stands for is the general premise that doing a Boolean AND search in a Web search engine reduces the number of search results. Ex. 1057, 27. Petitioner does not allege that True uses a Boolean AND search of the kind discussed in Exhibit 1057. Thus, the statement in Exhibit 1057 regarding reducing the number of search results is tied to a manner of performing Boolean searches that Petitioner does not propose to incorporate from True into Entourage. In other words, Exhibit 1057 is of little relevance to the Entourage-True combination, and does not serve as a factual underpinning in support of Petitioner's reasoning. Petitioner also cites to several other references that discuss Boolean-style searches, but these generic background disclosures do not explain why a person of ordinary skill in the art would have considered it obvious to use one or more features of the probabilistic-based interest level search from True in a document-finding application like Entourage. *See* Pet. 10 (string citing over a dozen references with no pinpoint citations). Accordingly, there is little evidence of any reason, other than hindsight, for why one of ordinary skill would have incorporated some aspect of True's text-as-proxy-for-sounds system into Entourage's filter or Find feature to provide more targeted searching. Ex. 2020 ¶ 115 (Mr. Haeberli testifying that "a POSITA working on the problem of searching large document collections would not look to [True,] a system designed for navigating imprecise transcriptions of single audio files, as the technical approaches and constraints are entirely different").

Patent Owner argues, and we agree, that another problem with Petitioner's combination is that True does not appear to actually provide more targeted searching. PO Resp. 59 ("True's search capabilities are inherently imprecise") (citing Ex. 1009, 1:18–27, 3:20–33, 3:43–46; Ex. 2001 ¶ 137); *see also* Ex. 1009, 4:31–37 (explaining that True uses a speech

recognition system that calculates confidence levels based on a "% likelihood that [a word] corresponds to the portion of the digitized audio [at a given time]"); Ex. 2020 ¶ 109 (Mr. Haeberli stating that "the purpose of True is not to definitively locate where a search string actually appears"). Petitioner does not explain what about True would make some aspect of Entourage's filter or Find features more targeted. Even if it was known to use Boolean AND searches, for example, Petitioner is not citing True to modify Entourage to include a Boolean search. True uses an "interest level" search instead, which is more inclusive than a Boolean AND search (it adds together hits for "queen" OR "mother," and may return search results containing neither). Ex. 1009, 5:29–50, Fig. 2 (showing results list 22 having several results which do not indicate any of the searched-for words); *see* Ex. 2020 ¶ 111 (Mr. Haeberli testifying that True's interest level search would also include "text [that] does not show the query word").

Petitioner does not appear to be proposing to incorporate an interest level search or a Boolean-style search into the filter or Find features, and we are not persuaded that True in fact provides more targeted searching. Even if the quality of the transcripts in True happened to be perfect (*see* Pet. Reply 9–10, arguing that "True describes searching a text file that has 100% accuracy"), the fact remains that True is attempting to find text in particular time windows using an interest level search based on the probability that a word or words were said in a given time window, which is completely different from how Entourage performs searches (*see* Ex. 2020 ¶¶ 108–11, 114–15).

In its Reply, Petitioner states that True teaches "an incremental search for multiple search strings entered in a search field." Pet. Reply. 9. This is a broad generalization of how True operates, set forth with the benefit of

hindsight. But as we explain above, True's interest-level search is different from both of the kinds of searches performed in Entourage and from the Boolean-style web searches described in the background references. *See, e.g.*, Ex. 1057, 27. This means that Petitioner is not merely proposing that we take teachings from one system and apply them to another system. True is doing something different, and it's not clear how True's search would actually provide "precise, targeted search results" (Pet. Reply 3) given that True appears to suggest a less-targeted search than required by the claims. *See, e.g.*, Ex. 1001, 29:51–59, requiring locating first and second groups of documents that have "at least a [word] that begins with the [relevant] string" and then "identifying one or more documents that are included in each of the first and second groups of documents"); *see also* Ex. 2001 ¶¶ 129 (Mr. Haeberli testifying that "Entourage and True are directed to completely different purposes"), 130–32 (further analysis); Ex. 2020 ¶ 109 (stating that "the purpose of True is not to definitively locate where a search string actually appears"), ¶ 111 (stating that True may return text that does not even have the search word). True does not teach, suggest, or support Petitioner's general premise that there is a benefit for an incremental search for multiple search strings entered in a search field or Petitioner's argument that such searches would provide precise, targeted search results.

The problems with Petitioner's contentions become more apparent when the Petition turns to the "string separator character" of claims 1 and 10. Notably, it is the claimed string separator character that converts a field with a single search string into a field with two search strings. *See, e.g.*, Ex. 1098, 37:12–41–4 (deposition of Mr. Haeberli) (discussing that a string separator character is what instructs the computer to break up one string of

characters into separate strings of characters). Thus, one cannot discuss the string separator character apart from the limitations requiring first and second strings in a single search field. Nevertheless, Petitioner treats the "string separator character" in isolation. *See* Pet. 55–56. Petitioner's reason for modifying one or both of Entourage's filter and/or Find feature(s) to have a string separator character is because "it was common to [use string separator characters]." *Id.* at 55. However, even if something was known, this does not mean it would have been obvious to incorporate it into one or both of the search features in Entourage. *KSR*, 550 U.S. at 418 ("a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art").

Adding a string separator character to divide a single string into two separately searched strings changes how the software would function. *See* Ex. 1098, 39:22–40:8 (Mr. Haeberli testifying that software has to look for a string separator character in order to know multiple search strings were entered in the field). The space character in "Orson Welles," used in the Advanced Find example in Barber, for example, is not a string separator character. This is because one looking to exclude hits with "Orson Welles" is searching for the string "Orson Welles"—the space is another character in the string whose inclusion serves a specific semantic purpose (delineating between a first and last name). *See* Ex. 2001 ¶ 150 ("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.'"); Ex. 1006, 387 (discussing how exclude the specific person "Orson Welles" from the search); *see also* Ex. 1003 ¶ 128 (Dr. Terveen implicitly acknowledging that the space in

"Orson Welles" is not a string separator character). Thus, in Entourage, the space character in "Orson Welles" does not signal to the computer to break the string up into two parts at the space and perform separate searches for the strings on either side of the string separator and then join those search results together. Petitioner has not provided a persuasive reason to modify Entourage's filter or Find function to include a string separator character to enable the first search field to have first and second strings that are separately searched. Petitioner proposes to modify a sophisticated piece of software to meaningfully change its functionality simply because it was allegedly known to use string separator characters in other contexts. This meager rationale is not commensurate with the significant differences between the functionality of Entourage and the claimed subject matter. Accordingly, for this additional reason, we find that Petitioner's reason to combine Schwartz, Sebastian, Barber, and True does not have sufficient factual underpinnings to meet a preponderance of the evidence standard.

For the reasons set out above, and based on the entire record before us, we determine that Petitioner has not demonstrated persuasively that one of ordinary skill in the art would have had reason to combine the teachings of Schwartz, Sebastian, Barber, True, and Baeza-Yates in a manner that would satisfy limitations [1.2] and [1.4] of claim 1 or limitations [10.2] or [10.4] of claim 10, or that such a person would have had a reasonable expectation of success in so doing. *See* Pet. 57–66, 81–82. Petitioner's contentions for claims 2–9 (which depend from claim 1) rely on the same arguments discussed above and are unpersuasive for the same reasons. *See* Pet. 71–81. Accordingly, Petitioner has not demonstrated by a preponderance of the evidence that claims 1–10 are unpatentable as obvious over Schwartz, Sebastian, Barber, True, and Baeza-Yates.

b) Incrementally searching for each of the two strings and identifying one or more documents included in both sets of search results

Limitations [1.3] and [1.5] of independent claim 1 recite the steps of performing incremental searches (i.e, "incrementally locating") for each of the two strings in order to locate two groups of documents, and then identifying one or more documents that are included in each of the two groups of documents. Ex. 1001, 29:41–51; Ex. 1040 (Petitioner's claim listing including bracketed references). Limitations [10.3] and [10.5] of independent claim 10 are substantively similar. *See* Ex. 1001, 30:38–49; Ex. 1040.

Given that these steps are the implementation of a search using the search strings gathered in prior steps [1.2], [1.4], [10.2], and [10.4], Petitioner's analysis is largely similar to what we have already discussed and found lacking. *See* Pet. 63–72, 82. Indeed, Petitioner's discussion of why it would have been obvious to modify Entourage in its analysis of these claim elements is "to provide more targeted searching," which is the same insufficient reason as discussed above. *Id.* at 60. Petitioner adds that it would have been obvious "to incrementally locate documents containing two strings (as opposed to just one string) to provide the user with *more relevant results*" (*id.* at 63 (emphasis added)), but this presumes that it would have been obvious to modify Entourage to include two search strings separated by a string separator character as claimed, which we have already determined Petitioner has not sufficiently shown.

Even taking Petitioner's "more relevant results" rationale as an explanation of why a person of ordinary skill in the art would have had reason to use a string separator character to separate a string in a single

search field into two strings, and then to separately search those two strings, and also to do all of that incrementally, we still find Petitioner's explanation lacking. Even if each of these elements were taught or suggested in isolation, it would remain unclear how the prior art is being applied to Entourage and what justifications are being offered for those modifications. Petitioner states that "incrementally locat[ing] documents containing two strings" "is suggested by the Entourage references," yet Petitioner does not cite to an example of Entourage incrementally locating documents containing two search strings. Pet. 63. Petitioner also alleges that "incrementally locat[ing] documents containing two strings" "was known to a POSITA," yet the citations provided do not appear to teach incrementally locating documents containing two search strings. *Id.* at 63 (citing "Section IV.B.2" [Pet. 10–11]; Ex. 1057, 27; Ex. 1021, 125–127; Ex. 1003 ¶ 139). Exhibit 1057 is a book discussing web search engines, and page 27 does not talk about incrementally locating documents containing two search strings. Exhibit 1021 (Raskin) talks about incremental searches (e.g., Ex. 1021, 125) and even incremental Boolean searches (*id.* at 127), but it does not discuss incremental searches using two search strings in a search field separated by a string separator character.[15] Even if it did, that would merely establish knowledge; it would not provide an adequate reason to combine. Exhibit 1003 is Petitioner's expert declaration, which merely restates the language of the Petition. Ex. 1003 ¶ 139. Having reviewed the evidence cited by Petitioner, we agree with Patent Owner's declarant, Mr. Haeberli, that Petitioner has not shown the type of incremental searching *claimed* was well

[15] Raskin is discussed in more depth above in our analysis of Ground 1, above.

known to one or ordinary skill in the art at the time of the invention. Ex. 2020 ¶¶ 18–25; Ex. 2001 ¶¶ 39–47.

As explained above, the Entourage references disclose incremental searching via the filter function but not the Find function, and can search for text within a selected document using the Find function but not the filter function. Patent Owner's complaint that Petitioner fails to explain which aspects of True are being used to modify one or both of Entourage's filter and/or Find functions is not an attempt to require "bodily incorporation," as Petitioner alleges. PO Resp. 50–60; Pet. Reply 9–10. Part of an obviousness analysis is establishing the "differences between the prior art and the claims at issue." *Graham*, 383 U.S. at 18. Having done so, we must then evaluate whether Petitioner has articulated "some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006). Here, a recurring problem with Petitioner's analysis is that it treats each of the differences between the prior art and individual claim limitations independently, such that Petitioner never cogently accounts for the differences between the prior art and the claims *as a whole*. *Stratoflex*, 713 F.2d at 1537.

For example, let us first assume Petitioner is proposing to modify the filter in Entourage to allow searching for two search terms, separated by a string separator character, resulting in two sets of documents that are then reviewed in order to identify and display documents appearing in both sets. Even if we assume that True teaches a two-search-string incremental search using a string separator character, Petitioner would still need to establish a reason, with sufficient factual underpinnings, to modify Entourage's filter. Petitioner states the modification would "provide more targeted searching."

Pet. 50, 54, 56–57, 60. The evidence Petitioner cites, however, is unavailing. As discussed above, Exhibit 1057 merely discusses Boolean AND operators in a Web search, and Dr. Terveen merely repeats the same language as the Petition. Ex. 1003 ¶ 139. True does not itself suggest any particular "more targeted searching" benefit; True suggests the opposite. Ex. 2020 ¶ 114 (Mr. Haeberli testifying that "when [True] returns a search result, that does not necessarily even mean the term is in the search result"); Ex. 2001 ¶ 137 (stating that "True, by the nature of its invention, is imprecise"). Thus, there is an insufficient factual basis for Petitioner's rationale.

In addition, we do not find True's search to be "more targeted" because True's searches are performed within a single document, not across a database of documents like in Entourage, and because True's search is looking for windows of time where a calculated interest level is met, not looking for the presence of first and second strings. Ex. 1009, 5:29–50; *see also* Ex. 2020 ¶ 111 (Mr. Haeberli testifying that True's interest level search would also include "text [that] does not show the query word"); Ex. 2001 ¶ 137 (stating, "it would not have made any sense to turn to True's inherently imprecise interest level method"). Put another way, Petitioner's reason for combination is contrary to what True actually teaches.

Now let us instead assume that Petitioner is proposing to modify the Find feature in Entourage to allow incremental searching for two search terms, separated by a string separator character, resulting in two sets of documents that are then reviewed to identify and display documents appearing in both sets. A combination including the Find function would have additional challenges beyond those we identified above with respect to the filter function. For example, an incremental search displays results as

they are typed, but the Entourage Find feature builds a query in a pop up window and then displays the results in a separate window after the query is executed. In such a combination, how would incremental results be displayed? Would the separate results window pop up as soon as the user starts typing into the Find pop up window? If so, where would it go? Wouldn't this obstruct the Find window? *See* Ex. 2001 ¶¶ 138–39 (Mr. Haeberli's testimony that Petitioner's position "does not account for or otherwise acknowledge the obstacles that need to be overcome for the combination to properly function" and that "[g]iven this separate window and the use of a delimited search that requires an explicit press of a button to initiate the search, a POSITA would not contemplate making this search incremental"). This can be contrasted with Entourage's filter function, which does allow incremental searching and in which there are no windows involved. *See* Ex. 1004, 9 (in Figure 1.12, the filter components are in-line with the address book and the results); Ex. 2020 ¶ 104 (Mr. Haeberli's testimony that, "when a product maintains two separate search mechanisms rather than combining them, it typically reflects technical constraints or usability considerations that make combination undesirable or impractical"). Petitioner's proposed combination would meaningfully change the way a user would interact with the Find feature in Entourage, yet Petitioner provides no discussion of how the Entourage features would be modified to accommodate the features from True, much less why one of ordinary skill in the art would make these specific modifications. Without knowing the particular modification alleged, we cannot determine that it would have been obvious to so modify the teachings or determine whether there would have been a reasonable expectation of success. *See* PO Resp. 61–62 (alleging Petitioner's proposed combination would not have a reasonable expectation

of success); Ex. 2001 ¶ 138 (Mr. Haeberli's testimony that Petitioner's declarant "does not account for or otherwise acknowledge the obstacles that need to be overcome for the combination to properly function"); Ex. 2020 ¶¶ 120–21 (asserting that modifying the Find function to have incremental searching would impose significant technical hurdles).

To be clear, we are not trying to discern the specific coding modifications required to modify the Find function to be incremental. *Cf. In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012) ("It is well-established that a determination of obviousness based on teachings from multiple references does not require an actual, physical substitution of elements."). Even at a conceptual level, however, using Entourage's specific implementations as an example, because the prior art shows different ways to perform searches (e.g., filter vs. Find), due consideration must be given to the way the user interacts with the relevant functionality and the technical implications of incremental search. *See* Ex. 2001 ¶¶ 139–40 (Mr. Haeberli's testimony that the way the Find function is presented in Entourage would not be amenable to incremental search because of the interface as well as the time it takes to do a Find); Ex. 2020 ¶ 118 (a person of ordinary skill "would not expect incremental search in the Find or Advanced Find features of the Entourage references to yield a successful, workable search system due to speed and efficiency concerns"). The Petition does not do so.

In sum, for the reasons set out above and based on the entire record before us, Petitioner has not identified sufficient factual underpinnings in support of its proposal to modify Entourage to perform incremental searches for each of the two strings in order to locate two groups of documents, and then identify one or more documents that are included in each of the two groups of documents, as required by limitations [1.3], [1.5], [10.3], and

[10.5] of independent claims 1 and 10. *See* Ex. 1001, 29:41–51, 30:38–49. Petitioner's contentions for claims 2–9 (which depend from claim 1) rely on the same arguments discussed above and are unpersuasive for the same reasons. *See* Pet. 71–81. Accordingly, for this separate and independent reason, Petitioner has not demonstrated by a preponderance of the evidence that claims 1–10 are unpatentable as obvious over Schwartz, Sebastian, Barber, True, and Baeza-Yates.

III. CONCLUSION

For the reasons set forth above, we determine that Petitioner has not established by a preponderance of the evidence that any claim of the '035 patent is unpatentable as obvious under 35 U.S.C. § 103.

The following table summarizes the outcome of this proceeding:

Claim(s)	35 U.S.C. §	Reference(s)/Basis	Claim(s) Shown Unpatentable	Claim(s) Not Shown Unpatentable
1–10	103	Wilcox, Londergan, Raskin, Wu		1–10
1–10	103	Schwartz, Sebastian, Barber, True, Baeza-Yates		1–10

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that Petitioner has not established that any of claims 1–10 of U.S. Patent No. 7,370,035 B2 are unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

FOR PETITIONER:

Todd Siegel
Amy Haspel
Andrew Mason
Samuel Thacker
KLARQUIST SPARKMAN, LLP
todd.siegel@klarquist.com
amy.haspel@klarquist.com
andrew.mason@klarquist.com
samuel.thacker@klarquist.com

FOR PATENT OWNER:

Lori Gordon
Christie Larochelle
Sanjeet Dutta
GOODWIN PROCTER LLP
gordon-ptab@goodwinlaw.com
clarochelle@goodwinlaw.com
sdutta@goodwinlaw.com