Patent Owner's Demonstratives

Microsoft Corp. v. X1 Discovery, Inc.

IPR2025-00253 – U.S. Patent No. 7,370,035
IPR2025-00254 – U.S. Patent No. 10,552,490
IPR2025-00255 – U.S. Patent No. 9,633,139

April 27, 2026

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

The State of the Art and The Challenged Patents

The Inventors Identified Problems with Existing Search Applications

2. Description of the Related Art

Conventional search application programs often are slow and cumbersome to use. For example, many search engines require the user to type in a search term, click on a search button, and review the results. If the user is not satisfied with the results, as might be the case if too many items or items that are not of interest are found, then the user needs to edit the search terms, click on an initiate search button, and again review the results. In addition, many search application programs only provide for limited search filtering, wherein the user is limited to filtering the search based on dates, file location, and file contents.

'035 patent (EX1001), 1:32-42; '035 Paper 21 (POR), 5-6
'490 patent (EX1001), 1:43-53, '490 Paper 21 (POR), 5
'139 patent (EX1001), 1:37-48, '139 Paper 21 (POR), 5-6

Another drawback of some search engines is that the actual search process takes an annoying amount of time to complete a search. Furthermore, some search engines do not include search features directed to specific types of search targets, such as email, Web pages, and files. Alternatively and inconveniently, separate tools may be needed to search for email, Web pages, and files if the user desires corresponding specific search features.

'035 patent (EX1001), 1:43-50; '035 Paper 21 (POR), 5-6
'490 patent (EX1001), 1:54-61, '490 Paper 21 (POR), 5
'139 patent (EX1001), 1:49-56, '139 Paper 21 (POR), 5-6

Problems with prior art search applications:

- **Searches were cumbersome**
- **Searches took a significant amount of time to complete**
- **Searches would have to be redone each time if the search strings were not returning the desired results**

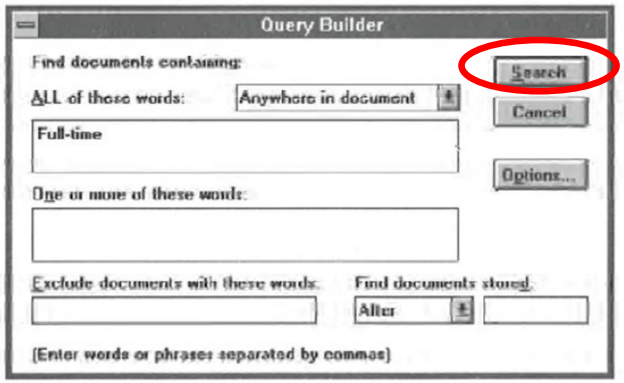
Problem Address by the Challenged Patents

- **Consumers were acquiring more electronic files, making it difficult to locate what you were looking for**
- **Traditional search addressed this by using:**
  - **Delimited search** when searching across documents
  - Search "builder" tools with **multiple text boxes** to help a user build its search
  - Use of **Booleans** to help narrow results
- **But these systems were still slow and cumbersome**

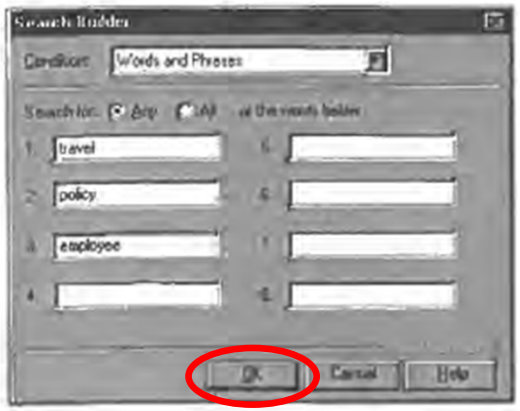
'035 Paper 21 (POR), 2-5
'035 EX2001 (Haeberli Decl.), ¶38
'490 Paper 21 (POR), 2-5
'490 EX2001 (Haeberli Decl.), ¶38
'139 Paper 21 (POR), 2-5
'139 EX2001 (Haeberli Decl.), ¶38

How the Prior Art Search Systems Addressed the Problem

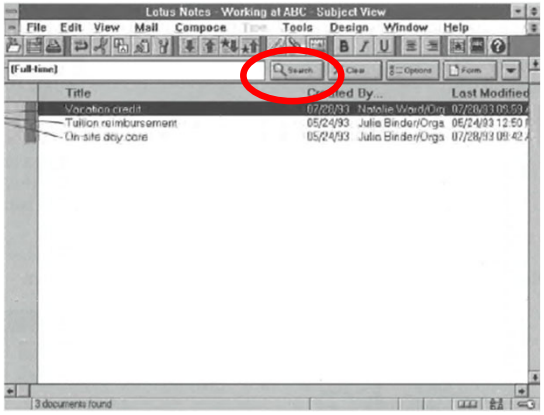
Wilcox (Lotus v.3) Query Builder



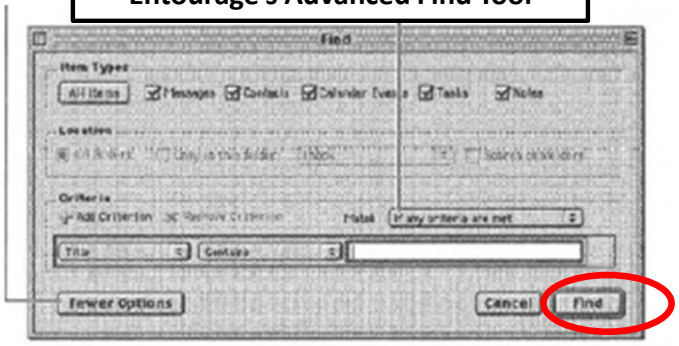
Londergan (Lotus R5) Search Builder



Wilcox (Lotus v.3) Search Bar



Entourage's Advanced Find Tool

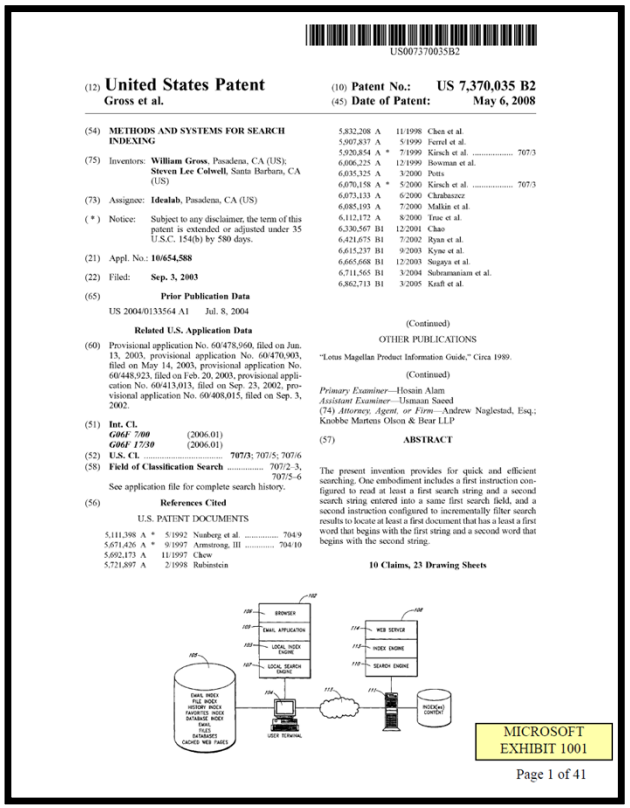


Delimited search button

- **A user types in a search string and must click a 'search' button or press the 'enter' key before results are returned**
- **Usually a single string per box or multiple strings with Booleans**

EX1059 (Wilcox), 163, 165
EX1061 (Londergan), 142
EX1004 (Schwartz), 11

The Solution: the ’035, ’490, and ’139 Patents



US007370035B2

(12) **United States Patent**
Gross et al.

(10) **Patent No.: US 7,370,035 B2**
(45) **Date of Patent: May 6, 2008**

(54) **METHODS AND SYSTEMS FOR SEARCH INDEXING**

(75) Inventors: **William Gross**, Pasadena, CA (US); **Steven Lee Colwell**, Santa Barbara, CA (US)

(73) Assignee: **Idealab**, Pasadena, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 580 days.

(21) Appl. No.: **10/654,588**

(22) Filed: **Sep. 3, 2003**

(65) **Prior Publication Data**

US 2004/0133564 A1 Jul. 8, 2004

Related U.S. Application Data

(60) Provisional application No. 60/478,960, filed on Jun. 13, 2003, provisional application No. 60/470,903, filed on May 14, 2003, provisional application No. 60/448,923, filed on Feb. 20, 2003, provisional application No. 60/413,013, filed on Sep. 23, 2002, provisional application No. 60/408,015, filed on Sep. 3, 2002.

(51) **Int. Cl.**
G06F 7/00 (2006.01)
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/3**; 707/5; 707/6

(58) **Field of Classification Search** 707/2–3, 707/5–6
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,111,398	A *	5/1992	Nunberg et al.	704/9
5,671,426	A *	9/1997	Armstrong, III	704/10
5,692,173	A	11/1997	Chew	
5,721,897	A	2/1998	Rubinstein	
5,832,208	A	11/1998	Chen et al.	
5,907,837	A	5/1999	Ferrel et al.	
5,920,854	A *	7/1999	Kirsch et al.	707/3
6,006,225	A	12/1999	Bowman et al.	
6,035,325	A	3/2000	Potts	
6,070,158	A *	5/2000	Kirsch et al.	707/3
6,073,133	A	6/2000	Chrabaszez	
6,085,193	A	7/2000	Malkin et al.	
6,112,172	A	8/2000	True et al.	
6,330,567	B1	12/2001	Chao	
6,421,675	B1	7/2002	Ryan et al.	
6,615,237	B1	9/2003	Kyne et al.	
6,665,668	B1	12/2003	Segaya et al.	
6,711,565	B1	3/2004	Subramaniam et al.	
6,862,713	B1	3/2005	Kraft et al.	

(Continued)

OTHER PUBLICATIONS

"Lotus Magellan Product Information Guide," Circa 1989.

(Continued)

Primary Examiner—Hosain Alam
Assistant Examiner—Usmaan Saeed
(74) *Attorney, Agent, or Firm*—Andrew Naglestad, Esq.; Knobbe Martens Olson & Bear LLP

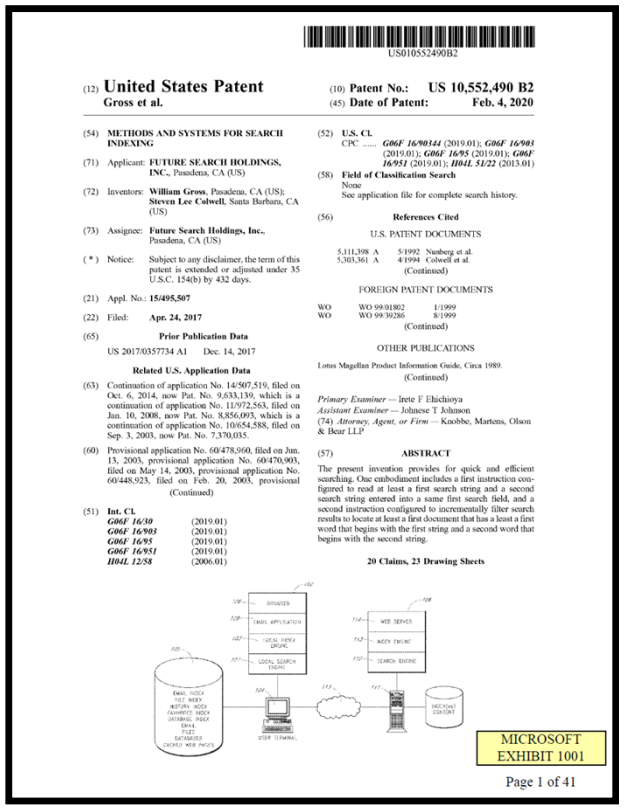
(57) **ABSTRACT**

The present invention provides for quick and efficient searching. One embodiment includes a first instruction configured to read at least a first search string and a second search string entered into a same first search field, and a second instruction configured to incrementally filter search results to locate at least a first document that has a least a first word that begins with the first string and a second word that begins with the second string.

10 Claims, 23 Drawing Sheets

MICROSOFT EXHIBIT 1001

Page 1 of 41



US010552490B2

(12) **United States Patent**
Gross et al.

(10) **Patent No.: US 10,552,490 B2**
(45) **Date of Patent: Feb. 4, 2020**

(54) **METHODS AND SYSTEMS FOR SEARCH INDEXING**

(71) Applicant: **FUTURE SEARCH HOLDINGS, INC.**, Pasadena, CA (US)

(72) Inventors: **William Gross**, Pasadena, CA (US); **Steven Lee Colwell**, Santa Barbara, CA (US)

(73) Assignee: **Future Search Holdings, Inc.**, Pasadena, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 432 days.

(21) Appl. No.: **15/495,507**

(22) Filed: **Apr. 24, 2017**

(65) **Prior Publication Data**

US 2017/0357734 A1 Dec. 14, 2017

Related U.S. Application Data

(63) Continuation of application No. 14/507,519, filed on Oct. 6, 2014, now Pat. No. 9,633,139, which is a continuation of application No. 11/972,563, filed on Jan. 10, 2008, now Pat. No. 8,856,093, which is a continuation of application No. 10/654,588, filed on Sep. 3, 2003, now Pat. No. 7,370,035.

(60) Provisional application No. 60/478,960, filed on Jun. 13, 2003, provisional application No. 60/470,903, filed on May 14, 2003, provisional application No. 60/448,923, filed on Feb. 20, 2003, provisional
(Continued)

(51) **Int. Cl.**
G06F 16/30 (2019.01)
G06F 16/903 (2019.01)
G06F 16/95 (2019.01)
G06F 16/951 (2019.01)
H04L 12/58 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 16/9014*** (2019.01); ***G06F 16/903*** (2019.01); ***G06F 16/95*** (2019.01); ***G06F 16/951*** (2019.01); ***H04L 51/22*** (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,111,398	A	5/1992	Nunberg et al.	
5,303,361	A	4/1994	Colwell et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO 99/01802	1/1999
WO	WO 99/39286	8/1999

(Continued)

OTHER PUBLICATIONS

Lotus Magellan Product Information Guide, Circa 1989.

(Continued)

Primary Examiner — Irete F Ehichioya
Assistant Examiner — Johnese T Johnson
(74) *Attorney, Agent, or Firm* — Knobbe, Martens, Olson & Bear LLP

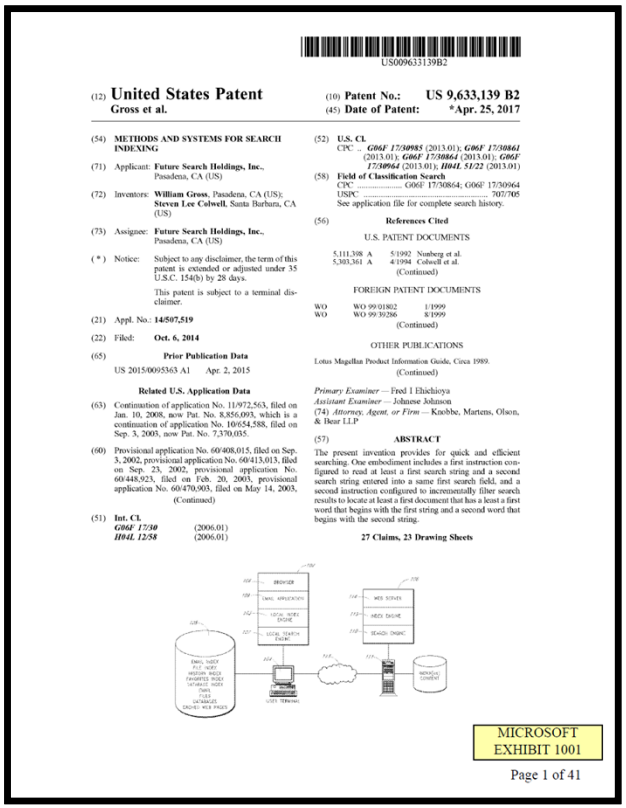
(57) **ABSTRACT**

The present invention provides for quick and efficient searching. One embodiment includes a first instruction configured to read at least a first search string and a second search string entered into a same first search field, and a second instruction configured to incrementally filter search results to locate at least a first document that has a least a first word that begins with the first string and a second word that begins with the second string.

20 Claims, 23 Drawing Sheets

MICROSOFT EXHIBIT 1001

Page 1 of 41



US009633139B2

(12) **United States Patent**
Gross et al.

(10) **Patent No.: US 9,633,139 B2**
(45) **Date of Patent: *Apr. 25, 2017**

(54) **METHODS AND SYSTEMS FOR SEARCH INDEXING**

(71) Applicant: **Future Search Holdings, Inc.**, Pasadena, CA (US)

(72) Inventors: **William Gross**, Pasadena, CA (US); **Steven Lee Colwell**, Santa Barbara, CA (US)

(73) Assignee: **Future Search Holdings, Inc.**, Pasadena, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 28 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/507,519**

(22) Filed: **Oct. 6, 2014**

(65) **Prior Publication Data**

US 2015/0095363 A1 Apr. 2, 2015

Related U.S. Application Data

(63) Continuation of application No. 11/972,563, filed on Jan. 10, 2008, now Pat. No. 8,856,093, which is a continuation of application No. 10/654,588, filed on Sep. 3, 2003, now Pat. No. 7,370,035.

(60) Provisional application No. 60/408,015, filed on Sep. 3, 2002, provisional application No. 60/413,013, filed on Sep. 23, 2002, provisional application No. 60/448,923, filed on Feb. 20, 2003, provisional application No. 60/470,903, filed on May 14, 2003,
(Continued)

(51) **Int. Cl.**
G06F 17/30 (2006.01)
H04L 12/58 (2006.01)

(52) **U.S. Cl.**
CPC ... ***G06F 17/30985*** (2013.01); ***G06F 17/30861*** (2013.01); ***G06F 17/30864*** (2013.01); ***G06F 17/30964*** (2013.01); ***H04L 51/22*** (2013.01)

(58) **Field of Classification Search**
CPC G06F 17/30864; G06F 17/30964
USPC 707/705
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,111,398	A	5/1992	Nunberg et al.	
5,303,361	A	4/1994	Colwell et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO 99/01802	1/1999
WO	WO 99/39286	8/1999

(Continued)

OTHER PUBLICATIONS

Lotus Magellan Product Information Guide, Circa 1989.

(Continued)

Primary Examiner — Fred I Ehichioya
Assistant Examiner — Johnese Johnson
(74) *Attorney, Agent, or Firm* — Knobbe, Martens, Olson, & Bear LLP

(57) **ABSTRACT**

The present invention provides for quick and efficient searching. One embodiment includes a first instruction configured to read at least a first search string and a second search string entered into a same first search field, and a second instruction configured to incrementally filter search results to locate at least a first document that has a least a first word that begins with the first string and a second word that begins with the second string.

27 Claims, 23 Drawing Sheets

MICROSOFT EXHIBIT 1001

Page 1 of 41

The Invention Overcomes Problems Identified in the Art

SUMMARY OF THE INVENTION

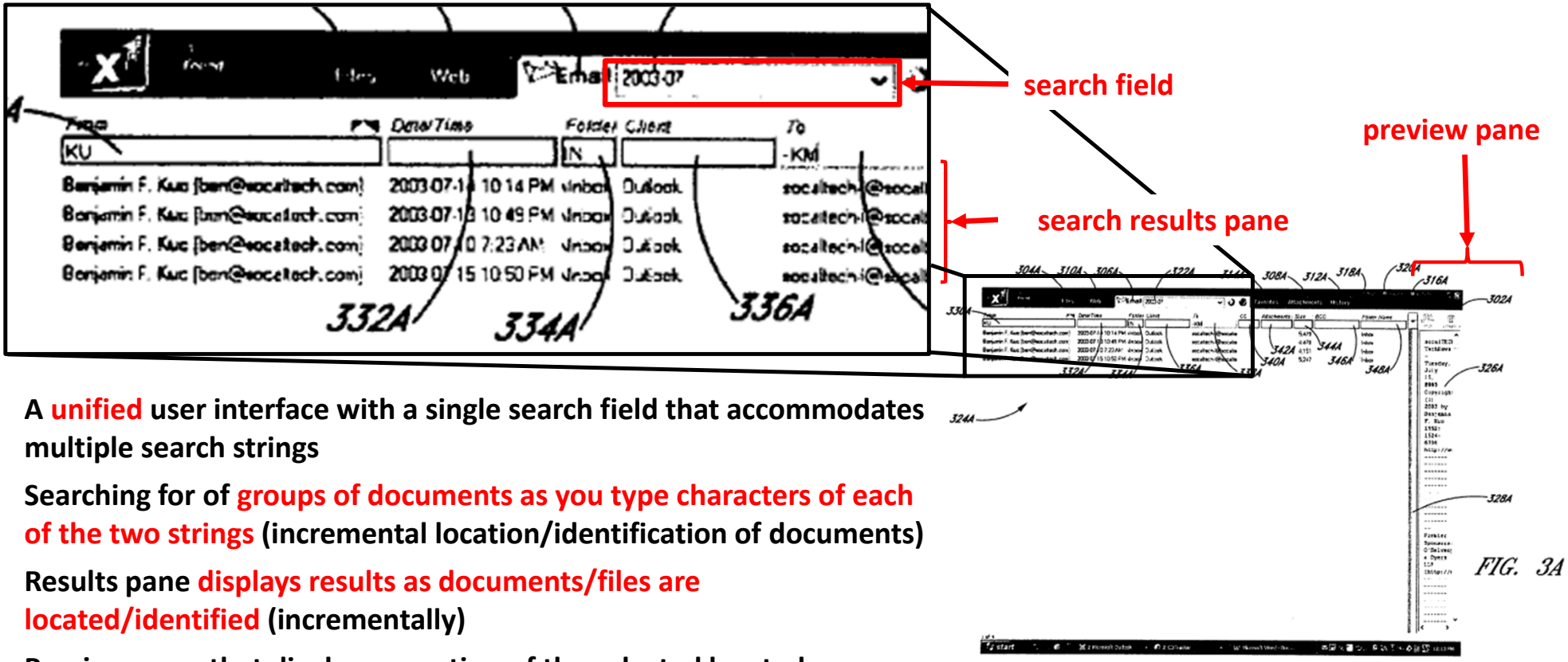
One example embodiment of the present invention advantageously provides for incremental or reactive searching of a variety of search targets. For example, the search targets can include one or more of files, emails, email attachments, Web pages, specific databases, and/or the like. Because a search is performed incrementally, the search results are provided or narrowed substantially immediately after each character in a search string is entered by a user. Thus, 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.

'035 patent (EX1001), 2:9-21
'035 Paper 21 (POR), 6
'490 patent (EX1001), 2:20-32
'490 Paper 21 (POR), 6
'139 patent (EX1001), 2:15-27
'139 Paper 21 (POR), 6

The inventors went in a new direction:

- **A new way of incrementally searching for and locating documents, files, emails, etc. such that search results populate *as the user types* the search string**
- **Allows dynamic searching and viewing across files**

The Invention



- A **unified** user interface with a single search field that accommodates multiple search strings
- Searching for of **groups of documents as you type characters of each of the two strings** (incremental location/identification of documents)
- Results pane **displays results as documents/files are located/identified** (incrementally)
- Preview pane that displays a portion of the selected located document

'035 patent (EX1001), Figure 3A; '035 Paper 21 (POR), 6-7
'490 patent (EX1001), Figure 3A; '490 Paper 21 (POR), 6-7
'139 patent (EX1001), Figure 3A; '139 Paper 21 (POR), 6-7

"Incremental Search" Within Single Document Is Not the Same as Incrementally Locating Groups of Documents

Prior Art uses the phrase "incremental search" as shorthand for searching within a single document

- **This is fundamentally different than what is claimed.**

The prior art's disclosures of single-document "incremental search" (i.e., sequentially scanning the text of a single open document) and list filtering do not translate to searching for groups of documents as the user types characters (incremental location of groups of documents):

- **Different approach to how to access document contents efficiently**
- **Different approach to managing memory and resource load**
- **Different approach to returning results at acceptable speed**

No prior art disclosed incrementally locating groups of documents

'035 Paper 21 (POR), 4-5
'490 Paper 21 (POR), 4-5
'139 Paper 21 (POR), 4-5
EX1021 (Raskin), 124-30

Prior Art "Incremental Search" Is Brute Force Sequential Scanning of a Single Open Document

The less common strategy is the **incremental search**, a popular example of which is found in EMACS, an editor used with the UNIX operating system (Stallman 1993). In most implementations of incremental searches, as with the delimited search, the user first summons a dialog box that contains a field in which the user can enter the pattern. When he types the first character of the pattern, however, the system uses this character alone as a complete pattern and immediately begins to search for the first instance of that character in the chosen search direction. If an instance of that first character is found before the next character of the pattern is typed, the instance is selected and the cursor placed immediately after the end of the selection. If the next character of the pattern is typed before an instance is found, that character is added to the pattern and the search continues, now looking for an instance of the now extended pattern. This method is repeated as each character is added to the pattern.

EX1021 (Raskin), 125
'035 EX2020 (Suppl. Haeberli Decl.), ¶11
'490 EX2020 (Suppl. Haeberli Decl.), ¶11
'139 EX2020 (Suppl. Haeberli Decl.), ¶11

Example: EMACS text editor

- **Single open document**
- **Scans text for matching text string**
- **Moves the cursor to the first instance**
- **NO list of search results returned**

The ’139 and ’035 Patents

’139 Patent, Claim 1

1. A method of performing a search comprising:
receiving a first partial search string in a first search field;
receiving a second partial search string in the first search field;
wherein the first and second partial search strings are separated by a string separator character in the first search field;
incrementally searching a set of documents in order to identify search results, wherein the incrementally searching comprises:
searching for a first one or more documents each including at least one string that corresponds to the first partial search string;
searching for a second one or more documents each including at least one string that corresponds to the second partial search string; and
updating the search results as each successive character of the first partial search string and the second partial search string is received respectively in the first search field by identifying one or more documents included in both the first one or more documents and the second one or more documents, wherein the first partial search string and the second partial search string are non-adjacent in at least one of the identified documents.

’139 patent (EX1001), claim 1

’035 Patent, Claim 1

1. A method of performing a search comprising:
receiving a first string in a first search field;
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;
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;
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
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.

’035 patent (EX1001), claim 1

The ’490 Patent

’490 Patent, Claim 1

1. A computing system comprising:
one or more hardware computer processors;
a display device viewable by a user of the computing system;
a computer readable storage medium storing:
an index data structure comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included; and
software instructions executable by the one or more hardware computer processors causing the computing system to:
cause display of an interactive user interface on the display device, the interactive user interface including at least a first search field;
receive a first partial search string in the first search field;
receive a second partial search string in the first search field;
wherein the first and second partial search strings are separated by a string separator character in the first search field;
searching the index data structure for a first one or more of the plurality of files each including at least one string that corresponds to the first partial search string;
searching the index data structure for a second one or more of the plurality of files each including at least one string that corresponds to the second partial search string; and
updating a set of matching files of the plurality of files as each successive character of the first partial search string and the second partial search string is received respectively in the first search field by identifying any of the plurality of files included in both the first one or more of the plurality of files and the second one or more of the plurality of files, wherein the first partial search string and the second partial search string are non-adjacent in at least one of the matching files.

’490 patent (EX1001), claim 1

Instituted Grounds

Ground	References	Basis	'035 Patent Claims	'490 Patent Claims	'139 Patent Claims
1	Wilcox (EX1059), Londergan (EX1061), Raskin (EX1021), and Wu (EX1019)	35 U.S.C. § 103	1-10	1-20	1-27
2	Schwartz (EX1004), Sebastian (EX1005), Barber (EX1006), True (EX1009), and Baeza-Yates (EX1023)	35 U.S.C. § 103	1-10	1-20	1-27

All Challenged Claims are Patentable Over Petitioner's Grounds

Petitioner's Grounds Fail

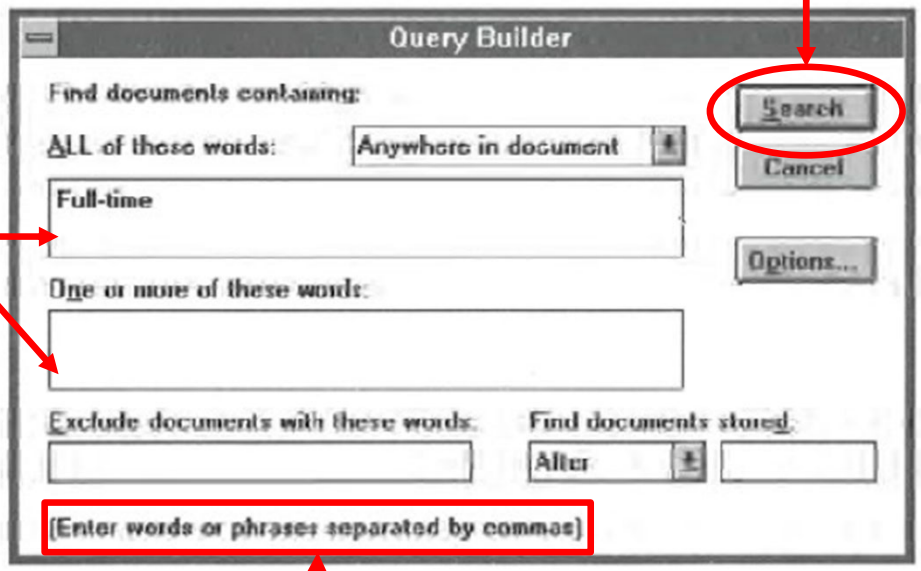
- **None of Petitioner's cited art discloses or renders obvious incrementally searching for and locating documents using multiple search strings for either Ground**
  - Petitioner's combinations fail to disclose all limitations of the claims
  - Petitioner fails to establish motivation to combine with a reasonable expectation of success for either Ground
    - ***Petitioner changes its stated motivation to combine in Reply for both Grounds***
  - Petitioner improperly gap-fills for certain dependent claims in both Grounds with knowledge of a POSITA

All Challenged Claims are Patentable over the Wilcox-Londergan-Raskin-Wu Combination

The Lotus Notes References Describe Delimited Searching Across Multiple Documents

Wilcox Query Builder

Delimited search button



Search string in search boxes

strings separated by comma

- **The Query Builder search tool performs delimited searches only (must hit the search button to search).**
- **The Query Builder search tool <u>does not</u> incrementally search for and locate documents**

EX1059 (Wilcox), 163
'035 Paper 21 (POR), 11
'490 Paper 21 (POR), 11
'139 Paper 21 (POR), 11

Wilcox and Londergan Describe *Multi-Document* Search

Sometimes you might want to mark documents as read or unread. Say, for example, you add a new discussion database to your workspace and it has 1,000 documents in it. Chances are you will not read all 1.000 documents. but rather will join new discussions as

EX1059 (Wilcox), 85
'035 Paper 37 (Sur-Reply), 4; '490 Paper 38 (Sur-Reply), 4; '139 Paper 37 (Sur-Reply), 4

Before you try searching, you should understand full-text indexing. A search for text is much speedier and offers more options when you have a *full-text index*, a behind-the-scenes list of text and documents that the search uses to find documents containing the text you want. If you're a manager or designer of a database or if you have a copy of the database on your own computer, you can create your own index. How about indexing your e-mail database? That way, you can easily find all those old messages that have your customer's names in them or quickly put your hands on all the messages that are about a certain project. You don't absolutely have to create a database FTI (as the experts sometimes call a full-text index), but an FTI does make searching easier.

EX1061 (Londergan), 135
'035 Paper 1 (Petition), 17; '490 Paper 1 (Petition), 21; '139 Paper 1 (Petition), 24

Wilcox and Londergan describe

- **indexed search systems/tools**
- **designed to search for documents**

within databases containing large numbers of documents

Raskin Describes Sequential Scanning of a Single Document

The less common strategy is the **incremental search**, a popular example of which is found in EMACS, an editor used with the UNIX operating system (Stallman 1993). In most implementations of incremental searches, as with the delimited search, the user first summons a dialog box that contains a field in which the user can enter the pattern. When he types the first character of the pattern, however, the system uses this character alone as a complete pattern and immediately begins to search for the first instance of that character in the chosen search direction. If an instance of that first character is found before the next character of the pattern is typed, the instance is selected and the cursor placed immediately after the end of the selection. If the next character of the pattern is typed before an instance is found, that character is added to the pattern and the search continues, now looking for an instance of the now extended pattern. This method is repeated as each character is added to the pattern.

EX1021 (Raskin), 125
'035 Paper 21 (POR), 19
'490 Paper 21 (POR), 19
'139 Paper 21 (POR), 19

Raskin's search tool

1. **sequentially scans through the characters in the text of**
2. **a single, open document**
3. **moves the cursor to the first instance of the search string**

Raskin's search tool does not

- **incrementally search for and locate documents**
- **return search results**

Raskin Describes Sequential Scanning *Within* a Single Document

The speed of incremental searches is increased when a few implementation tricks are used; for example, when you type the first character of the search string, the computer instantly proceeds to find the first instance of that character in the text, which is highlighted and, if it is not there already, moved with its context into the display window. This is usually a fast process

EX1021 (Raskin), 129

In the small texts for which string searches were originally designed, searching usually proceeded from the current cursor position to the end of the text. In larger texts, it is generally a good idea to continue the search, if it has not found a target by the time it has reached the end, circularly from the beginning back to the cursor location, in case the user has forgotten that the desired target is farther back in the text. Unpublished testing at Information

EX1021 (Raskin), 130

If an instance of the pattern does not occur in the text, the search fails. Many systems cease operating in this event and remain unusable until a particular key—typically Enter or Return—is pressed or a particular on-screen button is clicked. A modal message is placed on the display to let you know that you must make the required obeisance before you will be allowed to continue using the computer. In multidisplay systems or if the screen is visually busy, this message may be nowhere near your locus of attention. You may not notice it at all. It then seems to you that the computer will not respond to the keyboard; it seems to have crashed. In an incremental search, it is clear, without any special indication, that a search has failed; the cursor returns to its original location, and additional keystrokes do not have any effect. A short beep or a screen flash can also be helpful, especially if the search exceeded the duration of short-term memory, say, 10 seconds, so that the user has forgotten the appearance of the display prior to the search. The beep is also a useful cue for visually impaired users.

EX1021 (Raskin), 127

to review the beginning of, and any heading on, each document. When the desired document is found, an incremental search for the target is initiated. This guarantees that the chosen document is searched first; the advantage of this is that you can typically use a shorter pattern if the area to be searched is smaller. If you did not know in what document the target resided or if you did not want to look for the document, you could perform a specific search from the beginning, which would find the target anyway.

EX1021 (Raskin), 132

Wu Describes Intersecting Search Results

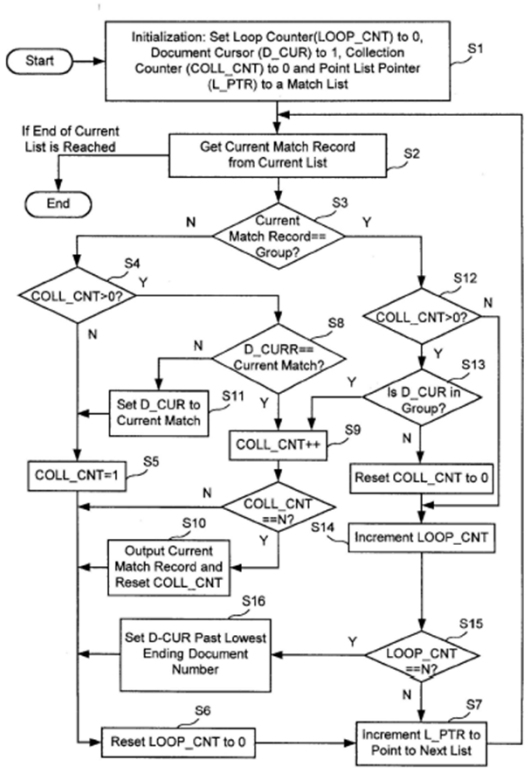


FIG. 6

EX1019 (Wu), Fig. 6

Wu describes using "match lists" to generate search results by:

1. **searching for each of two or more strings separately**
2. **generating a "match list" for each string**
3. **then performing an "AND" operation to determine which documents are in both match lists**

EX1019 (Wu), 6:51-66, 8:66-11:32
'035 Paper 1 (Petition), 22-23
'490 Paper 1 (Petition), 17
'139 Paper 1 (Petition), 20-21

Petitioner Fails to Establish Obviousness Over the Wilcox-Londergan-Raskin-Wu Combination

- **None of the four references, alone or in combination, discloses or renders obvious incrementally searching for and locating documents**
  - The combination does not result in a system or method that "incrementally locat[es] … documents" ('035) / "incrementally search[es] … documents … to identify search results" ('139) / "search[es] [for files] … as each successive character is … received" ('490)
  - Petitioner has not articulated any reasoning with rational underpinnings for motivation to combine the distinct search tools disclosed in Wilcox and Londergan
  - A POSITA would not be motivated to combine the Wilcox and Londergan delimited document search with Raskin's disclosure of brute-force sequential scanning for text within a single document
  - A POSITA would not have a reasonable expectation of success in the combination
  - Petitioner improperly gap-fills with knowledge of a POSITA for certain dependent claims

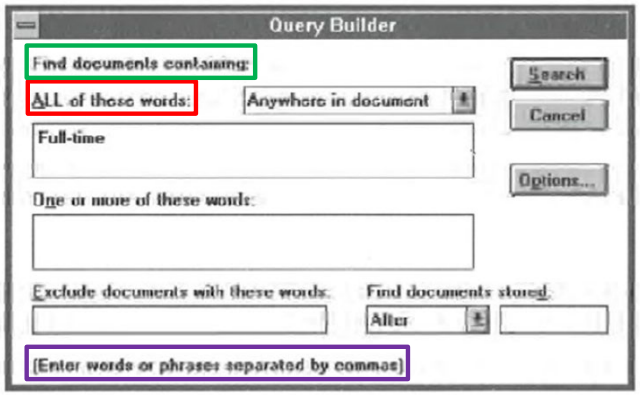
Petitioner's Picking and Choosing from Multiple References

Exemplary Claim 1, '035 Patent

1. A method of performing a search comprising:
receiving a first string in a first search field;
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;
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;
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
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.

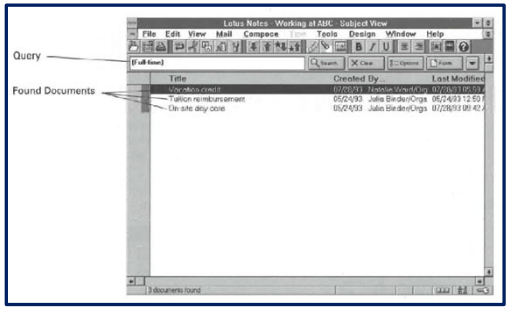
'035 patent (EX1001), claim 1

Wilcox's Query Builder



EX1059 (Wilcox), 163

Wilcox's Results Display



EX1059 (Wilcox), 164

Wu's "Intersection" Approach

The use of document repository **20** and word index **22** will now be described with reference to an example. In this example, a user is searching for documents and presents a search request with a query string "The game of Go". Search engine **36** looks up each of the terms in word index **22**. Because they are so common, "the" and "of" are either ignored by search engine **36** or word index **22** returns instructions to ignore those words, as described above. Search engine **38** then reads the document lists for "game" and "go", generates a match list for each term and applies an "AND" operator to the match lists as described below in connection with FIG. **6**.

EX1019 (Wu), 6:51-62

Raskin's "Incremental" Scanning

The less common strategy is the **incremental search**, a popular example of which is found in EMACS, an editor used with the UNIX operating system (Stallman 1993). In most implementations of incremental searches, as with the delimited search, the user first summons a dialog box that contains a field in which the user can enter the pattern. When he types the first character of the pattern, however, the system uses this character alone as a complete pattern and immediately begins to search for the first instance of that character in the chosen search direction. If an instance of that first character is found before the next character of the pattern is typed, the instance is selected and the cursor placed immediately after the end of the selection. If the next character of the pattern is typed before an instance is found, that character is added to the pattern and the search continues, now looking for an instance of the now extended pattern. This method is repeated as each character is added to the pattern.

EX1021 (Raskin), 125

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

The Wilcox-Londergan-Raskin-Wu Combination Does Not Disclose Incrementally Locating Documents

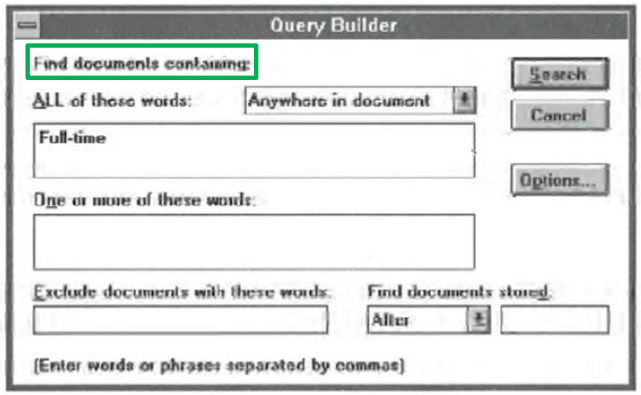
Petitioner Fails to Establish the Lotus Notes-Raskin-Wu Combination Teaches All Limitations of the Claims

Exemplary Claim 1, '035 Patent

1. A method of performing a search comprising:
receiving a first string in a first search field;
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;
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;
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
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.

'035 patent (EX1001), claim 1

Wilcox's Query Builder



EX1059 (Wilcox), 163

Raskin's "Incremental" Scanning

The less common strategy is the **incremental search**, a popular example of which is found in EMACS, an editor used with the UNIX operating system (Stallman 1993). In most implementations of incremental searches, as with the delimited search, the user first summons a dialog box that contains a field in which the user can enter the pattern. When he types the first character of the pattern, however, the system uses this character alone as a complete pattern and immediately begins to search for the first instance of that character in the chosen search direction. If an instance of that first character is found before the next character of the pattern is typed, the instance is selected and the cursor placed immediately after the end of the selection. If the next character of the pattern is typed before an instance is found, that character is added to the pattern and the search continues, now looking for an instance of the now extended pattern. This method is repeated as each character is added to the pattern.

EX1021 (Raskin), 125

- **Petitioner combines Raskin's "incremental" scanning with Wilcox's delimited search for the "incrementally locating … documents" ('035) / "incrementally searching … documents … to identify search results" ('139) / "searching [for files] … as each successive character is … received" ('490) limitations**
- **Petitioner has not established the combination teaches these limitations**

All Challenged Claims Require Incremental Search Across Documents

Claim 1 of the '035 Patent

1. A method of performing a search comprising:
receiving a first string in a first search field;
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;
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;
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
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.

'035 patent (EX1001), claim 1

Claim 1 of the '490 Patent

searching the index data structure for a first one or more of the plurality of files each including at least one string that corresponds to the first partial search string;
searching the index data structure for a second one or more of the plurality of files each including at least one string that corresponds to the second partial search string; and

'490 patent (EX1001), claim 1

Claim 1 of the '139 Patent

1. A method of performing a search comprising:
receiving a first partial search string in a first search field;
receiving a second partial search string in the first search field;
wherein the first and second partial search strings are separated by a string separator character in the first search field;
incrementally searching a set of documents in order to identify search results, wherein the incrementally searching comprises:

'139 patent (EX1001), claim 1

Neither Wilcox/Londergan Nor Raskin Disclose Incremental Search Across Documents

The less common strategy is the **incremental search**, a popular example of which is found in EMACS, an editor used with the UNIX operating system (Stallman 1993). In most implementations of incremental searches, as with the delimited search, the user first summons a dialog box that contains a field in which the user can enter the pattern. When he types the first character of the pattern, however, the system uses this character alone as a complete pattern and immediately begins to search for the first instance of that character in the chosen search direction. If an instance of that first character is found before the next character of the pattern is typed, the instance is selected and the cursor placed immediately after the end of the selection. If the next character of the pattern is typed before an instance is found, that character is added to the pattern and the search continues, now looking for an instance of the now extended pattern. This method is repeated as each character is added to the pattern.

EX1021 (Raskin), 125
'035 Paper 21 (POR), 19
'490 Paper 21 (POR), 19
'139 Paper 21 (POR), 19

Petitioner admits Wilcox and Londergan do not disclose incremental search across documents and instead relies on Raskin.

Raskin's search tool

- **scans through the characters in the text of**
- **a single, open document**
- **moves the cursor to the first instance of the search string**

Raskin's search tool does not

- **incrementally search for and locate documents**
- **return search results**

Petitioner Has Not Established How to Combine Raskin with Wilcox to Achieve Incrementally Locating Documents

A POSITA would have been motivated to perform Lotus Notes' multi-string search incrementally for the reasons stated by Raskin: speed, effectiveness, and ease of use. EX1021, 125-127. Indeed, 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. EX1021, 127 ("[B]uilding a Boolean search pattern is made more effective when the early results of the pattern appear as the user adds to the pattern's specificity."). EX1003, ¶75.

'035 Paper 1 (Petition), 25; '490 Paper 1 (Petition), 19; '139 Paper 1 (Petition), 23

A POSITA 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.

'035 Paper 1 (Petition), 25-26; '490 Paper 1 (Petition), 19-20; '139 Paper 1 (Petition), 23-24

- **Petitioner provides no explanation of how a POSITA would achieve the proposed combination**

Dr. Terveen Did Not Consider the Context of Raskin's Disclosures

Q So you would agree that this disclosure in Raskin, this is not a disclosure of incremental search across a set of documents?

MR. THACKER: Objection. Form.

A **Well, here 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. That is the teaching that I relied on for my opinion.**

EX2021 (Terveen Tr., Vol. I), 114:10-19
'035 Paper 21 (POR), 20
'490 Paper 21 (POR), 20
'139 Paper 21 (POR), 20

- **Dr. Terveen failed to consider whether Raskin's "idea of incremental search" by sequential scanning within the text of a single open document was applicable to the Lotus Notes indexed, multi-document search system**

Dr. Terveen Did Not Consider the Entirety of Raskin's Disclosures

Q When you prepared your initial declarations, did you review the entire Raskin reference?

MR. THACKER: Objection. Form. Scope.

A I don't recall if I reviewed it all. I obviously -- at the time when I prepared my initial declaration, the parts that I relied on for my opinion, I cited to.

EX2026 (Terveen Trans., Vol. II), 214:9-17

Q Did you consider whether any portion of the Raskin reference, aside from the pages that you actually cited, might teach away from incremental search across multiple documents?

A Not that I'm aware of, no.

EX2026 (Terveen Trans., Vol. II), 216:18-22

Petitioner Fails to Explain Why a POSITA Would Be Motivated to Combine With a Reasonable Expectation of Success Raskin's Single-Document, Sequential Search in the Multi-Document, Indexed Lotus Notes System

The Petition's Motivation to Combine Raskin and Wilcox/Londergan

> A POSITA would have been motivated to perform Lotus Notes' multi-string search incrementally for the reasons stated by Raskin: speed, effectiveness, and ease of use. EX1021, 125-127. Indeed, Raskin specifically points to the type of search

'035 Paper 1 (Petition), 25; EX1003 (Terveen Decl.), ¶75
'490 Paper 1 (Petition), 19; EX1003 (Terveen Decl.), ¶78
'139 Paper 1 (Petition), 23; EX1003 (Terveen Decl.), ¶75

- **Raskin's benefits are limited to scanning within a single document**
- **Neither the Petition nor Dr. Terveen explain how the combination improves the speed, effectiveness, or ease of use of searching within Lotus Notes**

Petitioner's Conclusory Statement of Expectation of Success

A POSITA 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. *E.g.*, *Keynetik, Inc. v. Samsung Elecs. Co.* No. 2022-1127, 2023 WL 2003932, at *2 (Fed. Cir. Feb. 15, 2023) ("[O]nce the function to be performed by software has been identified, writing code to achieve that function is within the skill of the art."). EX1003, ¶77.

'035 Paper 1 (Petition), 25-26; '490 Paper 1 (Petition), 19-20 ; '139 Paper 1 (Petition), 23-24

- **Petitioner relies on Dr. Terveen's conclusory statement**
- **Dr. Terveen's "simple and straightforward" opinion is rebutted by Mr. Haeberli's testimony**

In *Keynetik*, we held that expert testimony stating that certain software modifications were "straightforward" and "simple" was sufficient to establish a reasonable expectation of success. 2023 WL 2003932, at *1–2. However, in *Keynetik* there was no challenge to the expert's testimony as being conclusory or not credible, and the expert testimony was uncontradicted.

Juniper Networks, Inc. v. Correct Transmission, LLC, No. 2023-1046, 2024 WL 3517862, at *4 (Fed. Cir. July 24, 2024)

Patent Owner Established These Motivations Fail

A POSITA would have been motivated to perform Lotus Notes' multi-string search incrementally for the reasons stated by Raskin: speed, effectiveness, and ease of use. EX1021, 125-127. Indeed, Raskin specifically points to the type of search

'035 Paper 1 (Petition), 25; EX1003 (Terveen Decl.), ¶75
'490 Paper 1 (Petition), 19; EX1003 (Terveen Decl.), ¶78
'139 Paper 1 (Petition), 23; EX1003 (Terveen Decl.), ¶75

122. As I explained above regarding the Lotus Notes-Raskin combination, implementing incremental search in a multi-document search system would require either an increased index size, which would slow down searching or exceed the available computing power of a typical consumer computer at the time of the invention, or would require sequential search across all documents, which would be so slow as to no longer function as incremental search at all. (*See supra* §V.A.2.)

'035 EX2020 (Supp. Haeberli Decl.), ¶117
'490 EX2020 (Supp. Haeberli Decl.), ¶87
'139 EX2020 (Supp. Haeberli Decl.), ¶122

- **Two options for Petitioner's alleged combination:**
  - **Sequential scanning across all documents**
  - **Increasing index size**

Both are slow, cumbersome and would lead to an ineffective search tool

Implementing Raskin's Sequential Scanning in the Wilcox/Londergan System Would Be Slow

Patent Owner's Expert – Mr. Haeberli

52. The technical problem with this approach is one of scale. A POSITA applying the "incremental search" methodology of Raskin to Lotus Notes' search systems would understand that the sheer volume of material that would need to be sequentially searched would be so large that the search would slow down dramatically, to the point that it would cease to be incremental (providing search results as the user types) at all. Dr. Terveen himself agrees with this assessment,

'035 EX2020 (Supp. Haeberli Decl.), ¶52
'490 EX2020 (Supp. Haeberli Decl.), ¶52
'139 EX2020 (Supp. Haeberli Decl.), ¶52

- **Combination would be slow, cumbersome, and ineffective**

Implementing Raskin's Sequential Scanning in the Wilcox/Londergan System Would Be Slow

Petitioner's Expert – Dr. Terveen

148. Thus, rather than scanning every word of every file sequentially, which a POSITA knew takes more time and consumes more computing resources, a POSITA would have understood or at least found it obvious that the computing system indexes the computing device's files so that the search process (e.g., Entourage's search process) is carried out more quickly and efficiently, as was well known in the art. *E.g.*, EX1023, 6, 191. Specifically, a POSITA would have been

'035 EX2023 (Suppl. Terveen Decl.), ¶148
'490 EX2023 (Suppl. Terveen Decl.), ¶148
'139 EX2023 (Suppl. Terveen Decl.), ¶148

- **Raskin's sequential scanning is slower and "consumes more computing resources" than using an indexed search system**

The Combination Requires Increased Index Size

Patent Owner's Expert – Mr. Haeberli

53. Alternatively, one might attempt to increase the size of the index in the Lotus Notes system to accommodate all of the potential partial strings included within each word in every document in the database. However, this approach faces its own technical challenges.

'035 EX2020 (Supp. Haeberli Decl.), ¶53
'490 EX2020 (Supp. Haeberli Decl.), ¶53
'139 EX2020 (Supp. Haeberli Decl.), ¶53

The Combination Requires Increased Index Size

Petitioner's Expert - Dr. Terveen

17 Q So in order to implement incremental
18 search in an indexed system, would you agree
19 that the index would necessarily be larger than
20 in a non-incremental search system?
21 MR. THACKER: Objection. Form.
22 **A To enable incremental searching, the**
1 index must be able to allow searching on
2 characters as you go. So that would require,
3 you know, a different -- how would you put it --
4 more information or different representation of
5 information in the index.
6 Q Right, because I think what you're
7 saying is you would need all of the little
8 prefixes, right?
9 A You would need a way for the search
10 engine to be able to search by prefixes, yes, to
11 search incrementally.

EX2021 (Terveen Trans., Vol. I), 104:17-105:11

- **To enable incremental search requires indexing prefixes, which increases index size as compared to an index for delimited search**

Larger Index Leads to Slower Searches

Patent Owner's Expert – Mr. Haeberli

33. From a technical standpoint, the size of the index has a direct and significant impact on search performance—specifically, larger indexes result in slower search operations. (*See* EX2022, 6 ("To improve performance, we built a reduced version of the index"); EX2021, 38:1-39:17.) This relationship exists because larger indexes consume more processing power and require greater amounts of available RAM to operate efficiently. (*See* EX2021, 39:15-16 ("if you add more information, you will be taking more memory"); EX1059, 169 (explaining that including case-sensitive entries in the index increases the size by 25%);

'035 EX2020 (Supp. Haeberli Decl.), ¶33
'490 EX2020 (Supp. Haeberli Decl.), ¶33
'139 EX2020 (Supp. Haeberli Decl.), ¶33

- **For efficient searching, the index should be loaded into the RAM**
- **The larger the index, the more RAM used**
- **As RAM usage increases, performance degrades**

- **Larger index = slower search**

49. This level of resource consumption could lead to general performance degradation across the entire computer system, and could even result in search operations lagging to such an extent that the system would not be able to provide search results as the user types, meaning it would no longer function as an incremental search at all.

'035 EX2020 (Supp. Haeberli Decl.), ¶49
'490 EX2020 (Supp. Haeberli Decl.), ¶49
'139 EX2020 (Supp. Haeberli Decl.), ¶49

Larger Index Leads to Slower Searches

Petitioner's Expert (Dr. Terveen) Agrees

12 Q Would you also agree that increasing
13 the size of an index increases the amount of
14 memory that is used?
15 MR. THACKER: Objection. Form.
16 A Well, assuming – with the assumption
17 that you hold everything else equal, that you
18 have an index and you simply make that index
19 larger, it could increase the time it took to
20 search it, but it would also depend on how that
21 index is structured. So there could be
22 increases in size that you would make that would
1 lead to little or no increase in search time.
2 So, you know, there are factors that
3 you would have to consider to really answer that
4 question precisely.

EX2021, (Terveen Trans., Vol. I) 38:12-39:4
'035 Paper 21 (POR), 19
'490 Paper 21 (POR), 19
'139 Paper 21 (POR), 19

- **Larger index = slower search**

problem. EX1001, 7:26-37. This would result in an even larger index, there is no discussion of how exactly that index would be stored, and there certainly is no explanation as to how that index would provide results quick enough to perform an incremental search when searching across a large volume of files/documents.

'035 EX1100 (Supp. Terveen Decl.), ¶52
'490 EX1100 (Supp. Terveen Decl.), ¶52
'139 EX1100 (Supp. Terveen Decl.), ¶52

Petitioner's Shifting Motivation

EX2020, ¶¶8-9, 47-62, 114-118. The challenged claims, however, recite searching across a "group" of documents (i.e., two or more), not a large volume. Moreover, 2002-era computers were entirely capable of incrementally searching across a group of documents, especially considering that the median size of, for example, two emails in 2002 was approximately 4.4 kilobytes—0.00044% of the one gigabyte database assumed by Mr. Haeberli. EX1099, 190; EX1100, ¶¶13-32. Indeed, several

'035 Paper 27 (Reply), 4; EX1100 (Terveen Supp. Decl.), ¶13
'490 Paper 28 (Reply), 4; EX1100 (Terveen Supp. Decl.), ¶13
'139 Paper 27 (Reply), 4; EX1003 (Terveen Supp. Decl.), ¶13

Acknowledging the deficiencies in its motivation, Petitioner pivots to focus justifying their motivation for a system limited to two documents

- **Petitioner's lead references (Lotus Notes and Entourage references) describe systems designed to work with large sets of documents**
- **A POSITA would not have been motivated to limit the functionality of Lotus Notes to two documents**

The Wilcox-Londergan-Raskin-Wu Combination Does Not Render Obvious the Displaying and Highlighting Claims

Claims 2-4 of the ’035 Patent Recite Displaying and Highlighting

2. The method of claim **1**, further comprising displaying a portion of a selected document of the documents that are included in each of the first and second groups of documents.

3. The method of claim **2**, further comprising automatically highlighting a first occurrence of the first word in the selected document.

4. The method of claim **2**, further comprising highlighting at the same time a plurality of occurrences of the first word in the selected document while not highlighting portions of text between occurrences of the first word.

“displaying a portion of a selected document”

“highlighting”

‘035 patent (EX1001), claims 2-4

Claims 3-5 and 24-26 of the ’139 Patent Recite a “User Interface”

3. The method of claim **1**, further comprising:
displaying in a user interface an indication of the identified one or more documents included in both the first one or more documents and the second one or more documents.

4. The method of claim **3**, further comprising displaying via the user interface a portion of a selected identified document.

5. The method of claim **4**, further comprising highlighting a plurality of occurrences of the first and second partial search strings in the selected document while not highlighting portions of text between occurrences of the first and second partial search strings in the selected document.

“user interface”

“highlight … partial search strings”

“displaying a portion of a selected identified document”

‘139 patent (EX1001), claims 3-5

24. The computing system of claim **23**, wherein executing the software further causes the computing system to:
display, on a display device of the computing system, a user interface including an indication of the identified one or more documents included in both the first one or more documents and the second one or more documents.

25. The computing system of claim **24**, wherein executing the software further causes the computing system to:
display via the user interface a portion of a selected identified document.

26. The computing system of claim **25**, wherein executing the software further causes the computing system to:
highlight in the user interface a plurality of occurrences of the first and second partial search strings in the selected document while not highlighting portions between occurrences of the first and second partial search strings in the selected document.

‘139 patent (EX1001), claims 24-26

Claims 1, 10-11 of the ’490 Patent Recite an “Interactive User Interface”

1. A computing system comprising:
one or more hardware computer processors;
a display device viewable by a user of the computing system;
a computer readable storage medium storing:
an index data structure comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included; and
software instructions executable by the one or more hardware computer processors causing the computing system to:
cause display of an interactive user interface on the display device, the interactive user interface including at least a first search field;

‘490 patent (EX1001), claim 1

“interactive user interface”

“display a portion of the first file”

10. The computing system of claim 9, wherein the software instructions further cause the computing system to:
receive selection of a first file of the displayed set of matching files; and
display in the interactive user interface a portion of the first file.

11. The computing system of claim 10, wherein the software instructions further cause the computing system to:
highlight a plurality of occurrences of the first and second partial search strings in the first file while not highlighting portions of text between occurrences of the first and second partial search strings in the first file.

‘490 patent (EX1001), claims 10-11

“highlight partial search strings”

Dispute Turns on Plain Meaning of "User Interface"

- **Petitioner did not request construction of "user interface" in the Petition**
  - Petitioner's alleged combination requires ***opening*** a document to view a portion of the document or see the search terms highlighted
- **In Reply, Petitioner newly argues that a "user interface" in the abstract could include multiple windows**
- **This is inconsistent with how a POSITA would understand "user interface" in the context of the challenged claims**

Petition Mischaracterizes Wilcox and Londergan's Displaying and Highlighting Capabilities

The Lotus Notes+Raskin+Wu combination renders these claims obvious at least because the Lotus Notes references disclose displaying a portion of a selected document and highlighting occurrences of only the searched strings. EX1059, 164 ("The found text [] is indicated by a red box."), 165; EX1061, 145, 146 ("Notes [] shows you the document and highlights the text that you wanted to find.").

'035 Paper 1 (Petition), 34; '490 Paper 1 (Petition), 35-36; '139 Paper 1 (Petition), 34

You open a document uncovered by a search the same way that you open any other document: Double-click it or select it with your mouse and press Enter. Notes then shows you the document and highlights the text that you wanted to find. As you read a document retrieved by a full-text search, you can jump quickly to the next occurrence of the search text by pressing Ctrl++ (that's the Control key and the plus sign). You use Ctrl+– (Control and the minus sign) to jump quickly to the preceding occurrence of the word.

EX1061 (Londergan), 146
'035 Paper 21 (POR), 32; '490 Paper 21 (POR), 32, 34; '139 Paper 21 (POR), 32

Petitioner: selectively quotes Londergan to assert that Lotus Notes "displays a portion" of the document and highlights it as part of the search

Londergan:

- **user must open the document in a separate window and/or application to display the document and highlight search strings**
- **highlighting is not done as part of the search**

Petitioner Improperly Gap-Fills with Knowledge of a POSITA

> Moreover, it would have been obvious to display a portion of a selected document because it was well known in the art to do so to provide the user with more information about each of the search results. *Supra* Section IV.B.3. EX1003, ¶93.

'035 Paper 1 (Petition), 35; '490 Paper 1 (Petition), 35-36; '139 Paper 1 (Petition), 34-35, 45

> document and highlights the text that you wanted to find."). Moreover, a POSITA would have understood or at least found it obvious to highlight a plurality of occurrences of the search strings to point the user to more (e.g., all) of the relevant portions of the document, as was well known in the art. *See supra* Section VI.B.3.

'035 EX1003 (Terveen Decl.), ¶ 96
'490 EX1003 (Terveen Decl.), ¶¶ 112, 192
'139 EX1003 (Terveen Decl.), ¶¶97, 117, 172, 191

- **Petitioner and Dr. Terveen fall back on knowledge of a POSITA without pointing to where these elements are found in the prior art relied upon**

Petitioner Improperly Advances a New Claim Construction Argument for "User Interface" in its Reply

For the highlighting claims, the POR also attempts to distinguish the prior art based on an improper implicit claim construction argument. Though the claims recite highlighting search strings in the same user interface as the search results, the POR improperly reads in a requirement that they be displayed in the same window. *E.g.*, EX1001, 29:56-30:28 (claim 1); Paper 21, 32-35, 49, 66. Consistent with how a POSITA would have understood the term "user interface," Haeberli admitted that a user interface can include multiple windows, not just one. EX1098, 18:19-20:5;

'035 Paper 27 (Reply), 24
'490 Paper 28 (Reply), 24
'139 Paper 27 (Reply), 24

<u>Petitioner</u>: relies solely on extrinsic evidence, expert testimony taken out of context and divorced from the context of the patents

The Patents Describe a Unified User Interface

The preview pane is in the same window as the search field

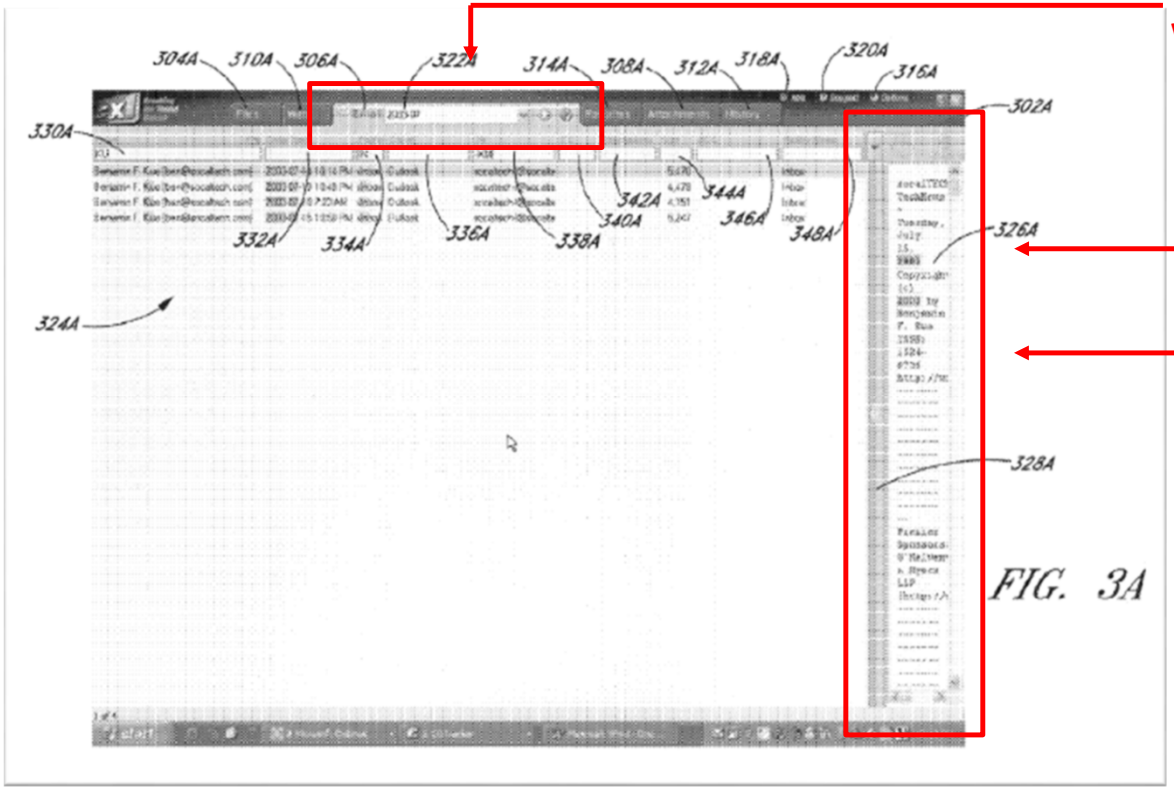


FIG. 3A

In addition, a view area or pane **326**A is provided that displays the content of the first item or user-selected item from the list pane or area **324**A. The user can print the search results displayed in the list pane or area **324**A, as well as the document, email, Web page, file, and the like, displayed in the view pane or area **324**A.

'035 Patent (EX1001), 14:31-36
'490 Patent (EX1001), 14:43-48
'139 Patent (EX1001), 14:37-42
'035 Paper 21 (POR), 31
'490 Paper 21 (POR), 33
'139 Paper 21 (POR), 31

'035 Patent (EX1001), Figure 3A; '490 Patent (EX1001), Figure 3A; '139 Patent (EX1001), Figure 3A
'035 Paper 21 (POR), 32; '490 Paper 21 (POR), 33; '139 Paper 21 (POR), 32

Patents Distinguish Between "Displaying" and Opening Documents

As illustrated in FIG. **3**A, the search results are displayed in a list area or pane **324**A, which includes a list of the search results, including relevant attributes in column format. In the example illustrated in FIG. **3**A, the attributes include a from (sender) attribute column, a to (addressee) attribute column, date/time sent and/or received attribute columns, a folder path attribute column, a client attribute column, a cc attribute column, a size attribute column, a bcc attribute column, and a subject field. In this example, each of the columns has a corresponding search field **330**A-**348**A.

In addition, a view area or pane **326**A is provided that displays the content of the first item or user-selected item from the list pane or area **324**A. The user can print the search results displayed in the list pane or area **324**A, as well as the document, email, Web page, file, and the like, displayed in the view pane or area **324**A.

Further, when the user clicks on a list pane entry, the corresponding contents are displayed in the view pane **326**A. If the user double-clicks, or presses the Enter button, on the list pane entry, the search application causes the entry to launch in its native application. For example, if the user double-clicks on a list pane entry corresponding to a Word file, the file will open in Word. Similarly, if the user

'035 Patent (EX1001), 14:21-48; '035 Paper 21 (POR), 31-32
'490 Patent (EX1001), 14:33-60; '490 Paper 21 (POR), 33;
'139 Patent (EX1001), 14:27-54; '139 Paper 21 (POR), 31-32

3. The method of claim **1**, further comprising:
displaying in a user interface an indication of the identified one or more documents included in both the first one or more documents and the second one or more documents.

4. The method of claim **3**, further comprising displaying via the user interface a portion of a selected identified document.

5. The method of claim **4**, further comprising highlighting a plurality of occurrences of the first and second partial search strings in the selected document while not highlighting portions of text between occurrences of the first and second partial search strings in the selected document.

'139 Patent (EX1001), claims 3-5

The specification describes a unified user interface:

- **Search bar**
- **List pane displays search results**
- **View pane displays a portion of the selected item**
- **Double-clicking an item causes it to open**

No Obviousness of Display/Highlighting Claims Under Plain and Ordinary Meaning

- In Petitioner's combination, the document must be opened to meet the display and highlighting claims
- Under the correct plain and ordinary meaning, the "user interface" is a single window with a preview pane
- Opening a document is fundamentally different from "display[ing] [in/via] the [interactive] user interface a portion of the [selected identified document/first file]" ('193, 4/'490, 10) / "displaying a portion of a selected document" ('035, 2)

The Wilcox-Londergan-Raskin-Wu Combination Does Not Render Obvious the Field Claims

The Field Claims Require Searching on Specific Fields

Searching email fields:

- Claims 7 and 9 of the ’035 patent
- Claims 5-7 and 15-19 of the ’490 patent
- Claims 8, 13, and 18 of the ’139 patent

Searching contact and calendar fields:

- Claims 10-11, 15-16, and 20-21 of the ’139 patent

Searching file/document fields:

- Claim 8 of the ’035 patent
- Claims 4 and 14 of the ’490 patent
- Claims 9, 12, 14, 17, and 19 of the ’139 patent

7. The method of claim **1**, wherein at least some of the documents comprise at least the following email fields in which one or more of the first and second strings may be located: to, from, cc, bcc, subject, body.

’035 patent (EX1001), claim 7

10. The method of claim **1**, wherein at least some of the documents comprise at least the following contacts fields in which one or more of the first and second partial search strings may be located: name, email address, phone number, company, job title, or content.

’139 patent (EX1001), claim 10

14. The computing system of claim **1**, wherein the first search field corresponds to a single field of the files and matching files are identified based only on content of the single field of the files.

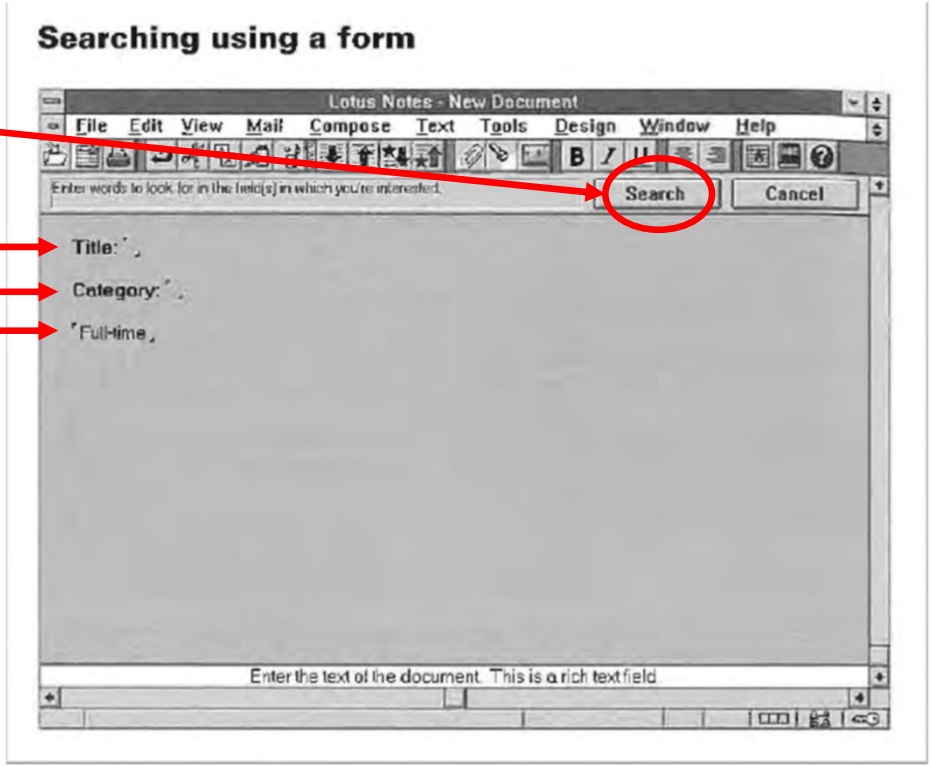
’490 patent (EX1001), claim 14

Petitioner Fails to Establish Obviousness of the Field Claims

- Petitioner fails to articulate a sufficient motivation to combine the Searching by Form tool with Query Builder or Search Bar
- No disclosure of using the Searching by Form functionality to search email fields
- Petitioner improperly gap-fills with knowledge of a POSITA

Petitioner Relies on a Different Search Tool From Wilcox and Londergan – Search by Form – to Show These Limitations

Searching by Form



Delimited search button

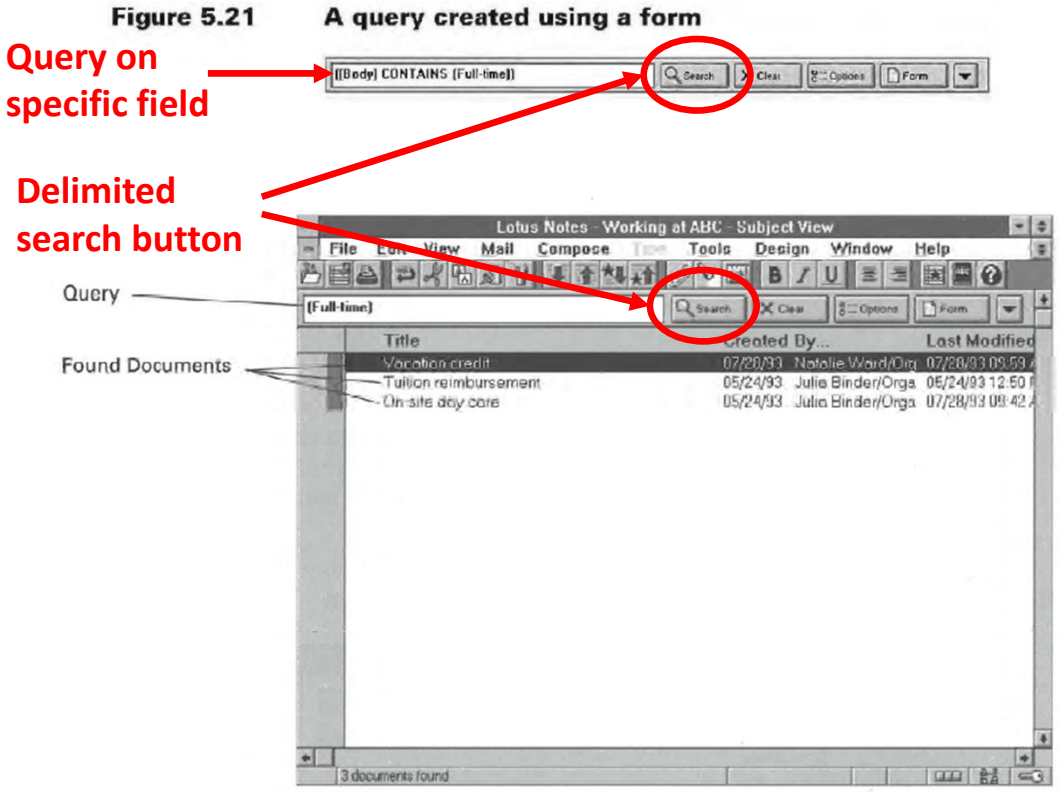
Single search string in box for field to be searched

Limited to searching documents/records created using that specific form

EX1059 (Wilcox), 168
'035 Paper 1 (Petition), 20
'490 Paper 1 (Petition), 15
'139 Paper 1 (Petition), 18

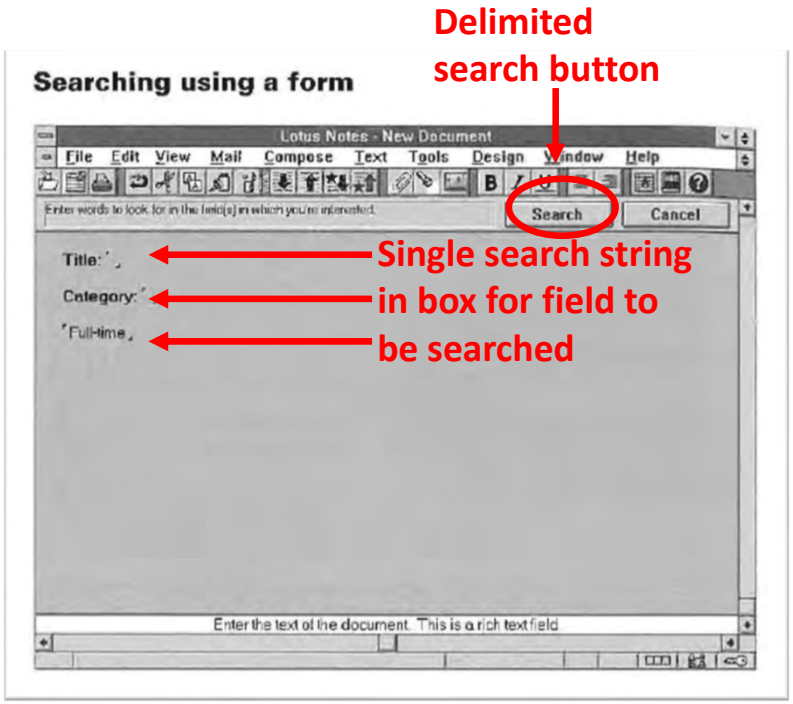
Petitioner Fails to Articulate Motivation to Combine Searching by Forms With Query Builder Which is Used for Claim 1

Searching by Form via Search Bar (Claim 1)



EX1059 (Wilcox), 164, 168

Searching by Form (Field Claims)



EX1059 (Wilcox), 168
'035 Paper 1 (Petition), 20
'490 Paper 1 (Petition), 15
'139 Paper 1 (Petition), 18

None of Petitioner's Motivations Provide The Requisite Showing for a Motivation to Combine

A POSITA would have been motivated to combine the complementary teachings on Lotus Notes from Londergan (EX1061) and Wilcox (EX1059) to obtain a more complete and comprehensive understanding of the Lotus Notes system, its features, and its capabilities. *See RPX Corp. v. Rothschild Connected Devices*

'035 Paper 1 (Petition), 24
'490 Paper 1 (Petition), 18
'139 Paper 1 (Petition), 22

Petition: no mention of combining different search tools; no reasons why it would be favorable to do so

field separated by a string separator character (a comma). Paper 21, 29-30. While these similar disclosures for the Search Bar and Search Builder tools certainly reinforce the obviousness of the claims, they are not necessary.[3]

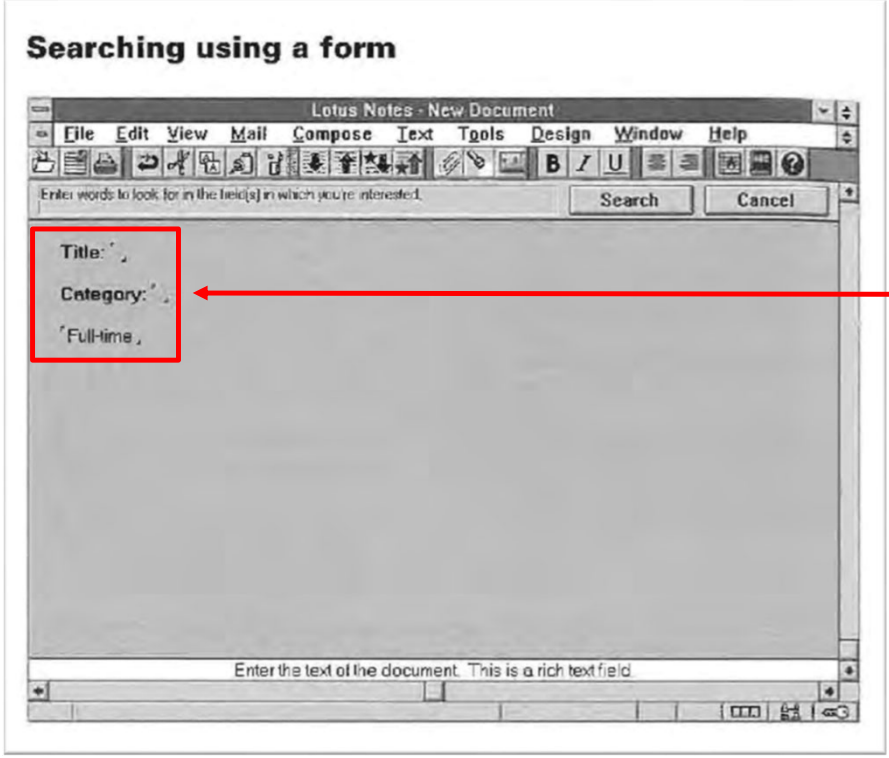
Moreover, combining these functionalities in a single tool was desirable and/or obvious to try. Paper 1, 10, 23-25, 27-29; EX1100, ¶55.

'035 Paper 27 (Reply), 17
'490 Paper 28 (Reply), 16
'139 Paper 27 (Reply), 17

Reply: new argument to combine search tools

- No explanation of why it was desirable
- Failed to meet the legal requirements for obvious to try

Search by Form Cannot Search Emails, Contacts, or Calendar Events



Neither Wilcox nor Londergan indicate that Search by Form can be used to search emails, contacts, or calendar events. The "Title" and "Category" fields do not correspond to these types of files.

EX1059 (Wilcox), 168
'035 Paper 21 (POR), 13
'490 Paper 21 (POR), 13
'139 Paper 21 (POR), 13

Petitioner Improperly Gap-Fills with Knowledge of a POSITA

Exemplary Gap-Filling: Dr. Terveen's opinion for claims 12-16 of the '139 patent:

104. As explained above for claims 8-11 (*supra* Sections VIII.E.7-VIII.E.10), the Lotus Notes references disclose the ability to search within a single field of the documents (such that the incremental searching is performed only on content of that field of the documents) and disclose that the documents comprise the fields recited in claims 12-16. EX1059, 93-146, 159-168, 321, 323; EX1061, 21-73, 141-144, 146-147, 150-153, 160-161, 186.

105. Thus, claims 12-16 are obvious for at least the reasons claims 8-11 are obvious. *Supra* Sections VIII.E.7-VIII.E.10. Moreover, a POSITA would have found it obvious to provide this targeted search ability, given that it was well known in the art. *Supra* Section VI.B.4. Thus, the Lotus Notes+Raskin+Wu combination renders these claims obvious.

'035 EX1003 (Terveen Decl.), ¶100-101; '490 EX1003 (Terveen Decl.), ¶115-116
'139 EX1003 (Terveen Decl.), ¶104-105

Petitioner and Dr. Terveen fail to articulate a combination that renders obvious the claims or a motivation to make a combination:

- **The existence of the recited fields alone does not render obvious the claimed limitations of incrementally searching for and locating groups of documents using multiple string entered into a single search field corresponding to a single field**
- **Knowledge of a POSITA and/or "background art" not identified with particularity**

Petitioner's Articulated Motivation to Combine the Search Tools is Insufficient for Claim 17 of the '139 Patent

17. The method of claim **1**, wherein the first partial search string applies to all textual content of the set of documents and the incremental searching of the second partial search string is performed only on content of a single field of documents within the set of documents.

'139 patent (EX1001), claim 17

Petitioner argues that Wilcox and Londergan's "default" is a full-text search, presumably referring to Query Builder, Search Builder, and/or Search Bar.

Petitioner then separately relies on the Search by Form functionality as allowing a search across a single field.

Petitioner fails to articulate any motivation to combine these search tools.

Claim 4 of the ’490 Patent

4. The computing system of claim 1, wherein the index data structure comprises indications of character strings associated with respective attribute fields of the files.

’490 patent (EX1001), claim 4

Petitioner points to no disclosures in the cited art of the index structure disclosed in this claim limitation

Petitioner impermissibly relies on knowledge of a POSITA

The Wilcox-Londergan-Raskin-Wu Combination Does Not Render Obvious an Index that Facilitates Incremental Search of Partial Strings

The ’490 Patent Claims Require an Index Capable of Facilitating Incremental Search of Partial Strings

1. A computing system comprising:
one or more hardware computer processors;
a display device viewable by a user of the computing system;
a computer readable storage medium storing:
an index data structure comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included; and
software instructions executable by the one or more hardware computer processors causing the computing system to:
cause display of an interactive user interface on the display device, the interactive user interface including at least a first search field;
receive a first partial search string in the first search field;
receive a second partial search string in the first search field;
wherein the first and second partial search strings are separated by a string separator character in the first search field;
searching the index data structure for a first one or more of the plurality of files each including at least one string that corresponds to the first partial search string;
searching the index data structure for a second one or more of the plurality of files each including at least one string that corresponds to the second partial search string; and
updating a set of matching files of the plurality of files as each successive character of the first partial search string and the second partial search string is received respectively in the first search field by identifying any of the plurality of files included in both the first one or more of the plurality of files and the second one or more of the plurality of files, wherein the first partial search string and the second partial search string are non-adjacent in at least one of the matching files.

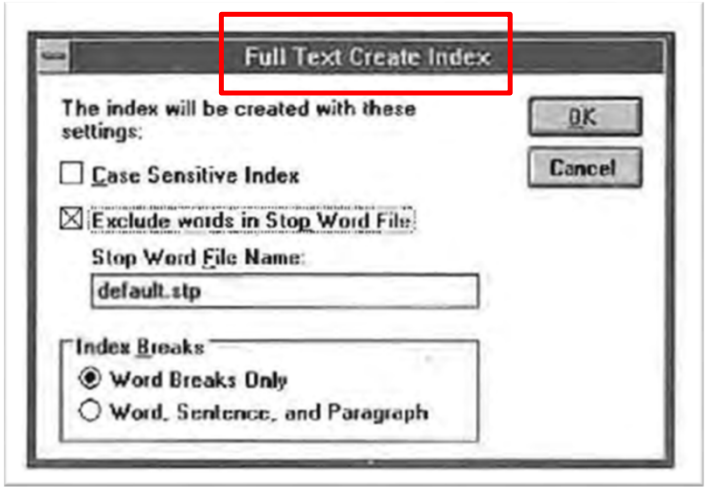
’490 patent (EX1001), claim 1

Petitioner First Relies on Wilcox and Londergan's "Full-Text" Index, Which Does Not Index Partial Strings or Otherwise Allow for Partial String Search

The Lotus Notes references expressly disclose indexing the computing system's files in an index data structure (a full-text index) comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included. EX1059, 137-138, 159-160, 169-171; EX1061, 41, 135-139. For example, Lotus Notes describes that "to support full-text searches, [the database] must have a current *full-text index*" in which "all the documents in the database are indexed." EX1059, 159-160. A POSITA would have understood that the full-text index comprises associations between each of a plurality of character strings to the associated files in which the character strings are included at least because a POSITA knew that this is how indexes work and because the Lotus Notes references expressly disclose that the full-text indexes "allow you to find documents that contain various combinations of *text strings* (a text string is any combination of characters; it may or may not be a word)." EX1059, 159; *supra* Section IV.B.3. EX1003, ¶84.

'490 Paper 1 (Petition), 21

A full-text index only includes each character string *as it appears in the document*. Petitioner offers no explanation how such an index could identify associations to partial strings that do not independently appear in the document.



EX1059 (Wilcox), 170
'490 Paper 1 (Petition), 21

Petitioner Improperly Relies on the Supposed Knowledge of a POSITA that "This is How Indexes Work", Citing a Conclusory Statement from Dr. Terveen

Petition

The Lotus Notes references expressly disclose indexing the computing system's files in an index data structure (a full-text index) comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included. EX1059, 137-138, 159-160, 169-171; EX1061, 41, 135-139. For example, Lotus Notes describes that "to support full-text searches, [the database] must have a current *full-text index*" in which "all the documents in the database are indexed." EX1059, 159-160. A POSITA would have understood that the full-text index comprises associations between each of a plurality of character strings to the associated files in which the character strings are included at least because a POSITA knew that this is how indexes work and because the Lotus Notes references expressly disclose that the full-text indexes "allow you to find documents that contain various combinations of *text strings* (a text string is any combination of characters; it may or may not be a word)." EX1059, 159; *supra* Section IV.B.3. EX1003, ¶84.

'490 Paper 1 (Petition), 21

Dr. Terveen's Declaration

84. The Lotus Notes references expressly disclose indexing the computing system's files in an index data structure (a full-text index) comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included. EX1059, 137-138, 159-160, 169-171; EX1061, 41, 135-139. For example, Lotus Notes describes that "to support full-text searches, [the database] must have a current *full-text index*" in which "all the documents in the database are indexed." EX1059, 159-160. A POSITA would have understood that the full-text index comprises associations between each of a plurality of character strings to the associated files in which the character strings are included at least because a POSITA knew that this is how indexes work and because the Lotus Notes references expressly disclose that the full-text indexes "allow you to find documents that contain various combinations of *text strings* (a text string is any combination of characters; it may or may not be a word)." EX1059, 159. Moreover, it was well known in the art to use indexes to expedite the search process. *Supra* Section VI.B.3.

'490 EX1003 (Terveen Decl.), ¶84

Petitioner Does Not Identify Any Background Reference Disclosing an Index Structure that Could Accommodate Partial String Search

Petitioner Disclosed a New Theory on Reply

Petitioner's Reply

EX1018, EX1023, EX1061), ¶¶46-49. Further, a system that indexes only whole words inherently indexes prefixes because common words that would be indexed in any system (e.g., the word "a") are prefixes for other words (e.g., "an," "and," "Andrew," etc.). EX1098, 25:11-27:1; EX1100, ¶64.

'490 Paper 28 (Reply), 21-22

- **Petitioner's new theory, offered on Reply, is refuted by Dr. Terveen's testimony.**
- **Most prefixes are not "common words that would be indexed in any system" as proposed by Petitioner.**

Dr. Terveen's Testimony

20 Q And in a prefixing index type system,
21 if you were going to index "declaration," would
22 you agree that you would need to index D, DE,

212

1 DEC, DECL, and so on and so forth?

16 So if you mean -- again, if you mean
17 a system -- you're specifying a system where
18 literally the requirement is that every prefix
19 of every word must be somehow indexed, then you
20 would have to have a way to index, for
21 "declaration," D, DE, DEC, DECL and so on.
22 So however many that would be all the

214

1 way up to "declaration," yeah.

EX2026 (Terveen Trans., Vol. II), 211:20-212:1, 213:16-214:1
'490 Paper 38 (Sur-Reply), 26

All Challenged Claims are Patentable over the Schwartz, Sebastian, Barber, True, Baeza-Yates Combination

Instituted Grounds

Ground	References	Basis	’035 Patent Claims	’490 Patent Claims	’139 Patent Claims
2	Schwartz (EX1004), Sebastian (EX1005), Barber (EX1006), True (EX1009), and Baeza-Yates (EX1023)	35 U.S.C. § 103	1-10	1-20	1-27

Petitioner's Picking and Choosing from Multiple References

Exemplary Claim 1, '035 Patent

1. A method of performing a search comprising:
receiving a first string in a first search field;
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;
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;
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
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.

'035 (EX1001), claim 1

Entourage's List Filtering Tool

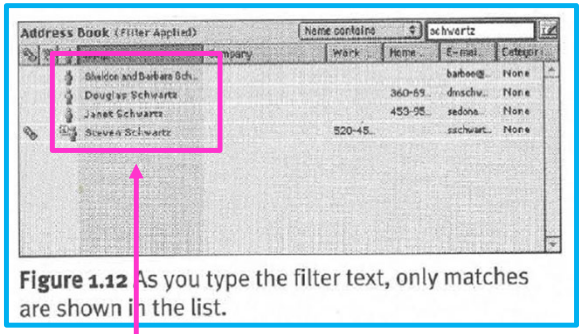
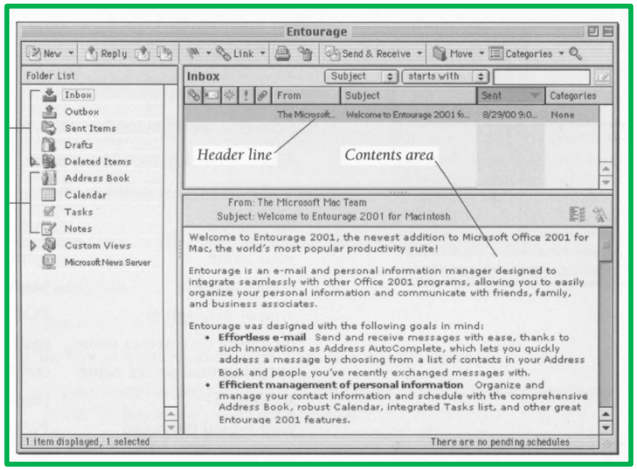


Figure 1.12 As you type the filter text, only matches are shown in the list.

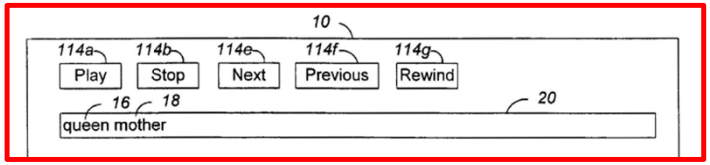
EX1004 (Schwartz), 9

Entourage's Delimited Find Tool



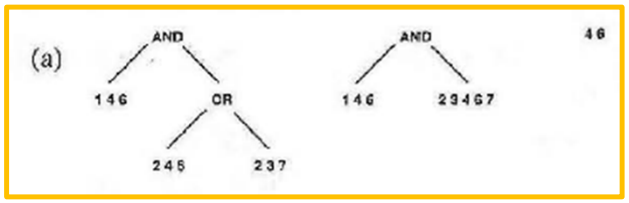
EX1005 (Sebastian), 173

True's Multiple Strings



EX1009 (True), Fig. 2

Baeza-Yates' Intersection Approach



EX1023 (Baeza-Yates), 208

Petitioner Fails to Establish Obviousness Over the Schwartz, Sebastian, Barber, True, and Baeza-Yates Combination

None of the five references, alone or in combination, discloses or renders obvious incrementally searching for and locating groups of documents

- **Petitioner has not articulated any reasoning with rationale underpinnings for motivation to combine the distinct Entourage search tools disclosed in Schwartz, Sebastian, and Barber**
- **Petitioner has not established motivation to combine the Entourage search tools with True**
- **The combination fails to disclose all limitations of the claims**
- **True is not analogous art**
- **Petitioner improperly gap-fills with knowledge of a POSITA for certain dependent claims**

Petitioner Has Not Shown Motivation to Combine With a Reasonable Expectation of Success for the Three Distinct Tools in Schwartz, Sebastian, and Barber

Petitioner's Articulated Motivation to Combine the Distinct Tools in the Entourage References is Insufficient

A POSITA would have been motivated to combine the teachings of the three Entourage references (Schwartz, Sebastian, and Barber) at least because they describe the same Microsoft email system—Entourage 2001 for Macintosh. *E.g.*, EX1004, (i); EX1005, (a)-(c), 171-188; EX1006, (i), 1-3, 5, 9, 283-392. *See RPX Corp*. IPR2016-00443, Paper 8 at 21 (PTAB Jul. 7, 2016), Paper 28 at 35 (PTAB Jul. 6, 2017). Specifically, a POSITA 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, especially since each reference provides additional teachings not found in the other references.

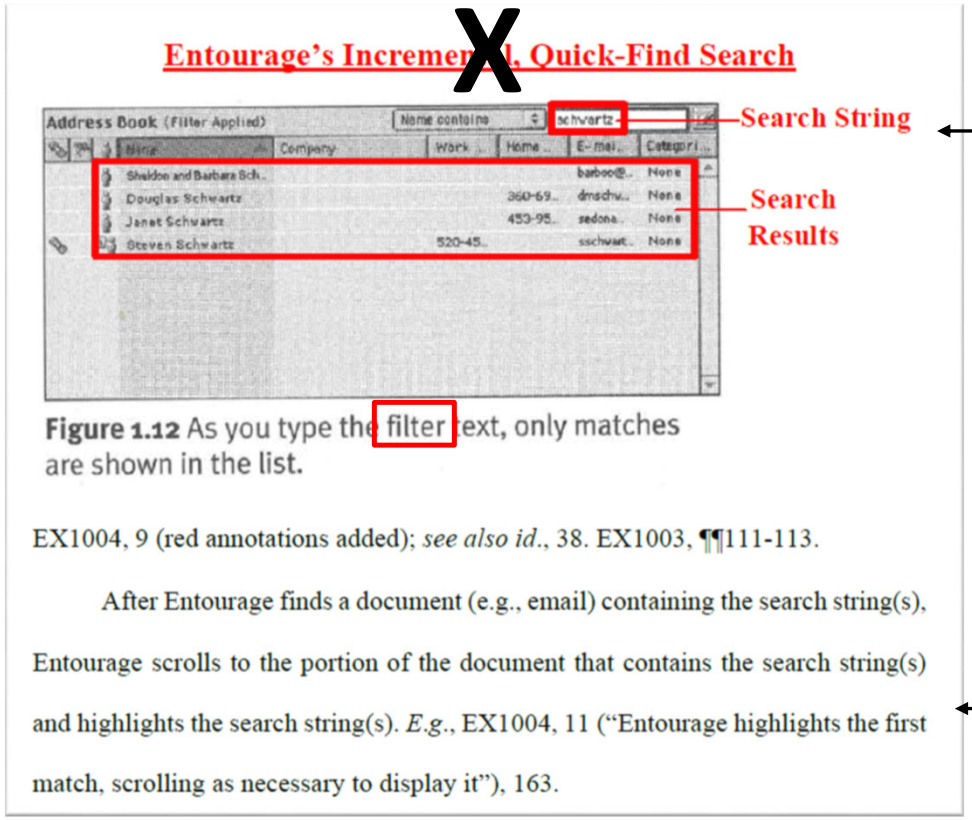
'035 Paper 1 (Petition), 45-46
'490 Paper 1 (Petition), 48-49
'139 Paper 1 (Petition), 51-52

Petitioner never clearly articulates what features and tools it is relying on for its combination

Petitioner has provided no motivation to combine the three distinct tools of Entourage

Petitioner has provided no details as to how the tools could be combined to achieve the claimed invention

Petitioner Mischaracterizes Entourage's Distinct Tools as if They Were One Search Tool



EX1004, 9 (red annotations added); *see also id.*, 38. EX1003, ¶¶111-113.

After Entourage finds a document (e.g., email) containing the search string(s), Entourage scrolls to the portion of the document that contains the search string(s) and highlights the search string(s). *E.g.*, EX1004, 11 ("Entourage highlights the first match, scrolling as necessary to display it"), 163.

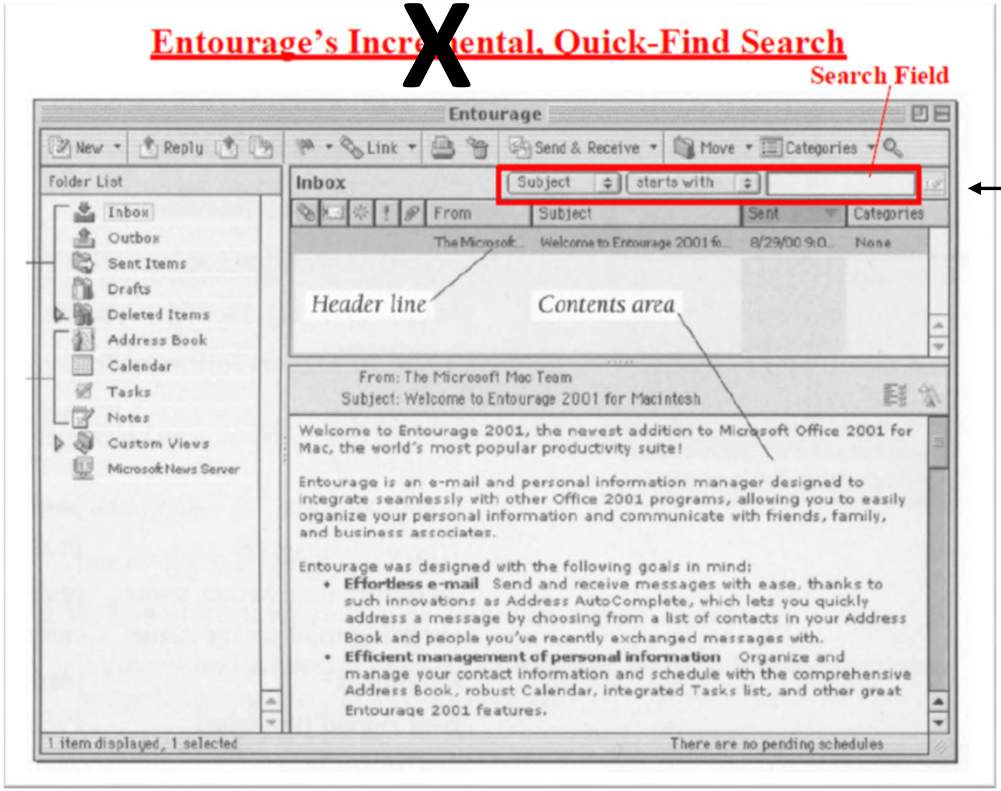
List Filtering Tool

- **Petitioner mischaracterizes the distinct tools as being one tool**
- **Petitioner misleadingly labels a figure showing the List Filtering Tool as if it were the Find Tool and then describes alleged functionality of the Find Tool as though that is what the figure depicts.**

Find Tool

'035 Paper 1 (Petition), 44; '490 Paper 1 (Petition), 47;
'139 Paper 1 (Petition), 50

Petitioner Mischaracterizes the Delimited Find Tool as Incremental

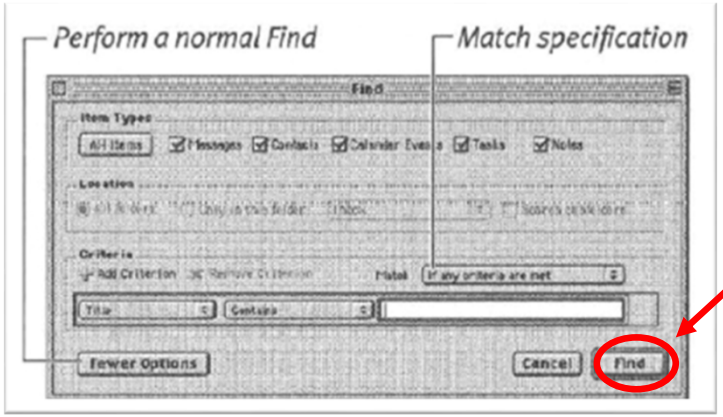


- **Petitioner similarly misleadingly labels a figure showing the Delimited Find Tool as if it were capable of incremental search**

'035 Paper 1 (Petition), 43; '490 Paper 1 (Petition), 46;
'139 Paper 1 (Petition), 49

Entourage's Three Distinct Tools

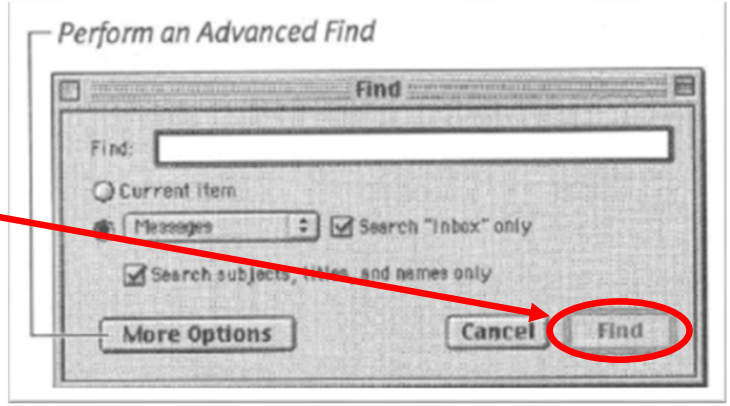
Entourage Advanced Find



EX1004 (Schwartz), 11
'035 Paper 21 (POR), 42-43
'490 Paper 21 (POR), 44-45
'139 Paper 21 (POR), 44-45

Delimited Search

Entourage Find



EX1004 (Schwartz), 9
'035 Paper 21 (POR), 40-42
'490 Paper 21 (POR), 41-44
'139 Paper 21 (POR), 41-44

Entourage List Filtering

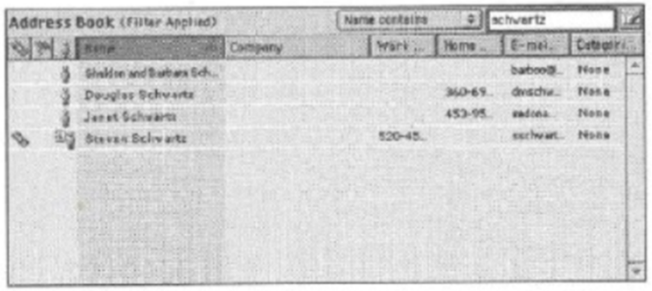


Figure 1.12 As you type the filter text, only matches are shown in the list.

EX1004 (Schwartz), 9
'035 Paper 21 (POR), 38-39
'490 Paper 21 (POR), 40-41
'139 Paper 21 (POR), 40-41

Entourage's List Filtering Tool Does Not Search for and Locate Documents and is Limited to a Single String

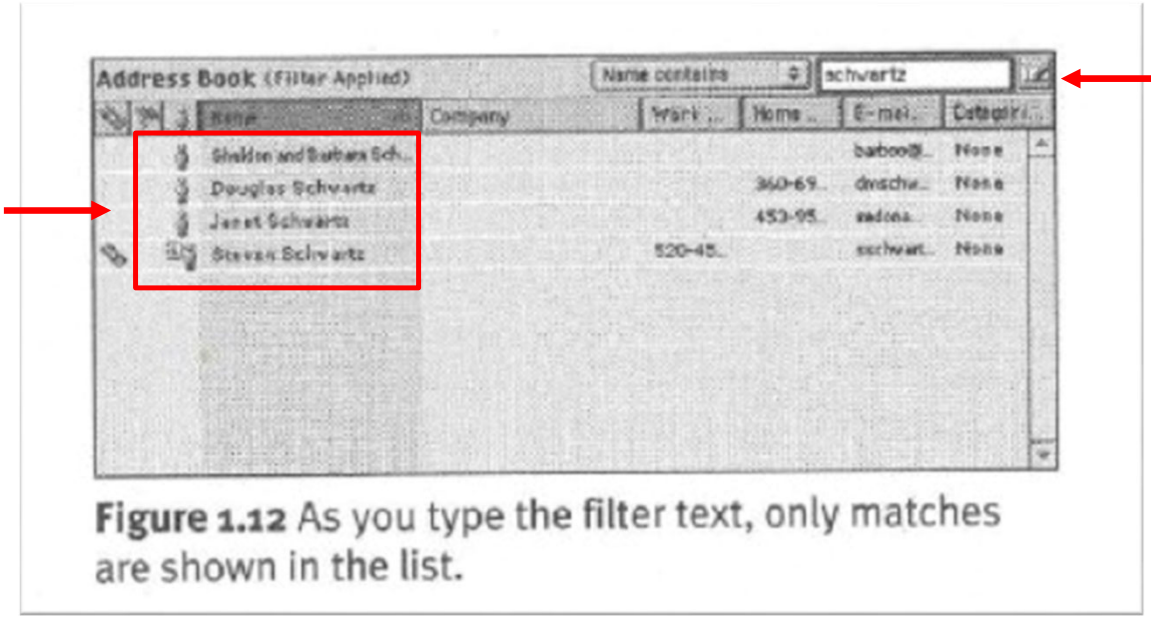


Figure 1.12 As you type the filter text, only matches are shown in the list.

EX1004 (Schwartz), 9; '035 Paper 21 (POR), 38-39;
'490 Paper 21 (POR), 40-41; '139 Paper 21 (POR), 40-41

- **Only available in certain Entourage windows displaying lists of Entourage items (e.g., address book entries)**
- **Petitioner fails to explain how this is equivalent to searching for and locating documents**
- **Limited to a single string in the search field**

Entourage's Find Is Not Incremental, Not Multi-String

Single String

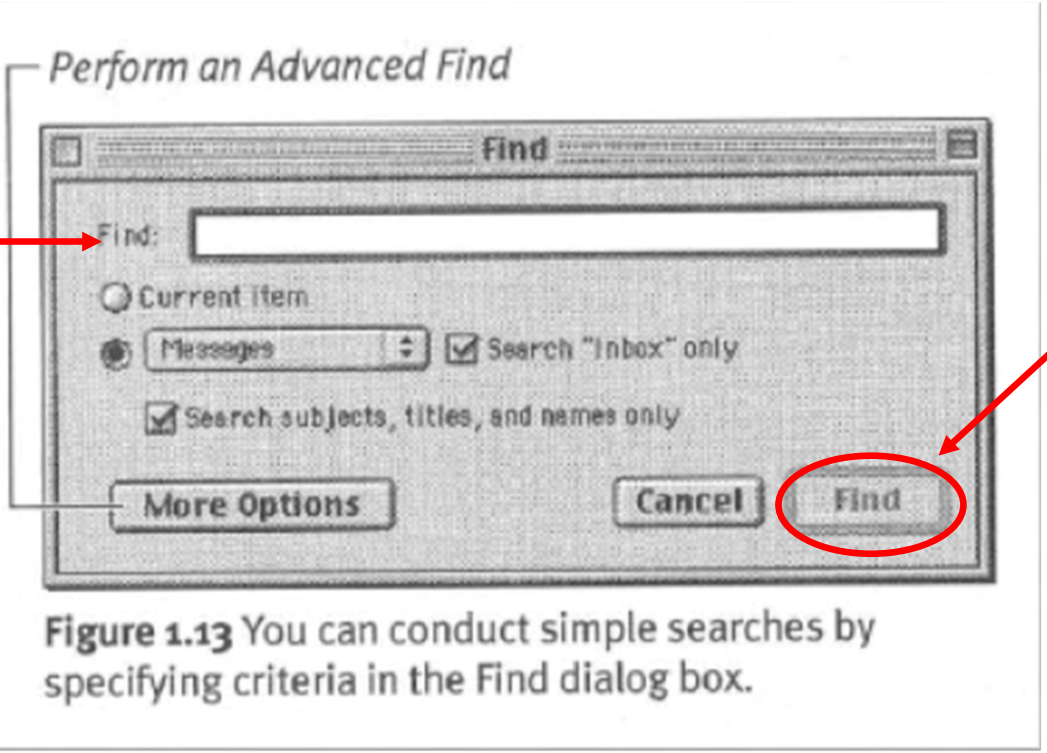


Figure 1.13 You can conduct simple searches by specifying criteria in the Find dialog box.

Delimited Search

EX1004 (Schwartz), 9
'035 Paper 21 (POR), 40-42
'490 Paper 21 (POR), 41-44
'139 Paper 21 (POR), 41-44

Entourage's Advanced Find Is Not Incremental, Not Multi-String

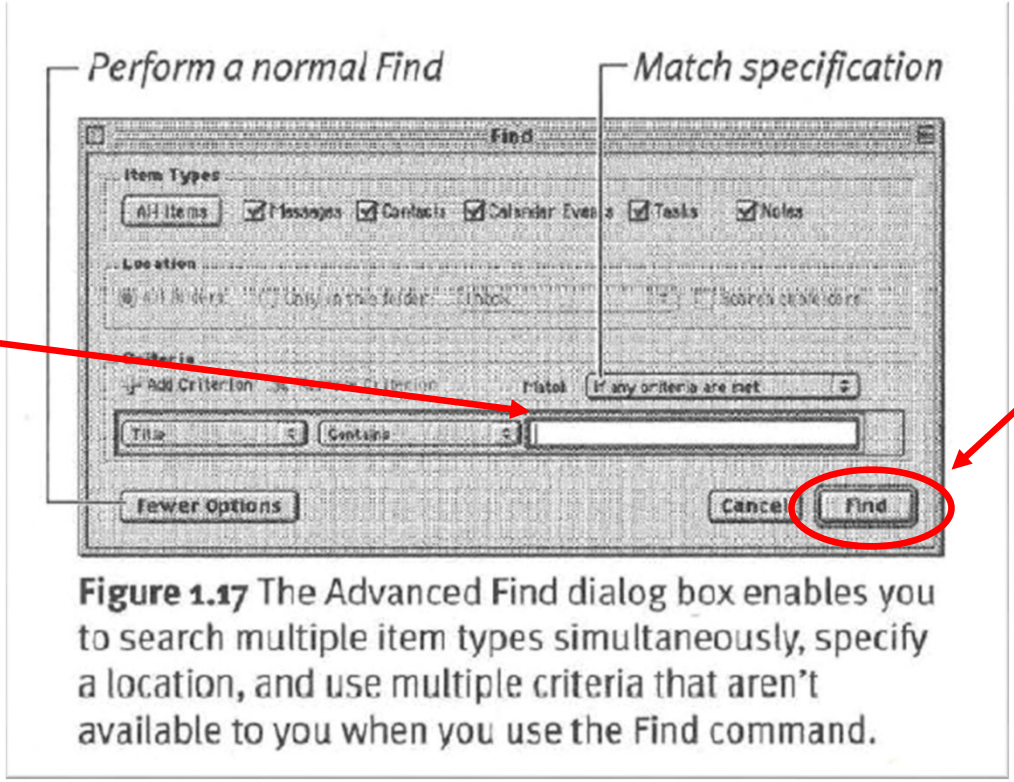


Figure 1.17 The Advanced Find dialog box enables you to search multiple item types simultaneously, specify a location, and use multiple criteria that aren't available to you when you use the Find command.

Single String

Delimited Search

EX1004 (Schwartz), 11
'035 Paper 21 (POR), 42-43
'490 Paper 21 (POR), 44-45
'139 Paper 21 (POR), 44-45

Entourage's Designers Made a Conscious Decision to Keep the Tools Separate

105. In Entourage's case, the separation appears driven by the performance issues I discussed earlier. (*Supra* §§IV.A.2, V.A.2.a.) The Entourage references explicitly describe the Find and Advanced Find features as being slower, particularly when searching across broader document sets. (EX1006, 384-86 (if you search "All Items," you should "be prepared to rent a video during the time Entourage will take the search its entire multimegabyte database."); EX1004, 9.) Maintaining the incremental list filtering as a separate, lightweight feature would have allowed users to quickly narrow down visible items without triggering the more resource-intensive indexed search operations required by Find and Advanced Find. Combining these approaches likely would have required either maintaining substantially larger indexes or performing real-time sequential scanning, either of which would exacerbate the existing performance problems and degrade the user experience. (*Supra* §§IV.A.2, V.A.2.a.)

'035 EX2020 (Supp. Haeberli Decl.), ¶105
'490 EX2020 (Supp. Haeberli Decl.), ¶75; '139 EX2020 (Supp. Haeberli Decl.), ¶110

- **Because the Entourage designers made a conscious decision to keep the search tools separate for performance considerations, a POSITA would not be motivated to combine them**
  - **Find and Advanced Find features are described as slower**
  - **Incremental list filtering was deliberately designed to be a separate, lightweight feature for quickly narrowing down a list of items**

'035 Paper 37 (Sur Reply), 11-15
'490 Paper 38 (Sur Reply), 10-13
'139 Paper 37 (Sur Reply), 10-14

Entourage's Find and Advanced Are Slow and Incompatible With Incremental Search

Find is "slower," can take hours to get results

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. But when the needle still eludes you in your rummage through the Entourage haystack, you can use Entourage's slower, more powerful search.

- Use the pop-up menu to search for the phrase within just one of the many facets of Entourage: Messages (your email), Contacts (your addresses), Calendar Events, Tasks, or Notes. If you can't even remember which of these features contains the elusive message, you can choose All Items—but be prepared to rent a video during the time Entourage will take to search its entire multimegabyte database.

EX1006 (Barber), 384-385
'035 Paper 21 (POR), 43; '490 Paper 21 (POR), 41; '139 Paper 21 (POR), 41

Search in Advanced Find can be "lengthy"

To use Advanced Find, choose Edit→Advanced Find (or click More Options in the Find window). As shown in Figure 11-7, the resulting mega-Find dialog box is divided into three sections:

- **Item Types.** Using the checkboxes in this section, you can select any combination of Messages, Contacts, Calendar Events, Tasks, or Notes to search; if you're willing to wait for a lengthy search, you can click All Items to check them all.

EX1006 (Barber), 385
'035 Paper 21 (POR), 42-43
'490 Paper 21 (POR), 44-45;
'139 Paper 21 (POR), 44-45

POSITA would not be motivated to combine Entourage's non-indexed "slow" multi-document search tools with incremental list filtering because doing so would result in search that is not incremental (fast as you type)

Sur-Reply, 12-1340

A POSITA Would Not Have Been Motivated With a Reasonable Expectation of Success to Combine True with the Entourage References

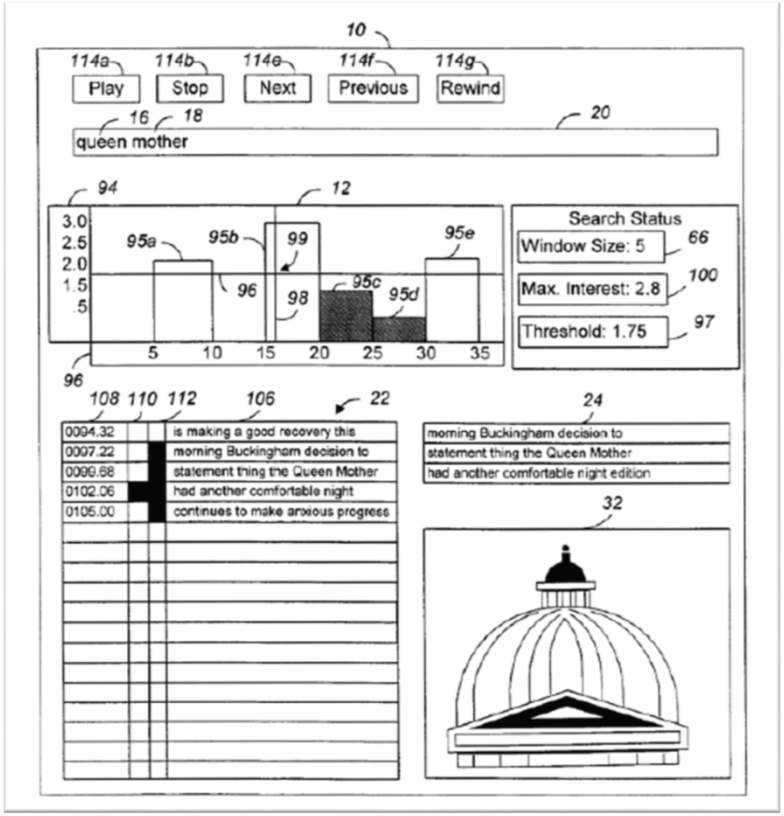
Petitioner's Motivation to Combine Entourage with True

A POSITA would have been motivated to combine the Entourage references' teachings to incrementally search at least one search string entered in a single search field (*e.g.*, EX1004, 8-10, 38-39, 89, 161, 204-205, 209, 267; EX1005, 173; EX1006, 358, 384, 387; *infra* Section VIII.E.1.c) with True's express teachings to do so for multiple search strings entered in the search field (EX1009, 2:26-31, 2:39-48, 4:38-43, 4:65-5:6, 5:41-50, FIG. 2; *infra* Sections VIII.E.1.b-VIII.E.1.c) at least because a POSITA knew that such a multi-string search would return more relevant results than a single-string query. *Infra* Sections VIII.E.1.b-VIII.E.1.c; EX1023, 25-26, 61, 101-105; EX1057, 27 ("Searching for two or more keywords—both of which *must* appear in the results—is an excellent strategy for doing precise searches and greatly reducing the number of search results you have to consider."). EX1003, ¶121.

Petitioner misunderstands True

'035 Paper 1 (Petition), 50
'490 Paper 1 (Petition), 53
'139 Paper 1 (Petition), 56

True Is Not Directed to Searching Documents



EX1009 (True), Figure 2
'035 Paper 21 (POR), 50; '490 Paper 21 (POR), 52;
'139 Paper 21 (POR), 51

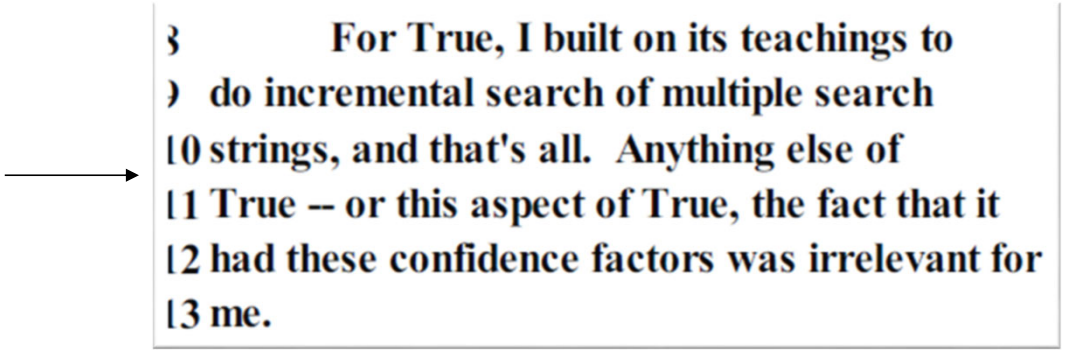
True discloses a system for visualizing the contents of a *single* audio file to determine the accuracy of the transcript

Result window provides a visual indication of which segments of the transcript (transcript windows) might contain at least one of the words in the query

POSITA would not be motivated to combine with True because it would not return accurate results

True's "Incremental Search" is Not Compatible with Multi-Document Incremental Search

Dr. Terveen ignored the context of True -a sequential, single-document search - which is incompatible with incremental search of Entourage's multi-document environment



For True, I built on its teachings to do incremental search of multiple search strings, and that's all. Anything else of True -- or this aspect of True, the fact that it had these confidence factors was irrelevant for me.

Petitioner's Conclusory Statement of Expectation of Success

A POSITA would have reasonably expected to succeed in combining True's and Baeza-Yates' teachings with the Entourage references' teachings at least because it would have required only simple and straightforward software modifications. *See Keynetik*, WL 2003932 at *2 (Fed. Cir. Feb. 15, 2023). EX1003, ¶¶122, 124.

'035 Paper 1 (Petition), 51; '490 Paper 1 (Petition), 54; '139 Paper 1 (Petition), 57

- **Petitioner relies on Dr. Terveen's conclusory statement**
- **Dr. Terveen's "simple and straightforward" opinion is rebutted by Mr. Haeberli's testimony**

In *Keynetik*, we held that expert testimony stating that certain software modifications were "straightforward" and "simple" was sufficient to establish a reasonable expectation of success. 2023 WL 2003932, at *1–2. However, in *Keynetik* there was no challenge to the expert's testimony as being conclusory or not credible, and the expert testimony was uncontradicted.

Juniper Networks, Inc. v. Correct Transmission, LLC, No. 2023-1046, 2024 WL 3517862, at *4 (Fed. Cir. July 24, 2024)

Petitioner Fails to Rebut Patent Owner's Reasonable Expectation of Success Evidence

Patent Owner's evidence that implementation of Entourage references with True would not be simple and straightforward:

- Incremental search requires a larger index
  - RAM constraints
- Would cause search to be slow such that they it would no longer be "incremental"

In reply, Petitioner changed its motivation to combine

Under the Petition's motivation to combine, Petitioner has not rebutted this evidence

> The only context in which a motivation to combine could plausibly exist is searching across a larger corpus of documents. Petitioner offers its two-document framing precisely to sidestep the feasibility concerns raised by Mr. Haeberli. (EX2020, ¶¶47-62, 114-118.) That retreat is telling. As explained in the POR, the unresolved technical challenges evident in the prior art, including index size, RAM constraints, and resulting performance degradation, would deprive a POSITA of any reasonable expectation of success in implementing incremental search across a non-trivial document set. Petitioner has not shown that the proposed combinations would successfully address any of these challenges. (EX2020, ¶¶47-62.) Petitioner's argument fails from a contradiction of its own making: a POSITA would have no motivation to combine in the two-document scenario, but the large-corpus scenario (i.e., the only one that could supply motivation) lacks any reasonable expectation of success. (*See infra* §§IV.C, IV.F.)

'035 Paper 37 (Sur-Reply), 9
'490 Paper 38 (Sur-Reply), 7-8
'139 Paper 37 (Sur-Reply), 8

The Schwartz, Sebastian, Barber, True and Baeza-Yates Combination Does Not Disclose Incremental Search Across Documents

Petitioner Fails to Establish the Schwartz, Sebastian, Barber, True, Baeza-Yates Combination Teaches All Limitations of the Claims

Exemplary Claim 1, '035 Patent

1. A method of performing a search comprising:
receiving a first string in a first search field;
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;
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;
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
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.

Entourage's List Filtering Tool

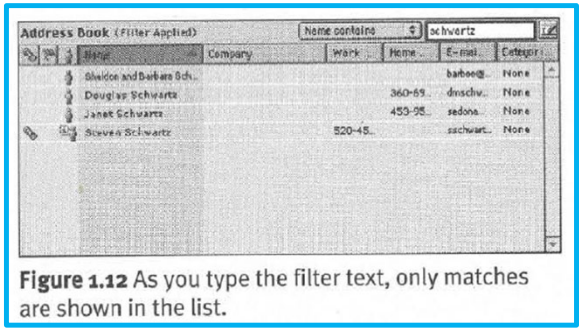
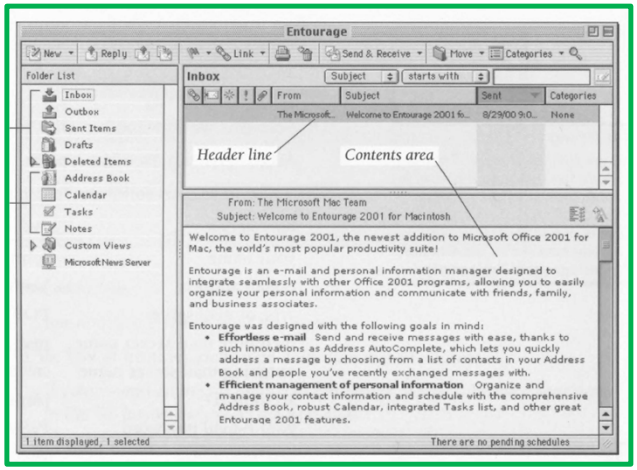


Figure 1.12 As you type the filter text, only matches are shown in the list.

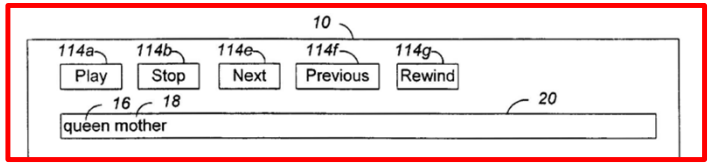
EX1004 (Schwartz), 9

Entourage's Delimited Find Tool



EX1005 (Sebastian), 173

True's Multiple Strings



EX1009 (True), Figure 2

- **Petitioner combines list filtering with the delimited Find tool for the "incrementally locating … documents" ('035) / "incrementally searching … documents … to identify search results" ('139) / "searching [for files] … as each successive character is … received" ('490) limitations**
- **Petitioner has not established the combination teaches these limitations**

All Challenged Claims Require Incremental Search Across Documents

Claim 1 of the '035 Patent

1. A method of performing a search comprising:
receiving a first string in a first search field;
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;
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;
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
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.

'035 patent (EX1001), claim 1

Claim 1 of the '490 Patent

searching the index data structure for a first one or more of the plurality of files each including at least one string that corresponds to the first partial search string;
searching the index data structure for a second one or more of the plurality of files each including at least one string that corresponds to the second partial search string; and

'490 patent (EX1001), claim 1

Claim 1 of the '139 Patent

1. A method of performing a search comprising:
receiving a first partial search string in a first search field;
receiving a second partial search string in the first search field;
wherein the first and second partial search strings are separated by a string separator character in the first search field;
incrementally searching a set of documents in order to identify search results, wherein the incrementally searching comprises:

'139 patent (EX1001), claim 1

Entourage's List Filtering Tool Does not Disclose Incrementally Searching Across Multiple Documents for Multiple Strings

- **Only available in certain Entourage windows displaying lists of Entourage items (e.g., address book entries)**
  - Petitioner fails to explain how this is equivalent to searching for and locating documents
- **Limited to a single string in the search field**

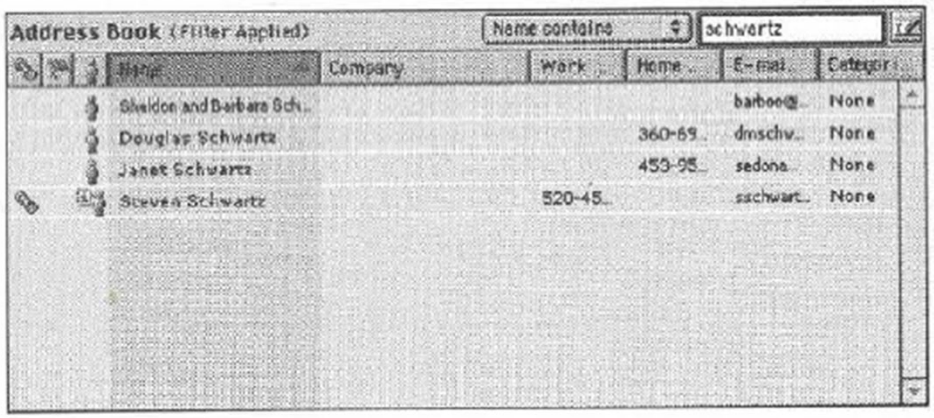


Figure 1.12 As you type the filter text, only matches are shown in the list.

EX1004 (Schwartz), 9

'035 Paper 21 (POR), 38-39; '490 Paper 21 (POR), 40-41; '139 Paper 21 (POR), 40-41

True Does Not Cure the Deficiencies of the Entourage References

- True teaches searching withing a single document (i.e., a transcript)
- True's search returns portions of the transcript ("transcript windows") that may include one or more of the search strings
  - Transcript windows are not documents
- True is not directed to precise matches to queries
  - Can return results that do not include both search strings

'035 Paper 21 (POR), 57-58; '490 Paper 21 (POR), 60-61; '139 Paper 21 (POR), 59-60

True is Not Analogous Art

True is Not in the Same Field of Endeavor as the Challenged Patents

Burden is on Petitioner to show that prior art is analogous by showing one of two tests is satisfied:

(1) **Same Field of Endeavor Test:** if the art is from the same field of endeavor or

(2) **Same Problem Test:** when the reference is not within the field of the inventor's endeavor, if it is reasonably pertinent to the particular problem with which the inventor is involved

The Field of Endeavor of the Challenged Patents

The field of endeavor for the '035 patent is systems and methods for incrementally searching for and locating indexed documents. (EX1001, Title, 1:29-30; 2:9-37; 3:34-46; 11:34-58; 15:50-65; 16:20-39; 20:28-45.) True does not relate

'035 Paper 21 (POR), 51 '490 Paper 21 (POR), 53; '139 Paper 21 (POR), 52

The Field of Endeavor of True

relate to searching documents for any given search strings. Instead, True provides a method for executing a search query on a single text file where the text does not necessarily correspond to the underlying raw data, and where the results of the query only indicate how likely it is that the text file accurately captures the underlying raw data. (EX2020, ¶¶112-13; EX2001, ¶131.) In other words, a query

'035 Paper 21 (POR), 51 '490 Paper 21 (POR), 53; '139 Paper 21 (POR), 52

Petitioner's Field of Endeavor for True Is Too Broad

The Allen–Bradley art is not in the same field of endeavor as the claimed subject matter merely because it relates to memories. It involves memory circuits in which modules of varying sizes may be added or replaced; in contrast, the subject patents teach compact modular memories. Thus, based on the evidence of record, the jury could reasonably have found that the first criterion of the analogous art test has not been met and that the prior art and the claimed subject matter are not in the same field of endeavor.

Wang Lab'ys, Inc. v. Toshiba Corp., 993 F.2d 858, 864 (Fed. Cir. 1993)

However, Sydansk cannot be considered to be within Clay's field of endeavor merely because both relate to the petroleum industry.... Clay's field of endeavor is the storage of refined liquid hydrocarbons. The field of endeavor of Sydansk's invention, on the other hand, is the extraction of crude petroleum. The Board clearly erred in considering Sydansk to be within the same field of endeavor as Clay's.

In re Clay, 966 F.2d 656, 659 (Fed. Cir. 1992)

'035 Paper 21 (POR), 52
'490 Paper 21 (POR), 54
'139 Paper 21 (POR), 53-54

True is Not Reasonably Pertinent to the Problem Solved by the Challenged Patents

Conventional search application programs often are slow and cumbersome to use. For example, many search engines require the user to type in a search term, click on a search button, and review the results. If the user is not satisfied with the results, as might be the case if too many items or items that are not of interest are found, then the user needs to edit the search terms, click on an initiate search button, and again review the results. In addition, many search application programs only provide for limited search filtering, wherein the user is limited to filtering the search based on dates, file location, and file contents.

Another drawback of some search engines is that the actual search process takes an annoying amount of time to complete a search. Furthermore, some search engines do not include search features directed to specific types of search targets, such as email, Web pages, and files. Alternatively and inconveniently, separate tools may be needed to search for email, Web pages, and files if the user desires corresponding specific search features.

'035 patent (EX1001), 1:32-50
'035 Paper 21 (POR), 53-54
'490 patent (EX1001), 1:43-61
'490 Paper 21 (POR), 55-56
'139 patent (EX1001), 1:38-56
'139 Paper 21 (POR), 54-56

POSITA would not have looked to True to address the Problem Identified by the Challenged Patents

- **Problem to be solved by Challenged Patents was slow and inefficient search across multiple documents, inability to accurately locate what you are searching for** (‘035 Paper 21 (POR); 53, ‘035 patent (EX1001), 1:32-2:5.)
- True would not solve this problem

  Uses sequential (brute force) scan of a single document (not groups of documents) – incompatible with searching across documents
- “Search results” of True may not even contain query

‘035 patent (EX1001), 1:32-2:5
‘035 Paper 21 (POR), 53-54
‘490 patent (EX1001), 1:43-2:16
‘490 Paper 21 (POR), 55-56
‘139 patent (EX1001), 1:38-2:11
‘139 Paper 21 (POR), 54-56

The Schwartz, Sebastian, Barber, True, Baeza-Yates Combination Does Not Render Obvious the Highlighting Claims

Claims 2-4 of the ’035 Patent Recite Highlighting

2. The method of claim **1**, further comprising displaying a portion of a selected document of the documents that are included in each of the first and second groups of documents.

3. The method of claim **2**, further comprising automatically highlighting a first occurrence of the first word in the selected document.

4. The method of claim **2**, further comprising highlighting at the same time a plurality of occurrences of the first word in the selected document while not highlighting portions of text between occurrences of the first word.

“highlighting”

‘035 patent (EX1001), claims 2-4

Claims 5 and 26 of the ’139 Patent Recite Highlighting

5. The method of claim **4**, further comprising highlighting a plurality of occurrences of the first and second partial search strings in the selected document while not highlighting portions of text between occurrences of the first and second partial search strings in the selected document.

“highlight … partial search strings”

‘139 patent (EX1001), claim 5

26. The computing system of claim **25**, wherein executing the software further causes the computing system to:

highlight in the user interface a plurality of occurrences of the first and second partial search strings in the selected document while not highlighting portions between occurrences of the first and second partial search strings in the selected document.

‘139 patent (EX1001), claim 26

Claim 11 of the ’490 Patent Recite Highlighting

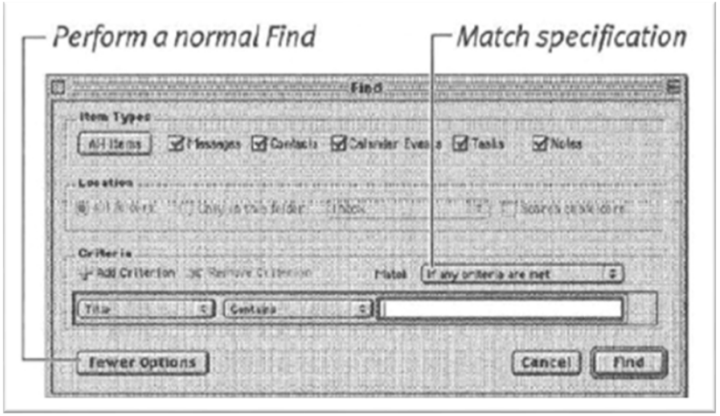
11. The computing system of claim **10**, wherein the software instructions further cause the computing system to: highlight a plurality of occurrences of the first and second partial search strings in the first file while not highlighting portions of text between occurrences of the first and second partial search strings in the first file.

‘490 patent (EX1001), claims 10-11

“highlight partial search strings”

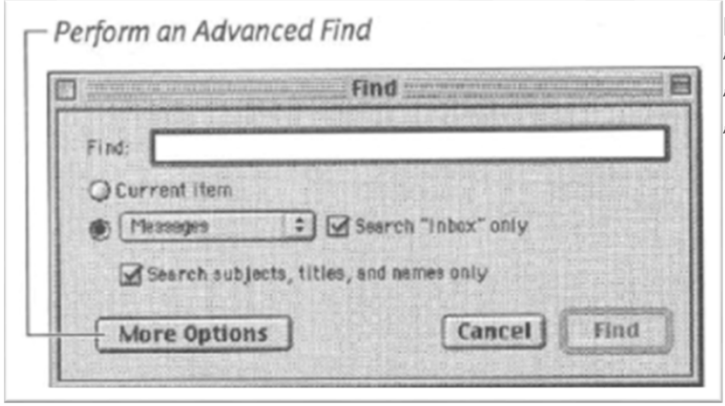
Petitioner Uses a Fourth Entourage Tool for the Highlighting Claims

Entourage Advanced Find



EX1004 (Schwartz), 11
'035 Paper 21 (POR), 42-43
'490 Paper 21 (POR), 44-45
'139 Paper 21 (POR), 44-45

Entourage Find



EX1004 (Schwartz), 9
'035 Paper 21 (POR), 40-42
'490 Paper 21 (POR), 41-44
'139 Paper 21 (POR), 41-44

Entourage List Filtering

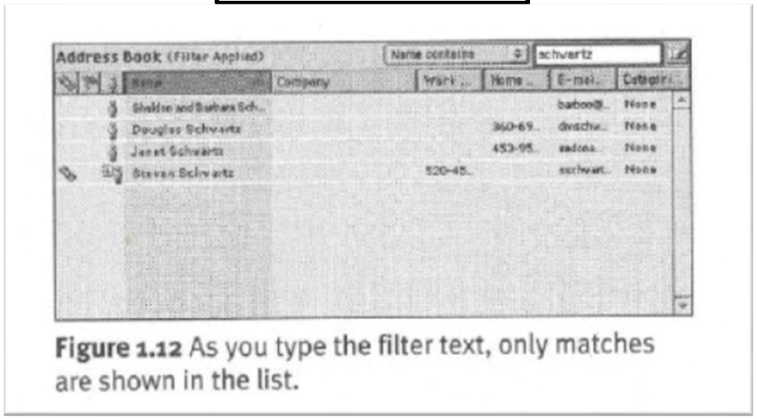


Figure 1.12 As you type the filter text, only matches are shown in the list.

EX1004 (Schwartz), 9
'035 Paper 21 (POR), 38-39
'490 Paper 21 (POR), 40-41
'139 Paper 21 (POR), 40-41

Entourage Find Within A Single Document

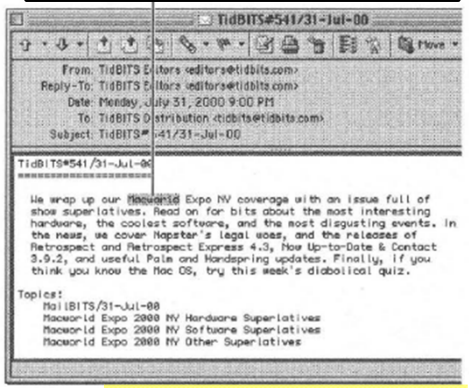


Figure 1.16 When you search for matching text within an item, Entourage highlights the first match, scrolling as necessary to display it.

EX1004 (Schwartz), 11
'035 Paper 21 (POR), 63
'490 Paper 21 (POR), 66
'139 Paper 21 (POR), 64

The Newly Added Feature to the Combination Does Not Highlight Strings When Searching Across Multiple Documents

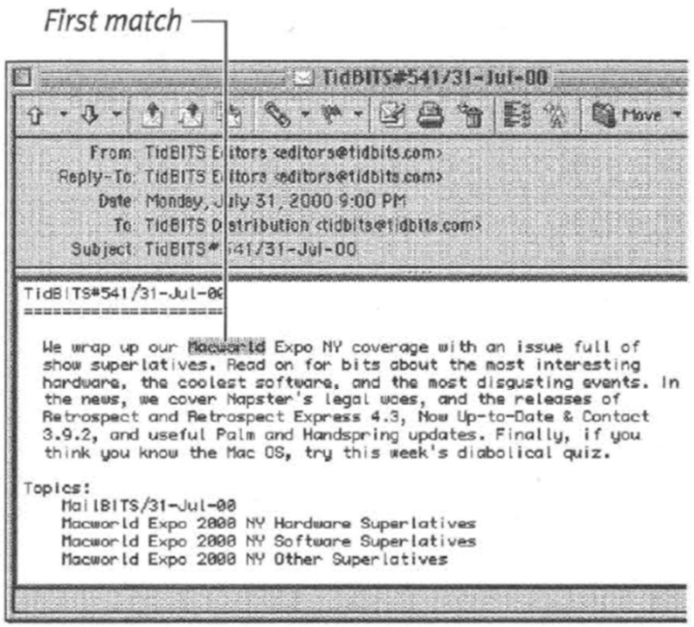


Figure 1.16 When you search for matching text within an item, Entourage highlights the first match, scrolling as necessary to display it.

5. Click the Find button.

The first instance of the text string (if there is one) is highlighted (**Figure 1.16**). To search for additional instances, press ⌘G.

✔ **Tip**

- When you search for matching contacts in the Address Book, it doesn't matter whether you check the Search Subjects, Titles, and Names Only checkbox. Regardless of your choice, Find searches only the Name field. To search other Address Book fields, use the Advanced Find command.

The Advanced Find command

The only disclosure of highlighting is associated with the Find Tool, not List Filtering

The Find Tool Highlights Only When Searching *Within a Single Document*

EX1004 (Schwartz), 11

Petitioner Improperly Gap-Fills with Knowledge of a POSITA

Moreover, it would have been obvious to highlight the first occurrence of the first word in the selected document at least because it was well known in the art to do so to draw the user's attention to the most relevant portions of the document. *Supra* Section IV.B.3; EX1023, 289. EX1003, ¶155.

'035 Paper 1 (Petition), 76; '490 Paper 1 (Petition), 84; '139 Paper 1 (Petition), 79

155. Moreover, it would have been obvious to highlight the first occurrence of the first word in the selected document at least because it was well known in the art to do so to draw the user's attention to the most relevant portions of the document. *Supra* Section VI.B.3. For example, as described by Baeza-Yates

'035 EX1003 (Terveen Decl.), ¶155
'490 EX1003 (Terveen Decl.), ¶192
'139 EX1003 (Terveen Decl.), ¶172

- **Petitioner and Dr. Terveen fall back on knowledge of a POSITA without pointing to where these elements are found in the prior art relied upon**

Petitioner Presented a New Argument in the Reply

Jul. 6, 2017). Specifically, a POSITA 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, especially since each reference provides additional teachings not found in the other references. For example, Schwartz provides extra details on text highlighting (e.g., EX1004, 163), Barber provides extra details on the Advanced Find search (e.g., EX1006, 387), and Sebastian provides extra details on the main email interface (e.g., EX1005, 173). EX1003, ¶130.

'035 Paper 1 (Petition), 45-46
'490 Paper 1 (Petition), 48-49
'139 Paper 1 (Petition), 51-52

41-45, 51-82. As also explained in the Petition, the highlighting functionality in the Find tool and multi-string search capability of the Advanced Find tool had known benefits: multi-string queries provide more targeted search results and search term highlighting directs the user to the relevant portion(s) of the results and gives the user more confidence in the results. Paper 1, 11, 52-54, 71-76; EX1003, ¶¶45-49, 121, 128, 149-157. Thus, a POSITA would have been motivated with a reasonable

'035 Paper 27 (Reply), 19-20
'490 Paper 28 (Reply), 18-19
'139 Paper 27 (Reply), 19-20

- **<u>None</u> of cited paragraphs of Petition and Terveen Declaration (EX1003) articulate a motivation to combine Find or Advanced Find because of these alleged favorable benefits**
- **New motivation is still insufficient for explaining why POSITA would ignore deliberate design choices by developers of the product**

Claims 9/14/19 and 11/16/21 of the ’139 Patent Require Searching on Specific Fields

9. The method of claim **1**, wherein at least some of the documents comprise at least the following file fields in which one or more of the first and second partial search strings may be located: name, path, size, item type, date modified, date created, or content.

14. The method of claim **12**, wherein the single field of the documents comprises one of: name, path, size, item type, date modified, date created, or content fields of file documents.

19. The method of claim **17**, wherein the single field of the documents comprises one of: name, path, size, item type, date modified, date created, or content fields of file documents.

’139 patent (EX1001), claims 9, 14, 19

- **Petitioner relies wholly on the knowledge of a POSITA for these claims without any explanation as to why a POSITA would have been motivated to modify Entourage, *an email client*, to search *non-email* files**

Petitioner Improperly Relied on Knowledge of a POSITA

Petition argument claim 9

on its name, file type, and location." EX1006, 18; *see also id.*, 17. A POSITA would have found it obvious to implement Entourage's document-specific search as an incremental search given the Entourage references' express teachings to perform incremental searches, and the known benefits of doing so, including making the search faster and more responsive. *Supra* Sections IV.B.1, VIII.E.1.c; EX1004, 8-

'139 Paper 1 (Petition), 83

Petition argument claim 14

As explained above for claims 8-11 (*supra* Sections VIII.E.7-VIII.E.10), the Entourage references disclose the ability to search within a single field of the documents (such that the incremental searching is performed only on content of that field of the documents) and disclose that the documents comprise the fields recited in claims 12-16. Thus, claims 12-16 are obvious for at least the reasons claims 8-11 are obvious. *Supra* Sections VIII.E.7-VIII.E.10. EX1003, ¶181.

'139 Paper 1 (Petition), 86

Petition argument claim 19

Claims 18-21 are obvious for at least the same reasons claims 8-11 are obvious. *Supra* Section VIII.E.7-VIII.E.10. EX1003, ¶183.

'139 Paper 1 (Petition), 87

Petitioner Improperly Relies on Knowledge of a POSITA for Claim 17 of the ’139 Patent

Although the Entourage references do not expressly disclose searching all textual content of the files for the first search string and only a single field of the files for the second search string within the same search field, the Entourage reference do expressly disclose doing the ability to restrict the search of a search string to a specific field of the files (albeit in a separate search field). EX1004, 11-13, 162-163; EX1006, 385-387. It would have been obvious to a POSITA, however, to permit such search functionality at least because it was well known to do so, such as by using field terms. For example, many search engines allowed users to enter a field term (e.g., “title,” “text”) before a search string to limit the search string to that particular field of the files. EX1023, 114; EX1057, 66 (Using the field term “title:” “[l]imits [the] search to the part of the Web page that the author labeled as the title.”), 67-68, 110-113, 152-153, 166. EX1003, ¶182.

Paper 1 (’139 Petition), 86-87

Petitioner Applies Its New "Two Documents" Claim Construction to Elements that Do Not Appear in the Claims of the '035 Patent

Petitioner's characterization of claims

EX2020, ¶¶8-9, 47-62, 114-118. The challenged claims, however, recite searching across a "group" of documents (i.e., two or more), not a large volume. Moreover, 2002-era computers were entirely capable of incrementally searching across a group of documents, especially considering that the median size of, for example, two emails in 2002 was approximately 4.4 kilobytes—0.00044% of the one gigabyte database assumed by Mr. Haeberli. EX1099, 190; EX1100, ¶¶13-32. Indeed, several prior art references cited by the Petition teach incrementally searching across documents, confirming that incrementally searching across documents (as opposed to within a single document) would not have required a leap of inventiveness. Paper 1, 9-10 (citing EX1004, EX1010, EX1012, EX1016, EX1021, EX1039); EX1100, ¶11. Moreover, searching across multiple documents was well known, as acknowledged by PO and its expert, Mr. Haeberli. EX1003, ¶¶33-44; Paper 21, 3; EX1098, 30:24-31:1; EX1100, ¶12; EX2020, ¶9.

'035 Paper 27 (Reply), 4; '139 Paper 27 (Reply), 4

What claims say

'035 Patent: "incrementally **locating a first group** of documents…. incrementally **locating a second group** of documents."
'139 Patent: "updating the search results as each successive character of the first partial search string and the second partial search string is received respectively in the first search field by **identifying one or more documents**..."

'035 patent (EX1001), claim 1
'139 Patent (EX1001), claim 1

The "group" of documents is the *output* of the incremental locating or identifying step, not the universe of documents to be searched.

Petitioner never says how (under "group" construction), a POSITA would find the claims *as drafted* obvious.

CERTIFICATE OF SERVICE

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

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

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

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

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

KLARQUIST SPARKMAN, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon, 97204
503-595-5300 (phone)
503-595-5301 (fax)
Msft-X1-IPRs@klarquist.com

Attorneys for Microsoft Corporation

Dated: April 22, 2026 By: /Lori A. Gordon/

Lori A. Gordon (Reg. No. 50,633)
GOODWIN PROCTER LLP
1900 N Street, N.W.
Washington, D.C. 20036
Phone: (202) 346 4435
Gordon-ptab@goodwinlaw.com

Lead Counsel for Patent Owner