

EX1003UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

MICROSOFT CORPORATION,  
Petitioner,

v.

X1 Discovery, Inc.,  
Patent Owner.

Case IPR2025-00254  
U.S. Patent No. 10,552,490  
Issued: February 4, 2020  
Application No. 15/495,507  
Filed: April 24, 2017

Title: METHODS AND SYSTEMS FOR SEARCH INDEXING

**PETITION FOR**  
**INTER PARTES REVIEW OF U.S. PATENT NO. 10,552,490**

**TABLE OF CONTENTS**

|                                                                                                                                                           | Page |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| LIST OF EXHIBITS .....                                                                                                                                    | vii  |
| MANDATORY NOTICES UNDER 37 C.F.R. § 42.8 .....                                                                                                            | xiii |
| 1. Real Party-In-Interest.....                                                                                                                            | xiii |
| 2. Related Matters .....                                                                                                                                  | xiii |
| 3. Lead And Back-Up Counsel, And Service Information.....                                                                                                 | xiv  |
| I. INTRODUCTION .....                                                                                                                                     | 1    |
| II. GROUNDS FOR STANDING PER 37 C.F.R. § 42.104(a) .....                                                                                                  | 1    |
| III. IDENTIFICATION OF CHALLENGE .....                                                                                                                    | 2    |
| A. Statement Of The Precise Relief Requested / Statutory Grounds.....                                                                                     | 2    |
| B. The Board Should Not Deny Institution Under § 325(d).....                                                                                              | 2    |
| C. The Board Should Not Deny Institution Under <i>Fintiv</i> .....                                                                                        | 3    |
| IV. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART.....                                                                                                  | 5    |
| A. Person Of Ordinary Skill In The Art.....                                                                                                               | 5    |
| B. State Of The Art .....                                                                                                                                 | 6    |
| 1. Incrementally (i.e., Character-By-Character) Locating<br>Documents Containing Partial Strings (e.g., Prefixes).....                                    | 6    |
| 2. Locating Documents Containing<br>Two Non-Adjacent Search Strings Entered<br>In The Same Search Field (e.g., Boolean AND Searching) .....               | 7    |
| 3. Using Indexes To Expedite The Searching Process .....                                                                                                  | 8    |
| 4. Displaying Portion(s) Of A Document Containing<br>The Search Term(s) At Non-Adjacent Locations<br>And Highlighting The Non-Adjacent Search Terms ..... | 8    |

|      |                                                                                                                   |    |
|------|-------------------------------------------------------------------------------------------------------------------|----|
| 5.   | Email-Specific Searching.....                                                                                     | 8  |
| V.   | THE '490 PATENT.....                                                                                              | 9  |
| A.   | The '490 Patent's Claims And Specification.....                                                                   | 9  |
| B.   | The Prosecution History.....                                                                                      | 10 |
| VI.  | CLAIM CONSTRUCTION .....                                                                                          | 11 |
| VII. | GROUND 1: CLAIMS 1-20 ARE<br>OBVIOUS OVER LOTUS NOTES<br>(EX1059, EX1061), RASKIN (EX1021), AND WU (EX1019) ..... | 11 |
| A.   | Overview Of The Lotus Notes References (EX1059, EX1061).....                                                      | 12 |
| B.   | Overview Of Raskin (EX1021).....                                                                                  | 15 |
| C.   | Overview Of Wu (EX1019) .....                                                                                     | 17 |
| D.   | Combining The Lotus Notes References, Raskin, And Wu .....                                                        | 18 |
| E.   | The Lotus Notes+Raskin+Wu<br>Combination Teaches Every Claim Element.....                                         | 20 |
| 1.   | Claim 1 .....                                                                                                     | 20 |
| a)   | Element [1.1]·4F .....                                                                                            | 20 |
| b)   | Elements [1.2-1.4] .....                                                                                          | 20 |
| c)   | Element [1.5] .....                                                                                               | 21 |
| d)   | Element [1.6] .....                                                                                               | 22 |
| e)   | Element [1.7] .....                                                                                               | 22 |
| f)   | Elements [1.8-1.10] .....                                                                                         | 23 |
| g)   | Elements [1.11-1.13] .....                                                                                        | 25 |
| 2.   | Claims 2-3 .....                                                                                                  | 30 |
| 3.   | Claim 4.....                                                                                                      | 31 |
| 4.   | Claims 5-7 .....                                                                                                  | 32 |

|     |                    |    |
|-----|--------------------|----|
| 5.  | Claim 8.....       | 34 |
| 6.  | Claim 9.....       | 34 |
| 7.  | Claims 10-11 ..... | 35 |
| 8.  | Claim 12.....      | 36 |
| 9.  | Claim 13.....      | 36 |
| 10. | Claims 14-15.....  | 37 |
| 11. | Claim 16.....      | 40 |
| 12. | Claim 17.....      | 41 |
| 13. | Claim 18.....      | 41 |
| 14. | Claim 19.....      | 42 |
| 15. | Claim 20.....      | 42 |

|       |                                                                                                                                   |    |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|----|
| VIII. | GROUND 2: CLAIMS 1-20 ARE OBVIOUS<br>OVER THE ENTOURAGE REFERENCES<br>(EX1004-EX1006), TRUE (EX1009), AND BAEZA-YATES (EX1023) .. | 43 |
| A.    | Overview Of The Entourage References (EX1004-EX1006).....                                                                         | 43 |
| B.    | Combining The Entourage References .....                                                                                          | 48 |
| C.    | Overview Of The True (EX1009)<br>And Baeza-Yates (EX1023) References .....                                                        | 49 |
| D.    | Combining Entourage With The<br>True And Baeza-Yates References .....                                                             | 53 |
| E.    | The Entourage+True+Baeza-Yates<br>Combination Teaches Every Claim Element .....                                                   | 54 |
| 1.    | Claim 1 .....                                                                                                                     | 54 |
| a)    | Element [1.1] .....                                                                                                               | 54 |
| b)    | Elements [1.2-1.4] .....                                                                                                          | 55 |
| c)    | Element [1.5] .....                                                                                                               | 55 |

|     |                                              |    |
|-----|----------------------------------------------|----|
| d)  | Element [1.6] .....                          | 57 |
| e)  | Element [1.7] .....                          | 57 |
| f)  | Elements [1.8-1.10] .....                    | 59 |
| g)  | Elements [1.11-1.13] .....                   | 64 |
| 2.  | Claims 2-3 .....                             | 73 |
| 3.  | Claim 4.....                                 | 74 |
| 4.  | Claims 5-7 .....                             | 76 |
| 5.  | Claim 8.....                                 | 78 |
| 6.  | Claim 9.....                                 | 79 |
| 7.  | Claims 10-11 .....                           | 80 |
| 8.  | Claim 12.....                                | 84 |
| 9.  | Claim 13.....                                | 84 |
| 10. | Claims 14-15 .....                           | 84 |
| 11. | Claim 16.....                                | 87 |
| 12. | Claim 17.....                                | 88 |
| 13. | Claim 18.....                                | 89 |
| 14. | Claim 19.....                                | 90 |
| 15. | Claim 20.....                                | 90 |
| IX. | NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS..... | 91 |
| X.  | CONCLUSION.....                              | 92 |
|     | CERTIFICATE OF COMPLIANCE.....               | 94 |
|     | CERTIFICATE OF SERVICE .....                 | 95 |

**TABLE OF AUTHORITIES**

Page(s)

**Cases**

|                                                                                                                     |        |
|---------------------------------------------------------------------------------------------------------------------|--------|
| <i>Fox Factory, Inc. v. SRAM, LLC</i> ,<br>944 F.3d 1366 (Fed. Cir. 2019) .....                                     | 91     |
| <i>Keynetik, Inc. v. Samsung Elecs. Co., Ltd.</i><br>No. 2022-1127, 2023 WL 2003932 (Fed. Cir. Feb. 15, 2023) ..... | passim |
| <i>Novartis AG v. Torrent Pharms. Ltd.</i> ,<br>853 F.3d 1316 (Fed. Cir. 2017) .....                                | 92     |
| <i>Phillips v. AWH Corp.</i> ,<br>415 F.3d 1303 (Fed. Cir. 2005) .....                                              | 11     |
| <i>VidStream LLC v. Twitter, Inc.</i> ,<br>981 F.3d 1060 (Fed. Cir. 2020) .....                                     | 12, 44 |

**Board Decisions**

|                                                                                                                    |   |
|--------------------------------------------------------------------------------------------------------------------|---|
| <i>Advanced Energy Ind. v. Reno Techs.</i> ,<br>IPR2021-01397, Paper 7 (PTAB Feb. 16, 2022) .....                  | 3 |
| <i>Apple Inc. v. Fintiv, Inc.</i> ,<br>IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) .....                          | 4 |
| <i>Apple Inc., v. Fintiv, Inc.</i> ,<br>IPR2022-00976, Paper 9 (PTAB Nov. 15, 2022) .....                          | 4 |
| <i>Becton, Dickinson &amp; Co. v. B. Braun Melsungen AG</i> ,<br>IPR2017-01586, Paper 8 (PTAB Dec. 15, 2017) ..... | 3 |
| <i>Lyft, Inc. v. RideShare Displays, Inc.</i> ,<br>IPR2021-01602, Paper 7 (PTAB April 11, 2022) .....              | 3 |
| <i>NetNut LTD. v. Bright Data LTD.</i> ,<br>IPR2021-01492, Paper 12 (PTAB Mar. 21, 2022) .....                     | 4 |
| <i>PEAG LLC v. VARTA Microbattery GmbH</i> ,<br>IPR2020-01212, Paper 8 (PTAB Jan. 6, 2021) .....                   | 4 |

|                                                                                                                         |        |
|-------------------------------------------------------------------------------------------------------------------------|--------|
| <i>RPX Corp. v. Rothschild Connected Devices Innovations, LLC</i> ,<br>IPR2016-00443, Paper 28 (PTAB Jul. 6, 2017)..... | 18, 48 |
| <i>RPX Corp. v. Rothschild Connected Devices Innovations, LLC</i> ,<br>IPR2016-00443, Paper 8 (PTAB Jul. 7, 2016).....  | 18, 48 |
| <i>Samsung Elecs. Co., v. Power2B, Inc.</i> ,<br>IPR2021-01190, Paper 11 (PTAB Jan. 6, 2022).....                       | 3      |

## **Statutes**

|                       |        |
|-----------------------|--------|
| 35 U.S.C. § 102 ..... | passim |
| 35 U.S.C. § 103 ..... | 2      |
| 35 U.S.C. § 325 ..... | 2      |

## **Rules**

|                          |    |
|--------------------------|----|
| 37 C.F.R. § 42.104 ..... | 1  |
| 37 C.F.R. § 42.24 .....  | 94 |
| 37 C.F.R. § 42.6 .....   | 95 |

## **Other**

|                                                                                                                                                                    |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 83 Fed. Reg. 51340 .....                                                                                                                                           | 11 |
| USPTO Memo.,<br><i>Interim Procedure for Discretionary Denials in AIA Post-Grant<br/>Proceedings with Parallel District Court Litigation</i> (June 21, 2022) ..... | 4  |

**LIST OF EXHIBITS**

| <b>No.</b> | <b>Description</b>                                                                                                                                                  |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1001       | U.S. Patent No. 10,552,490 (“ <b>490 patent</b> ”)                                                                                                                  |
| 1002       | File History of U.S. Patent No. 10,552,490                                                                                                                          |
| 1003       | Declaration of Dr. Loren Terveen, dated December 19, 2024                                                                                                           |
| 1004       | Steve Schwartz, <i>Visual QuickStart Guide Entourage 2001 for Macintosh</i> , Peachpit Press (2001) (“ <b>Entourage-1</b> ”)                                        |
| 1005       | Bonita Sebastian et al., <i>Microsoft Office 2001 A Hands-On Guide for Macintosh</i> , Computer Literacy Press (excerpts) (2000) (“ <b>Entourage-2</b> ”)           |
| 1006       | Nan Barber & David Reynolds, <i>Office 2001 for Macintosh: The Missing Manual</i> , Pogue Press/O’Reilly (1 <sup>st</sup> ed.) (Apr. 2001) (“ <b>Entourage-3</b> ”) |
| 1007       | <i>Eudora Email User Manual for Windows Version 5.0</i> , QUALCOMM (Aug. 2000) (“ <b>Eudora</b> ”)                                                                  |
| 1008       | Alan R. Neibauer, <i>Running Microsoft Outlook 2000</i> , Microsoft Press (excerpts) (1999) (“ <b>Outlook</b> ”)                                                    |
| 1009       | U.S. Patent No. 6,112,172 to True et al. (“ <b>True</b> ”)                                                                                                          |
| 1010       | U.S. Patent No. 5,444,823 to Trung D. Nguyen (“ <b>Nguyen</b> ”)                                                                                                    |
| 1011       | U.S. Patent No. 7,039,635 to Morgan et al. (“ <b>Morgan</b> ”)                                                                                                      |
| 1012       | U.S. Patent Application Publication No. 2002/0055981 to Spaey et al. (“ <b>Spaey</b> ”)                                                                             |
| 1013       | U.S. Patent No. 5,845,278 to Kirsch et al. (“ <b>Kirsch</b> ”)                                                                                                      |
| 1014       | U.S. Patent No. 4,255,796 to Gabbe et al. (“ <b>Gabbe</b> ”)                                                                                                        |
| 1015       | U.S. Patent No. 5,692,173 to Chee Heng Chew (“ <b>Chew</b> ”)                                                                                                       |
| 1016       | U.S. Patent No. 6,049,796 to Siitonen et al. (“ <b>Siitonen</b> ”)                                                                                                  |
| 1017       | U.S. Patent No. 5,303,361 to Colwell et al. (“ <b>Colwell</b> ”)                                                                                                    |

| No.  | Description                                                                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1018 | U.S. Patent No. 6,741,985 to Robin A. R. Green (“ <b>Green</b> ”)                                                                                                                                         |
| 1019 | U.S. Patent No. 5,991,756 to Jiong Wu (“ <b>Wu</b> ”)                                                                                                                                                     |
| 1020 | International Patent Application Publication No. WO 00/26762 to Mark Richanbach (“ <b>Richanbach</b> ”)                                                                                                   |
| 1021 | Jef Raskin, <i>The Humane Interface: New Direction for Designing Interactive Systems</i> , ACM Press (2000) (“ <b>Raskin</b> ”)                                                                           |
| 1022 | Richard M. Stallman, <i>GNU Emacs Manual</i> (1 <sup>st</sup> ed.) (Jun. 1985) (“ <b>GNU Emacs</b> ”)                                                                                                     |
| 1023 | Ricardo Baeza-Yates & Berthier Ribeiro-Neto, <i>Modern Information Retrieval</i> , ACM Press (1999) (“ <b>Baeza-Yates</b> ”)                                                                              |
| 1024 | Sunny Lam, <i>The Overview of Web Search Engines</i> (Feb. 9, 2001) (“ <b>Lam</b> ”)                                                                                                                      |
| 1025 | Oscar Vallarino, <i>On The Use Of Bit Maps For Multiple Key Retrieval</i> , Proceedings of the 1976 Conference on Data: Abstraction, Definition And Structure, 108-114 (Mar. 1976) (“ <b>Vallarino</b> ”) |
| 1026 | Steve Lawrence & C. Lee Giles, <i>Context And Page Analysis For Improved Web Search</i> , IEEE Internet Computing, 38-46 (Jul.-Aug. 1998) (“ <b>Lawrence</b> ”)                                           |
| 1027 | Rawley D. Johnson et al., <i>Technical Reference Document Database System (TDOCS) Requirements Definition</i> (Aug. 1993) (“ <b>Johnson</b> ”)                                                            |
| 1028 | Lev Finkelstein et al., <i>Placing Search in Context: The Concept Revisited</i> , ACM Transactions on Info. Sys., Vol. 20, No. 1, 116-131 (Jan. 2002) (“ <b>Finkelstein</b> ”)                            |
| 1029 | U.S. Patent No. 6,952,805 to Tafoya et al. (“ <b>Tafoya</b> ”)                                                                                                                                            |
| 1030 | U.S. Patent No. 6,009,459 to Belfiore et al. (“ <b>Belfiore</b> ”)                                                                                                                                        |
| 1031 | U.S. Patent No. 5,220,625 to Hatakeyama et al. (“ <b>Hatakeyama</b> ”)                                                                                                                                    |
| 1032 | U.S. Patent No. 6,185,551 to Birrell et al. (“ <b>Birrell</b> ”)                                                                                                                                          |

| No.  | Description                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1033 | U.S. Patent No. 6,839,702 to Patel et al. (“ <b>Patel</b> ”)                                                                                              |
| 1034 | U.S. Patent No. 6,973,450 to Silverbrook et al. (“ <b>Silverbrook</b> ”)                                                                                  |
| 1035 | U.S. Patent No. 6,834,276 to Jensen et al. (“ <b>Jensen</b> ”)                                                                                            |
| 1036 | U.S. Patent No. 5,946,678 to Ijsbrand Jan Aalbersberg (“ <b>Aalbersberg</b> ”)                                                                            |
| 1037 | European Patent Application Publication No. EP 1 209 866 to Baldonado et al. (“ <b>Baldonado</b> ”)                                                       |
| 1038 | Declaration of Delainey Tanner, dated November 18, 2024                                                                                                   |
| 1039 | Kaitlin Duck Sherwood, <i>Overcome Email Overload With Eudora 5</i> , World Wide Webfoot Press (excerpts) (1 <sup>st</sup> ed.) (2001)                    |
| 1040 | Claims Listing                                                                                                                                            |
| 1041 | Donald E. Knuth, <i>The Art Of Computer Programming Vol. 3 Sorting And Searching</i> , Addison Wesley Longman (excerpts) (2 <sup>nd</sup> ed.) (May 1998) |
| 1042 | United States District Court Judicial Caseload Profile (excerpt)                                                                                          |
| 1043 | Steven Schwartz & Robert Correll, <i>Running Microsoft Office 2001 for Mac</i> , Microsoft Press (excerpts) (2001)                                        |
| 1044 | Ben Shneiderman et al., <i>Clarifying Search: A User-Interface Framework For Text Searches</i> , D-Lib Mag. (Jan. 1997)                                   |
| 1045 | N/A                                                                                                                                                       |
| 1046 | N/A                                                                                                                                                       |
| 1047 | N/A                                                                                                                                                       |
| 1048 | N/A                                                                                                                                                       |
| 1049 | N/A                                                                                                                                                       |
| 1050 | N/A                                                                                                                                                       |
| 1051 | N/A                                                                                                                                                       |

| No.  | Description                                                                                                                                                                                    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1052 | Ben Shneiderman, <i>Designing The User Interface: Strategies For Effective Human-Computer Interaction</i> , Addison Wesley Longman (excerpts) (3 <sup>rd</sup> ed.) (Mar. 1998)                |
| 1053 | Claudine Badue et al., <i>Distributed Query Processing Using Partitioned Inverted Files</i> , Proc. Eighth Symp. on String Processing and Info. Retrieval, IEEE, 10-20 (2001)                  |
| 1054 | U.S. Patent No. 5,685,003 to Peltonen et al. (“ <b>Peltonen</b> ”)                                                                                                                             |
| 1055 | Dick Pountain, <i>The New Penguin Dictionary of Computing</i> , Penguin Group (excerpts) (1 <sup>st</sup> ed.) (2001)                                                                          |
| 1056 | File History of U.S. Patent No. 7,370,035                                                                                                                                                      |
| 1057 | Alfred and Emily Glossbrenner, <i>Search Engines For The World Wide Web</i> , PeachPit Press (3 <sup>rd</sup> ed.) (2001)                                                                      |
| 1058 | U.S. Patent No. 7,370,035 to Gross et al. (“ <b>the ’035 patent</b> ”)                                                                                                                         |
| 1059 | Adam A. Wilcox, <i>PC Learning Labs Teaches Lotus Notes 3.0</i> , Ziff-Davis Press (1993)                                                                                                      |
| 1060 | Declaration of Dr. Sylvia Hall-Ellis Dated December 16, 2024                                                                                                                                   |
| 1061 | Stephen Londergan & Pat Freeland, <i>Lotus Notes R5 For Dummies Quick Reference</i> , IDG Books Worldwide, Inc. (1999)                                                                         |
| 1062 | Declaration of Dr. Ingrid Hsieh-Yee Dated December 16, 2024                                                                                                                                    |
| 1063 | Exhibit CX-0196 to Complainant’s Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1064 | Exhibit CX-0198 to Complainant’s Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1065 | Exhibit CX-0199 to Complainant’s Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |

| No.  | Description                                                                                                                                                                                    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1066 | Exhibit CX-0203 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1067 | Exhibit CX-0204 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1068 | Exhibit CX-0206 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1069 | Exhibit CX-0208 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1070 | Exhibit CX-0209 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1071 | Exhibit CX-0211 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1072 | Exhibit CX-0213 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1073 | Exhibit CX-0214 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1074 | Exhibit CX-0215 to Complainant's Pre-Trial Brief in the Matter of <i>Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof</i> , ITC-337-TA-1389 (August 29, 2024) |
| 1075 | U.S. Patent No. 9,663,139 to Gross et al. (“ <b>139 patent</b> ”)                                                                                                                              |
| 1076 | File History of U.S. Patent No. 9,663,139                                                                                                                                                      |
| 1077 | N/A                                                                                                                                                                                            |

| No.  | Description                                                                                                                                                                                |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1078 | N/A                                                                                                                                                                                        |
| 1079 | Transcript, Evidentiary Hearing (Vol. IV) in the Matter of Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof, ITC-337-TA-1389 (excerpts) (October 9, 2024) |

**MANDATORY NOTICES UNDER 37 C.F.R. § 42.8**

**1. Real Party-In-Interest**

Microsoft Corporation files this Petition because Patent Owner has sued Microsoft in the Central District of California and Microsoft’s customers in the ITC for allegedly infringing the ’490 patent. Microsoft has sole control over this Petition and is the real party in interest (“RPI”) for this Petition. For completeness, Microsoft also identifies its customers Acer America Corporation, Acer Inc., ASUS Computer International, ASUSTeK Computer, Inc., Dell (Chengdu) Company Limited, Dell Products LP, and Dell Technologies Inc.—which have been named as respondents in the ITC proceeding—as additional RPIs for this petition.

**2. Related Matters**

**District Court**

The ’490 patent (EX1001) and contemporaneously-challenged family member patent nos. 7,370,035 (EX1058) and 9,633,139 (EX1075) have been asserted against Microsoft in the following district court litigation (the “**district court case**”):

- *XI Discovery, Inc. v. Microsoft Corp.*, Case No. 8:23-cv-02415 (C.D.Cal.), filed December 19, 2023.

ITC

Family member patent nos. 8,498,977 and 8,856,093 have been asserted against Acer America Corporation, Acer Inc., ASUS Computer International, ASUSTeK Computer, Inc., Dell (Chengdu) Company Limited, Dell Products L.P., and Dell Technologies Inc. in the following ITC proceeding (the “**ITC proceeding**”), and the district court case is currently stayed pending the resolution of the ITC proceeding:

- *Certain Computing Devices Utilizing Indexed Search Sys. and Components Thereof*; Inv. No. 337-TA-1389 (ITC), filed Dec. 19, 2023.

PTAB

Microsoft is concurrently challenging the '035 patent in IPR2025-00253 and the '139 patent in IPR2025-00255.

**3. Lead And Back-Up Counsel, And Service Information**

| <b>Lead Counsel</b>                                                                                                                  | <b>Back-up Counsel</b>                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Todd M. Siegel, Reg. No. 73,232<br>todd.siegel@klarquist.com                                                                         | Amy C. Haspel, Reg. No. 78,385<br>amy.haspel@klarquist.com<br><br>Andrew M. Mason, Reg. No. 64,034<br>andrew.mason@klarquist.com<br><br>Samuel B. Thacker, Reg No. 78,633<br>samuel.thacker@klarquist.com |
| KLARQUIST SPARKMAN, LLP<br>121 SW Salmon Street, Suite 1600<br>Portland, Oregon, 97204<br>503-595-5300 (phone)<br>503-595-5301 (fax) |                                                                                                                                                                                                           |

Petitioner consents to service via email at the above email addresses and the email address of Msft-X1-IPRs@klarquist.com.

Pursuant to 37 C.F.R. § 42.10(b), concurrently filed with this Petition is a Power of Attorney executed by Petitioner and appointing the above counsel.

Petitioner authorizes Account No. 02-4550 to be charged for any fees, including those enumerated in 37 C.F.R. § 42.15.

Microsoft Corporation (“Petitioner”) respectfully requests *inter partes* review (“IPR”) of all claims (claims 1-20) of U.S. Patent No. 10,552,490 (the “’490 patent”) (EX1001), allegedly assigned to X1 Discovery, Inc. (“Patent Owner”). For the reasons set forth below, these claims should be found unpatentable and cancelled.

## **I. INTRODUCTION**

The ’490 patent is a great-grandchild of the contemporaneously-challenged ’035 patent (U.S. 7,370,035) (EX1058) and contains very similar claims relating to incrementally searching two search strings entered in the same search field, but adds that the incremental search is performed on an index of the searched files (i.e., a data structure that contains associations between the search strings and the documents that contain the search strings). As explained below, however, the use of indexes to expedite a search was well known long before the alleged priority date of the ’490 patent. *Infra* Section IV.B.3. Thus, the challenged claims of the ’490 patent are obvious for at least the same and/or similar reasons that the challenged claims of the ’035 patent are obvious.

## **II. GROUND FOR STANDING PER 37 C.F.R. § 42.104(a)**

Petitioner certifies that the ’490 patent is available for IPR and that Petitioner is not barred or estopped from requesting an IPR challenging the patent claims on the grounds identified in this petition.

### III. IDENTIFICATION OF CHALLENGE

#### A. Statement Of The Precise Relief Requested / Statutory Grounds

Petitioner requests *inter partes* review of claims 1-20 (the “challenged claims”) of the ’490 patent, on the following statutory grounds:

| Ground | Reference(s)                                                                                   | Basis           | Claims |
|--------|------------------------------------------------------------------------------------------------|-----------------|--------|
| 1      | The Lotus Notes references <sup>1</sup> (EX1059, EX1061), Raskin (EX1021), and Wu (EX1019)     | 35 U.S.C. § 103 | 1-20   |
| 2      | The Entourage references <sup>2</sup> (EX1004-EX1006), True (EX1009), and Baeza-Yates (EX1023) | 35 U.S.C. § 103 | 1-20   |

For each ground, in Sections VII-VIII below, the petition presents evidence of unpatentability and establishes a reasonable likelihood that the Petitioner will prevail in establishing that each challenged claim is unpatentable.

#### B. The Board Should Not Deny Institution Under § 325(d)

The Board should not deny institution under Section 325(d) because neither “the same [n]or substantially the same prior art or arguments previously were presented to the Office.” 35 U.S.C. § 325(d). Eight of the nine cited references were not cited during prosecution and although Applicant cited the ninth cited reference

---

<sup>1</sup> Wilcox (EX1059) and Londergan (EX1061).

<sup>2</sup> Schwartz (EX1004), Sebastian (EX1005), and Barber (EX1006).

(True) in an IDS (EX1002<sup>3</sup>, 100), it was never cited in a rejection or substantively addressed by the Examiner. *See, e.g., Advanced Energy Ind. v. Reno Techs.*, IPR2021-01397, Paper 7 at 7 (PTAB Feb. 16, 2022); *Lyft, Inc. v. RideShare Displays, Inc.*, IPR2021-01602, Paper 7 at 14-15 (PTAB April 11, 2022); *Samsung Elecs. Co., v. Power2B, Inc.*, IPR2021-01190, Paper 11 at 19 (PTAB Jan. 6, 2022) (all finding *Becton Dickinson* factor (c) weighs against discretionary denial when a prior art reference was cited only in an IDS and was not cited in a rejection or substantively evaluated). *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8 at 17-18 (PTAB Dec. 15, 2017) (precedential).

Moreover, the Examiner erred in overlooking disclosure in True that teaches at least part of what the Examiner found lacking in the prior art of record—incrementally searching two search strings entered in the same search field. EX1002, 129; EX1009, 4:66-5:6, 5:41-50, FIG. 2. *See Becton Dickinson* at 18 (factors (e), (f)).

**C. The Board Should Not Deny Institution Under *Fintiv***

The Board should not discretionarily deny institution of this IPR under *Fintiv* at least because *Fintiv* factors 1-3 and 6 strongly favor institution. *Apple Inc. v.*

---

<sup>3</sup> Citations to EX1002 are to the page numbers (1-169) added by Petitioner.

*Fintiv, Inc.*, IPR2020-00019, Paper 11 at 5-12, 14-16 (PTAB Mar. 20, 2020) (precedential).

*Fintiv* factors 1 and 2 favor institution because: (1) the district court case is currently stayed pending resolution of the related ITC proceeding, and (2) even ignoring the stay, the median time-to-trial statistics for the district court (C.D.Cal.) put the district court trial date close to when a final written decision would be due. EX1042. *E.g., Apple Inc., v. Fintiv, Inc.*, IPR2022-00976, Paper 9 at 10-11 (PTAB Nov. 15, 2022); USPTO Memo., *Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation* (June 21, 2022) (“Vidal Memo”), 8-9.

*Fintiv* factor 3 favors institution because the district court and the parties in the district court case have invested very little effort in the district court case. Only the infringement complaint has been served, discovery has not begun, and defendant Microsoft has not served invalidity contentions. *E.g., NetNut LTD. v. Bright Data LTD.*, IPR2021-01492, Paper 12 at 11-12 (PTAB Mar. 21, 2022); *PEAG LLC v. VARTA Microbattery GmbH*, IPR2020-01212, Paper 8 at 17-20 (PTAB Jan. 6, 2021).

*Fintiv* factor 6 also favors institution because this petition presents compelling merits as explained below in Sections VII-VIII.

#### IV. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART

##### A. Person Of Ordinary Skill In The Art

A person of ordinary skill in the art for purposes of the '490 patent in September 2002 ("POSITA") would have had a bachelor's degree in computer science, or equivalent, with two or more years of experience developing and implementing search engines and/or search interfaces. More education, e.g., a Master's or Ph.D., would compensate for less work experience and vice versa. EX1003, ¶30.

As demonstrated by the background references in Section IV.B (State Of The Art), a POSITA would have been familiar with common, then-existing search tool functionality, including:

- incremental (i.e., character-by-character) searching in which search results are displayed as a user types their query (*infra* Section IV.B.1);
- locating documents containing two or more search strings, such as in response to a user including a Boolean AND operator in their search query (*infra* Section IV.B.2);
- using indexes to expedite the search process (*infra* Section IV.B.3);
- highlighting a searched string in the search results so that a user can easily and quickly see where the searched string appears in the located document(s) (*infra* Section IV.B.4); and

- email-specific search functionality, such as the ability to search within email-specific headers (e.g., To, From, Subject, etc.) (*infra* Section IV.B.5).

EX1003, ¶31.

## **B. State Of The Art**

The knowledge and skill of a POSITA encompassed the following claim-recited features, as reflected in both background references and the references that form the grounds of unpatentability presented in Sections VII-VIII. These features were well-known, widely deployed (including commercially), discussed in numerous patents, and often combined with one another. EX1003, ¶32.

### **1. Incrementally (i.e., Character-By-Character) Locating Documents Containing Partial Strings (e.g., Prefixes)**

By 2002, incrementally locating and displaying search results to a user as they type their query was well known and was commonly used in, for example, text editors, search engines, digital encyclopedias, help centers, and email clients at least because such incremental searching was known to provide several advantages over the other type of search, a delimited search (where a user must type their query in full before initiating a search and receiving results), including: improved responsiveness, efficiency, and usability. EX1003, ¶¶33-39 (citing/discussing

background references EX1004, EX1009-EX1018, EX1021-EX1022, EX1029, EX1039, EX1043-EX1044 and explaining the benefits of incremental searches).

**2. Locating Documents Containing  
Two Non-Adjacent Search Strings Entered  
In The Same Search Field (e.g., Boolean AND Searching)**

Locating documents containing multiple search strings/terms entered in a single search field (e.g., in response to the user including a space or Boolean AND operator between the search strings) was well known before 2002 and most, if not all, major search engines included such functionality. EX1003, ¶¶40-41 (citing/discussing background references EX1009, EX1019-EX1020, EX1057, EX1061).

It also was well known to locate documents containing multiple search strings by independently searching each string/term and then identifying the documents that appear in both sets of results. EX1003, ¶¶42-43 (citing/discussing background references EX1019-EX1020, EX1023-EX1026, EX1033, EX1041, EX1044, EX1052-EX1053, EX1057).

Moreover, it was well known that the search terms did not have to appear next to each other in the search results (i.e., the search terms could be non-adjacent). Indeed, many search tools allowed users to specify an allowable distance between search terms (a so-called “proximity search”). EX1003, ¶44 (citing/discussing

background references EX1017, EX1023-EX1024, EX1026, EX1044, EX1052, EX1059).

### **3. Using Indexes To Expedite The Searching Process**

Long before 2002, it was common to use indexes to speed up searches, and it was well known that indexes are not just convenient—but necessary—to perform searches in a reasonable amount of time, especially when the searched data structure is sufficiently large. EX1003, ¶¶45-47 (citing/discussing background references EX1013, EX1023, EX1032, EX1053-EX1055, EX1057, EX1059, EX1061 and explaining how indexes work and how they speed up a search).

### **4. Displaying Portion(s) Of A Document Containing The Search Term(s) At Non-Adjacent Locations And Highlighting The Non-Adjacent Search Terms**

It was well known before 2002 to display the portion(s) of a located document containing the search term(s) and highlight the search term(s) to give the user confidence in, and context for, the search results. EX1003, ¶¶48-52 (citing/discussing background references EX1004, EX1017, EX1020, EX1023, EX1027-EX1028, EX1030-EX1031, EX1033-EX1036, EX1038, EX1044, EX1057).

### **5. Email-Specific Searching**

Before 2002, it also was well known to search emails specifically. Indeed, most, if not all, then-existing email systems, such as Entourage, Eudora, and Outlook,

had search functionality allowing users to search for terms appearing in various fields of an email, including the To, Cc, From, and Subject fields, as well as the body of the email. EX1003, ¶53 (citing/discussing background references EX1004, EX1006-EX1008, EX1023, EX1032, EX1037, EX1039).

## **V. THE '490 PATENT**

The '490 patent is a child of the contemporaneously-challenged '139 patent (EX1075) and a great-grandchild of the contemporaneously-challenged '035 patent (EX1058). All three patents claim priority to a provisional application (60/408,015) filed September 3, 2002. Although Petitioner reserves the right to challenge this September 3, 2002, effective filing date, Petitioner does not do so here because the references cited herein are prior art to this September 2002 date.

### **A. The '490 Patent's Claims And Specification**

The '490 patent's claims and specification generally relate to incremental (i.e., character-by-character) searching where search results are displayed as the user types their search query, rather than waiting for the user to complete their query and press the "Enter" key or click a search button, as in a delimited search. *See, e.g.*, EX1001, 1:43-53 (disparaging delimited searches for requiring a user to click a search button before being able to view search results), 2:20-32 (touting the benefits of the application's incremental searching). EX1003, ¶26.

The claims specifically recite incrementally searching two strings entered into the same search field that are partial strings (e.g., the beginnings of words). EX1001, 29:55-30:28 (claim 1). EX1003, ¶27.

EX1040 includes a complete listing of the challenged claims.

For example, the specification describes that:

[I]f the user enter [sic] “Jo D” documents with the strings (“Jon” and “David”), (“Joe” and “Daniel”), (“John” and “Doe”) will be located. If user then adds an “h” to “Jo” so that the search strings are now “Joh D” then the list of search results will be substantially immediately narrowed to present only documents having at least one string beginning with “Joh” and one string beginning with “D”, such as a document containing (“John” and “Doe”).

EX1001, 23:8-16. EX1003, ¶¶28-29.

The ’490 patent’s dependent claims relate to specifics of the searching, including the types and fields of the files searched. EX1003, ¶29.

## **B. The Prosecution History**

The ’490 application was filed with one claim. EX1002, 80. Applicant filed a preliminary amendment cancelling that sole claim and adding new claims 2-21. *Id.*, 115-118. The Examiner then allowed these new claims without ever rejecting them. *Id.*, 124-131.

## **VI. CLAIM CONSTRUCTION**

Unless otherwise noted, Petitioner applies the plain and ordinary meaning of all claim terms below. 83 Fed. Reg. 51340; *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-1313 (Fed. Cir. 2005) (*en banc*). Petitioner does not, however, waive any argument in any litigation that claim terms in the '490 patent are indefinite or otherwise invalid nor does Petitioner waive its right to raise additional issues of claim construction in any litigation.

## **VII. GROUND 1: CLAIMS 1-20 ARE OBVIOUS OVER LOTUS NOTES (EX1059, EX1061), RASKIN (EX1021), AND WU (EX1019)**

As explained below, claims 1-20 are obvious over the Lotus Notes references (EX1059, EX1061), Raskin (EX1021), and Wu (EX1019). EX1003, ¶56.

The Lotus Notes references teach every element of the challenged claims, including locating files containing two search strings entered in a single search field, but do not expressly disclose two of the independent claims' features: (1) locating the files incrementally (i.e., character-by-character) and (2) doing so using an intersection approach (i.e., searching each string separately and then identifying the files that appear in both sets of search results). Both features were well known in the art (*supra* Sections IV.B.1, IV.B.2), however, and Raskin expressly discloses (1) and Wu expressly discloses (2). EX1003, ¶57.

**A. Overview Of The Lotus Notes References (EX1059, EX1061)**

The Lotus Notes references (EX1059, EX1061) are books describing IBM's Lotus Notes software. EX1059, (a), xvi-xviii; EX1061, (i), 2. EX1003, ¶58.

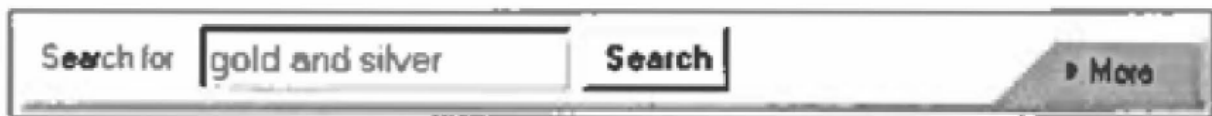
The Lotus Notes references qualify as prior art under Sections 102(a) and 102(b) (pre-AIA) because they published in 1993 and 1999 and were publicly accessible more than one year before the earliest alleged priority date of the '490 patent (September 3, 2002). EX1059, (h) (publication date); EX1060, ¶¶126-135 (showing public accessibility of EX1059); EX1061, (iii) (publication date); EX1062, ¶¶24-51 (showing public accessibility of EX1061); *see also* EX1061, (v) (describing how the publisher, IDG Books Worldwide, is the “#1 publisher of best-selling computer books in the United States” and how their “best-selling ... *For Dummies*® series has more than 50 million copies in print with translations in 31 languages.”); *VidStream LLC v. Twitter, Inc.*, 981 F.3d 1060, 1065 (Fed. Cir. 2020) (“When there is an established publisher there is a presumption of public accessibility as of the publication date.”).

The Lotus Notes references are analogous art to the '490 patent because they are from the same field of endeavor: systems and methods for providing a user interface through which a user can enter search strings, and locating files (e.g., emails) containing the search strings. *E.g.*, EX1001, 1:38-39, 2:20-32, 3:46-4:7,

4:41-48; FIGS. 3A, 3G-3I; EX1059, 83-85, 159-171; EX1061, 40-41, 133-148. EX1003, ¶59.

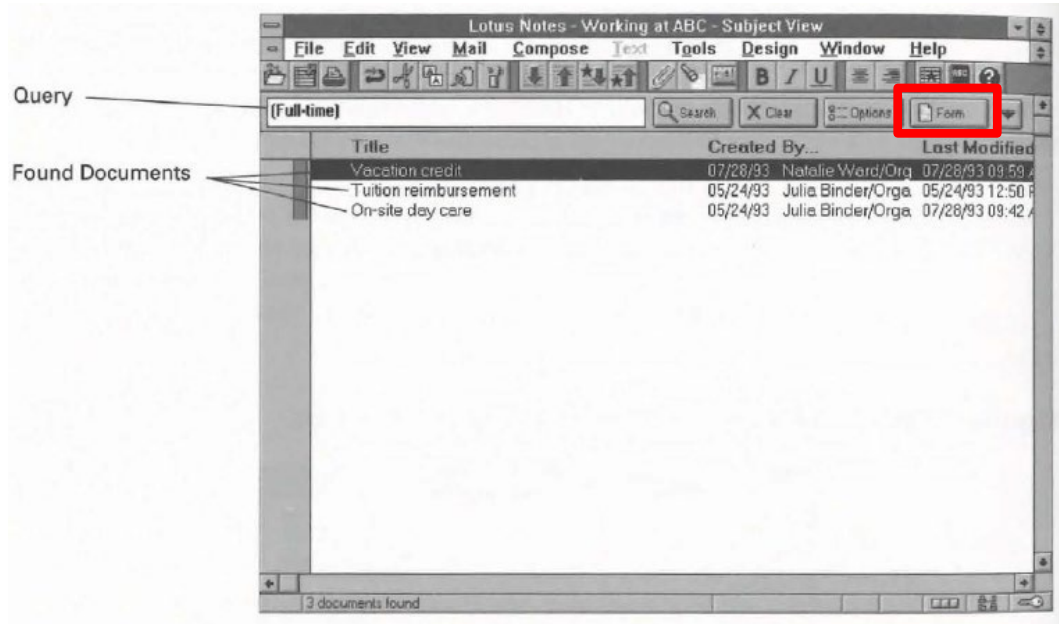
The Lotus Notes references explain how Lotus Notes works and how to use it. EX1059, *passim*. Londergan (EX1061) is directed to Lotus Notes R5 and Wilcox is directed to Lotus Notes 3.0 (an earlier version of Lotus Notes). EX1059, (a); EX1061, (i). EX1003, ¶60.

As these reference explain, Lotus Notes is a groupware product (EX1059, 1; EX1061, 5-8) that, *inter alia*, indexes files (EX1059, 168-173; EX1061, 41, 135-139), including email messages (EX1059, 94-134; EX1061, 41, 135-139), and allows users to search for files containing multiple search strings by entering the search strings in a single search field (EX1059, 160-165; EX1061, 139-142) of a simple Search bar or more advanced Query Builder/Search Builder and separating the strings with a comma or a Boolean AND operator. EX1059, 160-168; EX1061, 133-148. EX1003, ¶¶61-64.



EX1061, 140 (showing the Lotus Notes Search bar). EX1003, ¶63.

The Lotus Notes references also disclose searching within specific fields, including within the title, category, and/or body fields of the files, such as by selecting the “Form” button. EX1059, 166-168; EX1061, 143-144.



EX1059, 164 (red annotations added).

**Figure 5.20 Searching using a form**

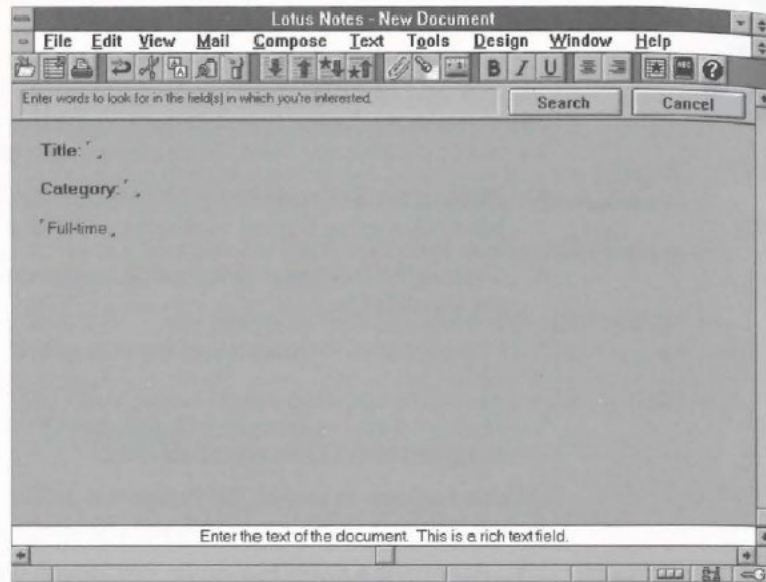


**Figure 5.21 A query created using a form**



EX1059, 168. EX1003, ¶65.

Although the Lotus Notes references do not expressly disclose performing an incremental search, Raskin does, and Raskin explains the well-known benefits of doing so, including better responsiveness, effectiveness, and usability. EX1021, 125-127. EX1003, ¶66.

## **B. Overview Of Raskin (EX1021)**

Raskin is a textbook describing user interface design principles, including for search interfaces. EX1021, (xi)-(xii), 124-132. EX1003, ¶67.

Raskin qualifies as prior art under Sections 102(a) and 102(b) (pre-AIA) because it published in 2000 and was publicly accessible more than one year before

the earliest alleged priority date of the '490 patent (September 3, 2002). EX1021, iv (publication date); EX1060, ¶¶81-89, 135 (showing public accessibility).

Raskin is analogous art to the '490 patent because it is from the same field of endeavor: systems and methods for providing a user interface through which a user can enter search strings, and locating files containing the search strings. *E.g.*, EX1001, 1:38-39, 2:20-32, 3:46-4:7, 4:41-48; FIGS. 3A, 3G-3I; EX1021, 118-132. EX1003, ¶68.

Specifically, Raskin describes that there are two main types of searches—delimited searches and incremental searches—and that incremental searches are almost always preferred because they are faster, more effective, and easier to use. EX1021, 125-127. An incremental search “immediately begins to search” as soon as the user “types the first character of the pattern” (EX1021, 125), and continues to update the search results as each character is added or deleted. EX1021, 125-126. Thus, incremental searches “provide[] constant feedback as to the results of the search,” thus “wast[ing] less of a user’s time” and allowing “the user to improve [their] search strategies through the feedback received.” EX1021, 126-127. EX1003, ¶¶69-70.

Although Lotus Notes and Raskin do not expressly disclose locating files containing both user-entered search strings by searching each string separately and

then combining/intersecting the results, Wu does. EX1019, 6:51-66, 8:66-9:44. EX1003, ¶71.

**C. Overview Of Wu (EX1019)**

Wu is a U.S. patent assigned to Yahoo, Inc., that describes a process for searching for, locating, and retrieving indexed files that contain user-entered search strings. EX1019, Abstract, 3:24-43, 6:51-9:33, FIGS. 3-6. EX1003, ¶72.

Wu qualifies as prior art under Sections 102(a) and 102(b) (pre-AIA) because it published in 1999 and was publicly accessible more than one year before the earliest alleged priority date of the '490 patent (September 3, 2002). EX1019, Cover.

Wu is analogous art to the '490 patent because it is from the same field of endeavor: systems and methods for providing a user interface through which a user can enter search strings, and locating files containing the search strings. *E.g.*, EX1001, 1:38-39, 2:20-32, 3:46-4:7, 4:41-48; FIGS. 3A, 3G-3I; EX1019, Abstract, 3:24-43, 6:51-9:33, FIGS. 3-6. EX1003, ¶73.

Specifically, Wu describes a process for locating files containing at least two search strings (e.g., the words “Game” and “Go”) by searching each string separately, generating match lists for each string, and then identifying the files in both sets of match lists. EX1019, 6:51-66, 8:66-11:32, FIGS. 5-6. EX1003, ¶74.

**D. Combining The Lotus Notes References, Raskin, And Wu**

A POSITA would have been motivated with a reasonable expectation of success to combine the teachings from the Lotus Notes references on locating indexed files containing multiple strings entered in a single search field with Raskin's teachings to provide search results incrementally (i.e., as the user types their query) and Wu's teachings to locate the files containing the multiple strings by searching each string separately and then intersecting the results. EX1003, ¶75.

A POSITA would have been motivated to combine the complementary teachings on Lotus Notes from Londergan (EX1061) and Wilcox (EX1059) to obtain a more complete and comprehensive understanding of the Lotus Notes system, its features, and its capabilities. *See RPX Corp. v. Rothschild Connected Devices Innovations, LLC*, IPR2016-00443, Paper 8 (institution decision) at 21 (PTAB Jul. 7, 2016), Paper 28 (final written decision) at 35 (PTAB Jul. 6, 2017) (finding a motivation to combine two references because, *inter alia*, they described the same system). For example, Wilcox provides additional details on the core functionality of Lotus Notes' user interface and search functionality (EX1059, 84-85, 159-168) while Londergan provides details on new features of Lotus Notes R5, including its calendaring and auto-complete features (EX1061, 45-74, 146-147). EX1003, ¶76.

A POSITA would have reasonably expected success in combining the two references' complementary teachings on Lotus Notes' search functionality at least

because the references describe using the same search tool (e.g., Lotus Notes' Search bar) for the same purpose: finding files containing multiple strings entered in a single search field. Moreover, a POSITA would have understood Lotus Notes' core user interface and search functionality to be the same and/or at least compatible between the two versions. EX1003, ¶77.

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, ¶78.

When implementing Lotus Notes' multi-string search, a POSITA would have been motivated to locate the files containing the two search strings by searching each string separately and then combining/intersecting the results, as taught by Wu, at least because it was one of a limited number of well-known, predictable approaches, was simple and easy to implement, and is especially well suited for incremental searches. EX1019, 6:52-62, 9:11-12; *see also supra* Section IV.B.2. EX1003, ¶79.

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, ¶¶80-81.

**E.     The Lotus Notes+Raskin+Wu  
Combination Teaches Every Claim Element**

**1.     Claim 1**

**a)     Element [1.1]<sup>4,5</sup>**

The Lotus Notes references teach a computing system (e.g., personal computer) on to which a user installs/downloads the Lotus Notes software. EX1059, xx, xxv-xxix, 3-5, 6 (“[T]he Notes workstation software must be installed on your computer’s hard drive.”); EX1061, 128-129, 136-137, 148, 187 (“Notes [is] installed on users’ computers.”), 190, 191. EX1003, ¶82.

**b)     Elements [1.2-1.4]**

The Lotus Notes references teach these claim elements: a processor (e.g., a 286-based system), a display device (e.g., a monitor), and a computer readable storage medium (e.g., a hard drive, disk). EX1059, xx-xxi, xxv-xxix, 3-5; EX1061,

---

<sup>4</sup> See EX1040 for a complete listing of the claim elements.

<sup>5</sup> Reference numbers in the format of [claim#.element#] are added throughout for ease of reference.

128-129, 136. Moreover, a POSITA would have understood that the computing system includes these components because they are fundamental components of a computing system. EX1003, ¶83.

**c) Element [1.5]**

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.

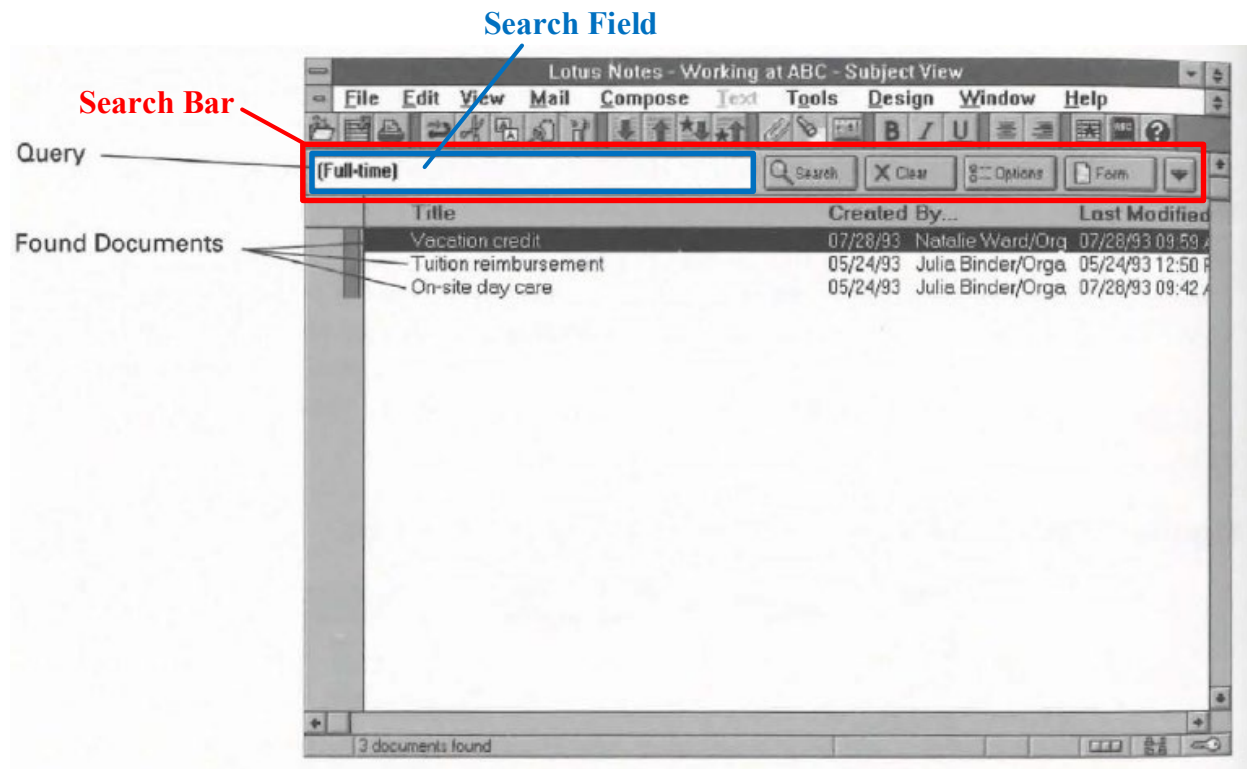
**d) Element [1.6]**

The Lotus Notes references teach this claim element at least because they disclose software instructions (the software that runs Lotus Notes) that are downloaded to the computing system's memory. EX1059, xx, xxv-xxix, 3-5, 6 (“[T]he Notes workstation software must be installed on your computer's hard drive.”); EX1061, 128-129, 136-137, 148, 187 (“Notes [is] installed on users' computers.”), 190, 191. A POSITA would have understood or at least found it obvious that the software is executed by the processor of the computing device to perform one or more operations (e.g., the claimed operations in elements [1.7-1.13] below) at least because this was a basic and fundamental operating principle of computers that a POSITA was well aware of. EX1003, ¶85.

**e) Element [1.7]**

The Lotus Notes references teach displaying an interactive user interface on the display device that includes at least a first search field. EX1059, 159-168; EX1061, 133-134, 139-145. EX1003, ¶86.

For example, the Lotus Notes references disclose displaying a Search bar on the display device that includes a search field in which a user can enter multiple search strings. EX1059, 161-162, 164-166; EX1061, 139-140.



EX1059, 164 (blue and red annotations added). EX1003, ¶87.

**f) Elements [1.8-1.10]**

The Lotus Notes references teach receiving a first search string (a word such as “gold”)<sup>6</sup> and a second search string (another word such as “silver”) in a single

---

<sup>6</sup> The '490 patent states “[t]he search character or strings can include one or more alpha characters, numeric characters, **words** and/or phrases.” EX1001, 15:38-39 (emphasis added); *see also id.*, 16:60-17:3, 17:34-39, 21:28-30, 22:2-7, 23:3-5, 24:40-41, 25:9-12, 26:18-23, 28:19-23. During prosecution of an ancestor

search field (e.g., a blank box of a Search bar and/or a blank box of a Query Builder/Search Builder dialog box) separated by a string separator character (e.g., a comma and/or a Boolean AND operator<sup>7</sup>). EX1059, 159-168; EX1061, 17, 133-148, 195.



---

application, Applicant admitted that two words or prefixes, such as “ree” and “\*live,” are different strings if they are searched separately. EX1056, 219.

<sup>7</sup> To the extent Patent Owner argues the Boolean AND operator is not a string separator character (which Petitioner does not acquiesce is appropriate), it would have been obvious to a POSITA when implementing Raskin’s teachings to perform the search incrementally to use a single-character string separator character, such as a plus sign, as was known in the art (e.g., EX1057, 60, 109, 129, 151), or a comma, as described by the Lotus Notes references (EX1059, 159-165), at least because a POSITA knew that a goal of the incremental search is to minimize the number of characters the user has to enter before obtaining the desired search results. *E.g.*, EX1021, 125-127.

EX1061, 140; *see also id.* (“[T]ype **gold and silver** to find documents that contain both the word *gold* and the word *silver*.”); EX1059, 162 (“You can use the *logical operators* AND ... to find documents that contain various combinations of words and text strings.”). EX1003, ¶¶88-90.

When implementing Lotus Notes’ search tool to perform incremental searches, as taught by Raskin (*see infra* Section VII.E.1.g), a POSITA would have understood or at least found it obvious that the received search strings are partial search strings because incremental searches provide users with results as the users type in their search query (i.e., before the user has entered the complete search strings). *E.g.*, EX1021, 126-127. EX1003, ¶91.

**g) Elements [1.11-1.13]**

The Lotus Notes+Raskin+Wu combination teaches these elements. The Lotus Notes references disclose locating files in an index data structure containing both the received search strings but do not expressly disclose doing so incrementally (i.e., locating files that contain words that begin with the partial/incomplete search strings as the user types in their search query) and doing so using an intersection approach (i.e., locating a first group of files that contain the first search string, locating a second group of files that contain the second search string, and then identifying the files included in both sets of files). EX1059, 159-168; EX1061, 134-135, 139-146. As noted above, however, Raskin discloses incremental searches (which were well

known, *see* Section IV.B.1 and EX1079 at 1007:20-22) and explains their known benefits, and Wu discloses the well-known intersection approach for locating files containing two search strings. EX1019, 6:51-53, 6:54-62 (“Search engine 36 looks up each of the terms in the word index 22. ... 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.”), 6:63-66, 8:66-9:10, 9:11-12 (“FIG. 6 is a flow chart of a process of ‘AND’ing two or more match lists to generate a new match list.”), 9:13-23, 9:24-32 (“The need for ‘AND’ing two or more match lists might come about where a search string contains an expression of the form ‘expression\_\_A AND expression\_\_B AND . . .’ . One match list is obtained for ‘expression\_\_A’ indicating the documents that contain that subterm, another for ‘expression\_\_B’, and so on. The resulting match list is a list of all the documents which contain all of the ‘AND’ed expressions . . .”), 9:33-11:32, FIGS. 5-6; EX1021, 125-127. EX1003, ¶92.

#### **Updating A Set Of Matching Files As Each Character Is Received (Performing An Incremental Search)**

As explained above in Section VII.D, a POSITA would have been motivated to combine the Lotus Notes references’ teachings on the Lotus Notes search tool with Raskin’s teachings to locate files incrementally for the reasons stated by Raskin: speed, effectiveness, and ease of use. EX1021, 125-127. Unlike a delimited

search in which a user has to wait until they have entered their full search query to view the search results, an incremental search, “immediately begins to search” as soon as the user “types the first character of the pattern” (EX1021, 125), and continues to update the search results as each character is added or deleted. EX1021, 125-126. Thus, incremental searches “provide[] constant feedback as to the results of the search,” thus “wast[ing] less of a user’s time” and allowing “the user to improve [their] search strategies through the feedback received.” EX1021, 126-127. EX1003, ¶93.

Specifically, the constant feedback allows the user to more quickly identify errors or problems with their search and to refine their query in real time to obtain the desired results more quickly than they would have with a delimited search. EX1021, 126-127. Moreover, if the user’s query is on point, the user does not have to type the full query before seeing the desired results. *Id.*, 126. 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, ¶94. Indeed, one of Patent Owner’s own experts admitted in the ITC proceeding that incremental searching was well known. EX1079 at 1007:20-22.

A POSITA would have understood that implementing Raskin's incremental search teachings involves treating the search strings as partial search strings/prefixes and locating files that contain words the begin with the search strings at least because POSITA knew that this is fundamentally how incremental searching works. EX1021, 125-127. EX1003, ¶95.

As also explained above in Section VII.D, a POSITA would have reasonably expected to succeed in combining these teachings from Raskin with Lotus Notes' teachings because it would have required only simple and straightforward software modifications. *E.g.*, *Keynetik*, WL 2003932 at \*2 (Fed. Cir. Feb. 15, 2023). EX1003, ¶96.

**Identifying Files Containing Both Search Strings By Separately Searching Each String And Identifying Files Present In Both Sets Of Search Results**

As explained above in Section VII.D, a POSITA would have been motivated to locate the files containing the two search strings by searching each string separately and then combining/intersecting the results, as taught by Wu, at least because it was one of a limited number of well-known, predictable approaches, and was simple and easy to implement. EX1019, 6:52-66, 8:66-9:32; *see also supra* Section IV.B.2. In particular, a POSITA would have recognized that there were a limited number of known ways (a few at most) to locate files containing both search strings, including either the claimed intersection approach, which searches each

string separately and then combines/intersects the results, a sequential approach, which sequentially checks each file for both search strings, or a sort of hybrid approach, which locates files containing one of the search strings and then searches within that set of files for files also containing the other search string. EX1003, ¶97.

Moreover, this intersection approach would have been a commonsense approach for locating files containing both search strings when performing an incremental search because a POSITA would have recognized that this approach's separate treatment of each string is well suited for an incremental search where a user can independently modify each string as they type in their query. EX1003, ¶98.

As also explained above in Section VII.D, a POSITA would have reasonably expected to succeed in combining these teachings from Wu with Lotus Notes' teachings because it would have required only simple and straightforward software modifications. *E.g.*, *Keynetik*, WL 2003932 at \*2 (Fed. Cir. Feb. 15, 2023). EX1003, ¶99.

### **Using An Index For The Search**

As explained above for element [1.5], the Lotus Notes references disclose using an index data structure to expedite the search. EX1059, 137-138, 159-160, 169-171; EX1061, 41, 135-139. EX1003, ¶100.

### **Including Files In Which The Search Strings Are Non-Adjacent**

Finally, a POSITA would have understood or at least found it obvious that the search strings are non-adjacent in at least one of the matching files at least because the POSITA understood that Lotus Notes' disclosed Boolean search cares only about whether the files contain the search strings, not whether the strings are adjacent or non-adjacent in the files. EX1059, 159-168; EX1061, 134-135, 139-146. Thus, a POSITA would have understood that Lotus Notes' disclosed Boolean search locates files containing the search strings at non-adjacent locations. EX1003, ¶101.

Moreover, it would have been obvious to locate files in which the search strings are non-adjacent at least because it was common/well known to do so, such as in proximity searches. *Supra* Section IV.B.2. EX1003, ¶102.

### **2. Claims 2-3**

The Lotus Notes+Raskin+Wu combination renders these claims obvious at least because, as explained above in Section VII.E.1.g, a POSITA would have understood that the Lotus Notes' disclosed Boolean search is order-agnostic: it cares only about whether a file contains the search strings, not the location of the search strings in the file. Thus, a POSITA would have understood or at least found it obvious that such a Boolean search would locate files where the first partial search string precedes the second partial string and vice versa. *Supra* Sections IV.B.2, VII.E.1.g. EX1003, ¶103.

### 3. Claim 4

The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose the ability to search within specific attribute fields (e.g., title, category, body, To, cc, bcc). EX1059, 100-101, 167-168; EX1061, 32-33, 142-144, 146-147.

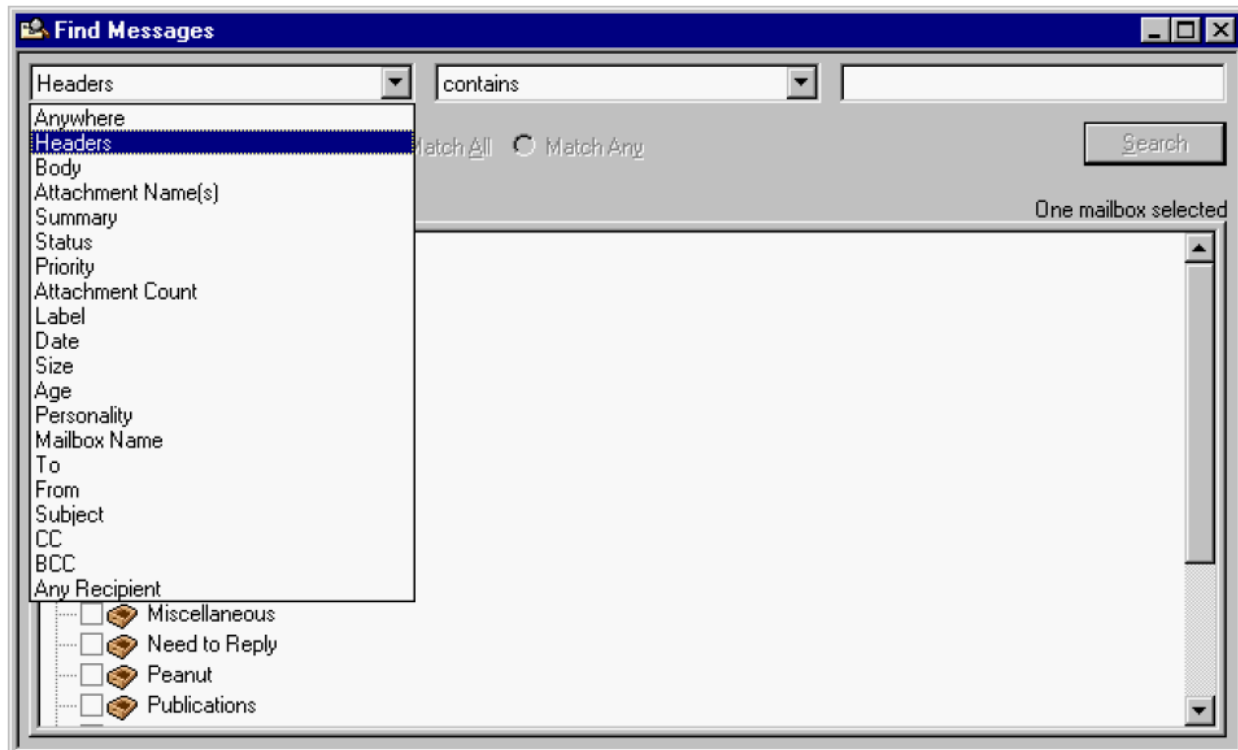
**Figure 5.21**      **A query created using a form**



EX1059, 168. EX1003, ¶104.

Moreover, it was well known to perform this kind of attribute field-specific search. *Supra* Section IV.B.5.

*Find Messages drop-down and conjunction options lists*



EX1007, 129. EX1003, ¶105.

To perform this kind of attribute field-specific searching, a POSITA would have understood or at least found it obvious that the index data structure would need to include indications of character strings associated with respective attribute fields of the files. EX1003, ¶106.

**4. Claims 5-7**

The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose searching all the files in a database (EX1059, 159-168; EX1061, 133-148) and disclose that the databases contain many different types of files, including emails having email-specific attribute fields

including at least a date field, a from field, and a sender field (EX1059, 93-134; EX1061, 21-44), documents such as word processing documents (EX1059, 321, 324; EX1061, 5-6, 188) having document-specific attribute fields including a file name field (EX1059, 321, 323; EX1061, 150-153, 160-161) and a date field (EX1059, 58-63, 176, 246-247, EX1061, 89, 141-142, 152), and email attachments including at least a name field (EX1059, 115-118; EX1061, 30, 150-151, 186). EX1003, ¶107.

A POSITA would have understood that users typically know what type of file they are looking for and would benefit from being able to search for only that file type because doing so would eliminate irrelevant files and return more tailored, digestible search results. Moreover, a POSITA would have understood that such file-type-specific searching would be advantageous for an incremental search where speed is important, because it would limit the number of files to be searched. Implementing this functionality would have required simple and straightforward software modifications, including creating and updating indexes for each type of file (i.e., an email index, a document index, and an email attachment index). *See* EX1059, 171; EX1061, 135-139 (describing generating and updating an index). Thus, a POSITA would have been motivated with a reasonable expectation of success to include email-, document-, and email attachment-specific search mode interfaces in the user interface. EX1003, ¶108.

**5. Claim 8**

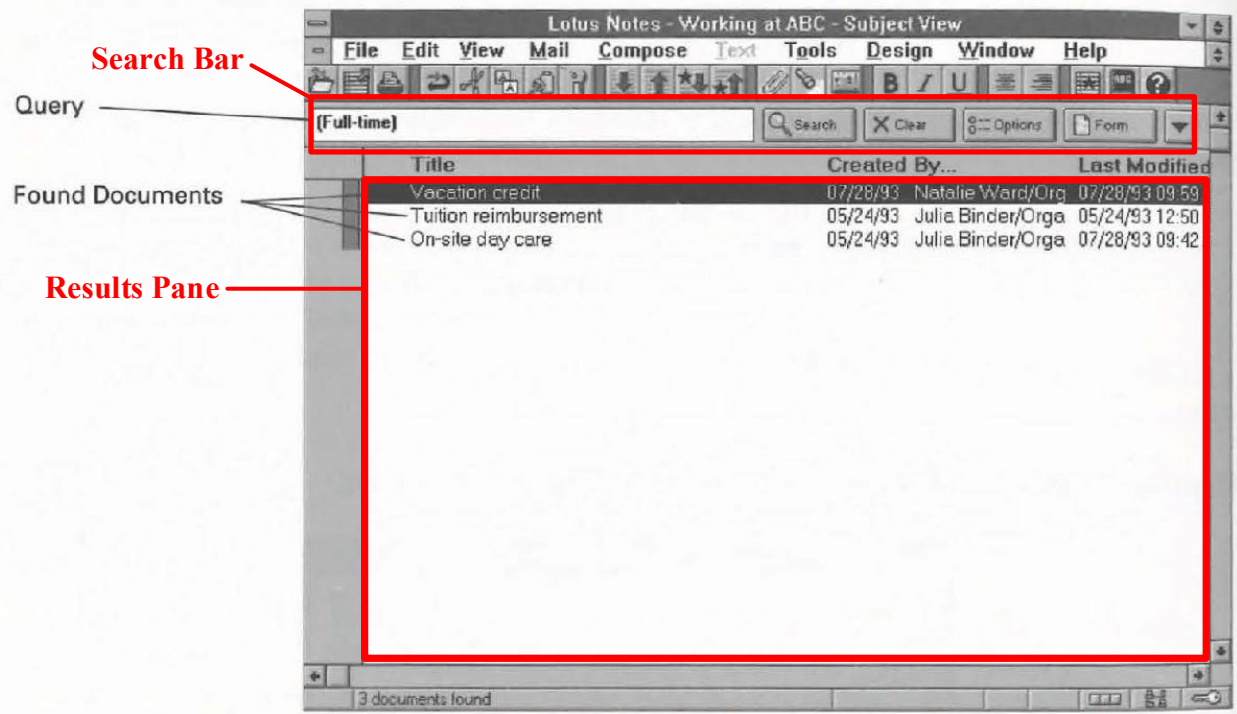
This claim is obvious at least because a POSITA understood that in an incremental search, a partial search string is a portion of string being searched for, such as the beginning of a word. *Supra* Sections VII.E.1.f-VII.E.1.g; EX1021, 125-127. EX1003, ¶109.

**6. Claim 9**

The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose that the computing system displays the set of matching files in the interactive user interface. EX1059, 84-85, 160-165; EX1061, 133-135, 139-140, 141 (“[C]lick Search, and wait for the proper documents to appear.”), 142-144, 145-146 (“Notes lists the documents that match your criteria in order of their relevance), 148. EX1003, ¶110.

For example, Lotus Notes displays the files containing the search strings in a results pane below the Search bar.

**Figure 5.17**      **The results of a query**



EX1059, 164 (red annotations added). EX1003, ¶111.

## 7. Claims 10-11

The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose displaying a portion of a selected file and highlighting occurrences of only the searched string. 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.”). 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 file, as was well known in the art. *See supra* Section IV.B.4. EX1003, ¶112.

#### **8. Claim 12**

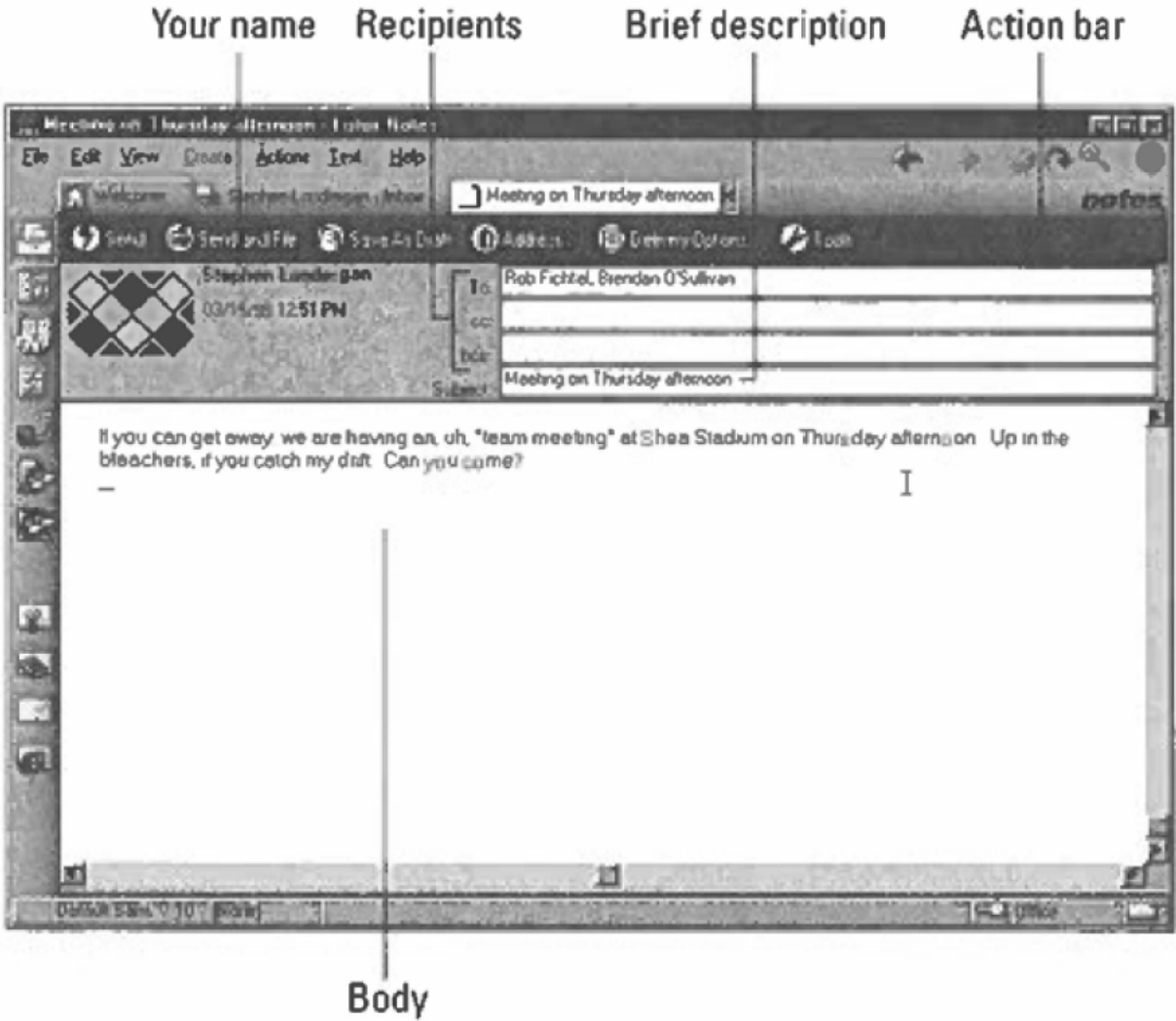
As explained above for claim element [1.10], the Lotus Notes references disclose that the string separator character includes a comma. EX1059, 159-165. Moreover, a POSITA would have found it obvious to use a plus sign (+) or a space in place of Lotus Notes' Boolean AND operator at least because it was well known to do so (*e.g.*, EX1007, 195; EX1019, 6:52-62, EX1020, 2:23-3:10; EX1024, 17; EX1057, 27, 60, 82, 89, 109, 129, 151) and because a POSITA knew that a goal of the incremental search is to minimize the number of characters the user has to enter before obtaining the desired search results. *E.g.*, EX1021, 125-127. Thus, the Lotus Notes+Raskin+Wu combination renders this claim obvious, EX1003, ¶113.

#### **9. Claim 13**

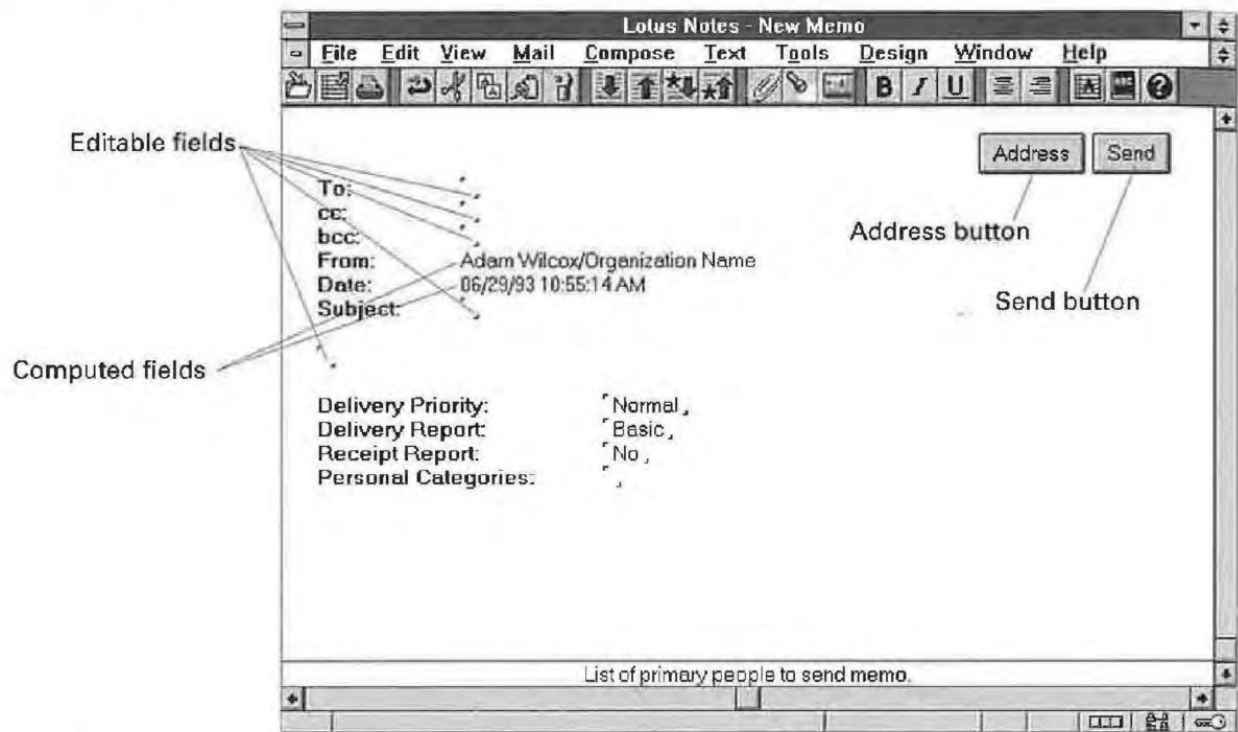
The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose that the files comprise, *inter alia*, emails (EX1059, 93-134; EX1061, 21-44), text documents (EX1059, 321, 324; EX1061, 5-6, 188), contact information (EX1059, 96, 121, 132; EX1061, 120-122, 147, 193) and appointment information (EX1061, 45-74). *Supra* Section VII.E.1.g. EX1003, ¶114.

**10. Claims 14-15**

The Lotus Notes+Raskin+Wu combination renders these claims obvious at least because as explained above for claim 4 (*supra* Section VII.E.3), the Lotus Notes references disclose that emails comprise title, category, body, To, cc, and bcc fields (EX1059, 100-112; EX1061, 31-35), and disclose the ability to search within specific fields of the files, such as by using a form (EX1059, 167-168; EX1061, 142-144, 146-147), and disclose searching within the “name, To, cc, and bcc fields in the memos in your email database” (EX1061, 147).



EX1061, 32.



EX1059, 102.

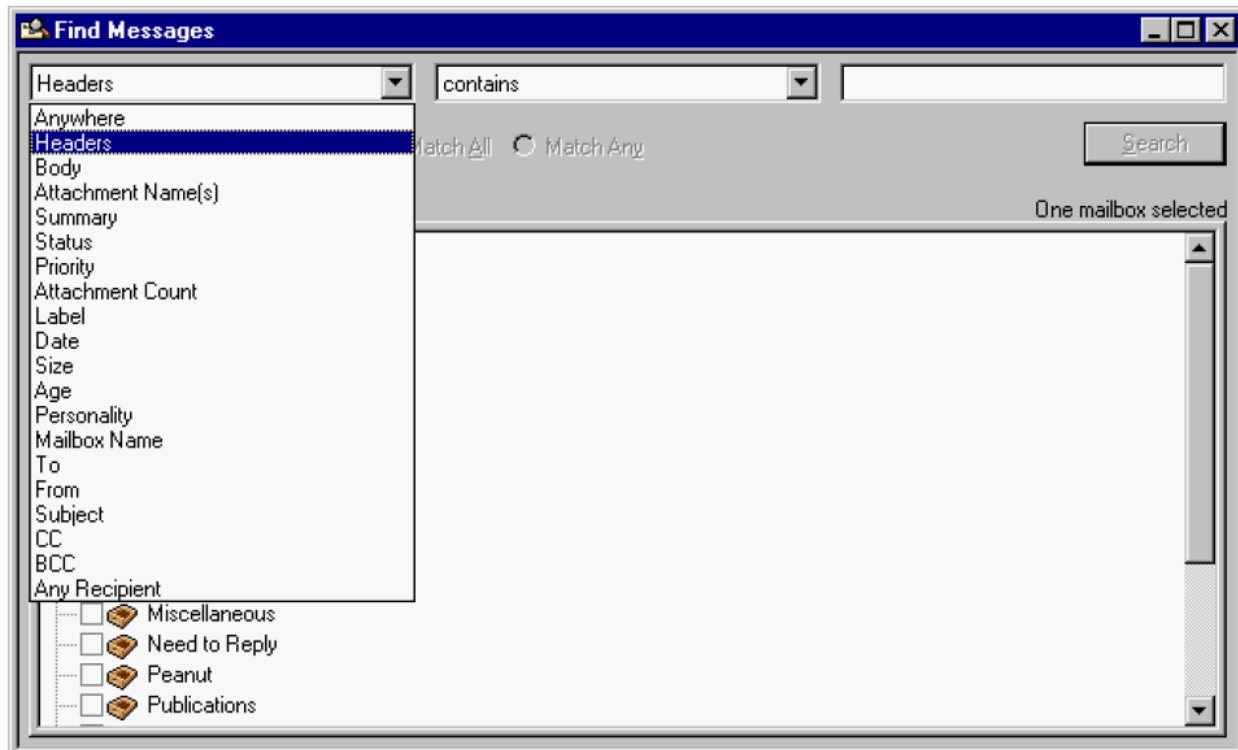
**Figure 5.21** A query created using a form



EX1059, 168. EX1003, ¶115.

Thus, it would have been obvious to a POSITA for the single search field to search within only one of the email fields, especially since this targeted search ability and its benefits were known in the art. *Supra* Section IV.B.5.

*Find Messages drop-down and conjunction options lists*



EX1007, 129. EX1003, ¶116.

**11. Claim 16**

The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose that the files comprise at least a name, date, and/or content (body) field (EX1059, 93-134, 321, 323; EX1061, 21-44, 150-153, 160-161, 186) of documents, and disclose the ability to search within specific fields of the documents, such as by using a form (EX1059, 167-168; EX1061, 142-144, 146-147).

**Figure 5.21**      **A query created using a form**

A screenshot of a search interface. It features a text input field containing the query "[(Body) CONTAINS (Full-time)]". To the right of the input field are four buttons: "Search" (with a magnifying glass icon), "Clear" (with an 'X' icon), "Options" (with a gear icon), and "Form" (with a document icon). A small dropdown arrow is visible on the far right of the button group.

EX1059, 168. EX1003, ¶117.

## 12. Claim 17

The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose that the files comprise name and email fields of contacts files in an address book (EX1059, 109-111; EX1061, 32-33, 147), and disclose the ability to search within specific fields of the documents, such as by using a form (EX1059, 167-168; EX1061, 142-144, 146-147).

**Figure 5.21**      **A query created using a form**

A screenshot of a search interface, identical to the one above. It features a text input field containing the query "[(Body) CONTAINS (Full-time)]". To the right of the input field are four buttons: "Search" (with a magnifying glass icon), "Clear" (with an 'X' icon), "Options" (with a gear icon), and "Form" (with a document icon). A small dropdown arrow is visible on the far right of the button group.

EX1059, 168. EX1003, ¶118.

## 13. Claim 18

The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose that the files comprise calendar files (e.g., appointment, event, meeting, etc.) comprising calendar fields, including, *inter alia*, subject, start date/time, recipients, and location fields (EX1061, 46-73), and disclose the ability to search within specific fields of the files, such as by using a form (EX1059, 167-168; EX1061, 142-144, 146-147).

**Figure 5.21**      **A query created using a form**



EX1059, 168. EX1003, ¶119.

**14.    Claim 19**

The Lotus Notes+Raskin+Wu combination references render this claim obvious at least because the Lotus Notes references disclose that the search defaults to searching all textual content of the files (a full-text search) (EX1059, 159-168; EX1061, 133-146) but that a user can also restrict the search for one of the search strings to apply to only a single field of the files, such as by entering a command in the search field (e.g., using a form) (EX1059, 167-168, EX1061, 141-144).

**Figure 5.21**      **A query created using a form**



EX1059, 168. EX1003, ¶120.

**15.    Claim 20**

The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose applying Boolean operators to either or both of the search strings. EX1059, 159-168; EX1061, 134-135, 139-146. EX1003, ¶121.

**VIII. GROUND 2: CLAIMS 1-20 ARE OBVIOUS  
OVER THE ENTOURAGE REFERENCES  
(EX1004-EX1006), TRUE (EX1009), AND BAEZA-YATES (EX1023)**

As explained below, claims 1-20 are obvious over the Entourage references (EX1004-EX1006) in view of True (EX1009) and Baeza-Yates (EX1023). EX1003, ¶122.

The Entourage references disclose or suggest every element of the challenged claims except two of the independent claims' features: (1) receiving two partial search strings in a single search field; and (2) locating indexed files containing the two partial search strings by searching each string separately in an index data structure and then identifying the files that appear in both sets of search results. Both features were well known in the art (*supra* Sections IV.B.2-IV.B.3), however, and True expressly discloses (1) and Baeza-Yates expressly discloses (2). EX1003, ¶¶123-124.

**A. Overview Of The Entourage References (EX1004-EX1006)**

The Entourage references—Schwartz (EX1004), Sebastian (EX1005), and Barber (EX1006)—are books describing aspects of the same Microsoft email system: Entourage 2001 for Macintosh (“Entourage”). EX1004, (i); EX1005, (a)-(c), 171-188; EX1006, (i), 1-3, 5, 9, 283-392. EX1003, ¶125.

The Entourage references qualify as prior art at least under Sections 102(a) and 102(b) (pre-AIA) because they were published and publicly accessible more

than one year before the earliest alleged priority date of the '490 patent (September 3, 2002). EX1004, (ii) (publication date); EX1005, (c) (same); EX1006, (iii) (same); EX1060, ¶¶48-75, 135. Moreover, at least Schwartz and Barber were published by established publishers (Peachpit Press (a division of Addison Wesley and an imprint of Pearson Education) and Pogue Press/O'Reilly & Associates, Inc.) and Schwartz was indexed and available for purchase on the publisher's searchable website more than one year before the '490 patent's earliest alleged priority date, and Barber was available for purchase on Amazon more than one year before the '490 patent's earliest alleged priority date. EX1038, 77-103; EX1060, ¶¶55, 74 ; *VidStream LLC v. Twitter, Inc.*, 981 F.3d 1060, 1065 (Fed. Cir. 2020).

The Entourage references are analogous art to the '490 patent because they are from the same field of endeavor: systems and methods for providing a user interface through which a user can enter search strings, and locating files (e.g., emails) containing the search strings. *E.g.*, EX1001, 1:38-39, 2:20-32, 3:46-4:7, 4:41-48; FIGS. 3A, 3G-3I; EX1004, 2-12, 161-164; EX1005, 173; EX1006, 283-285, 384-387. EX1003, ¶126.

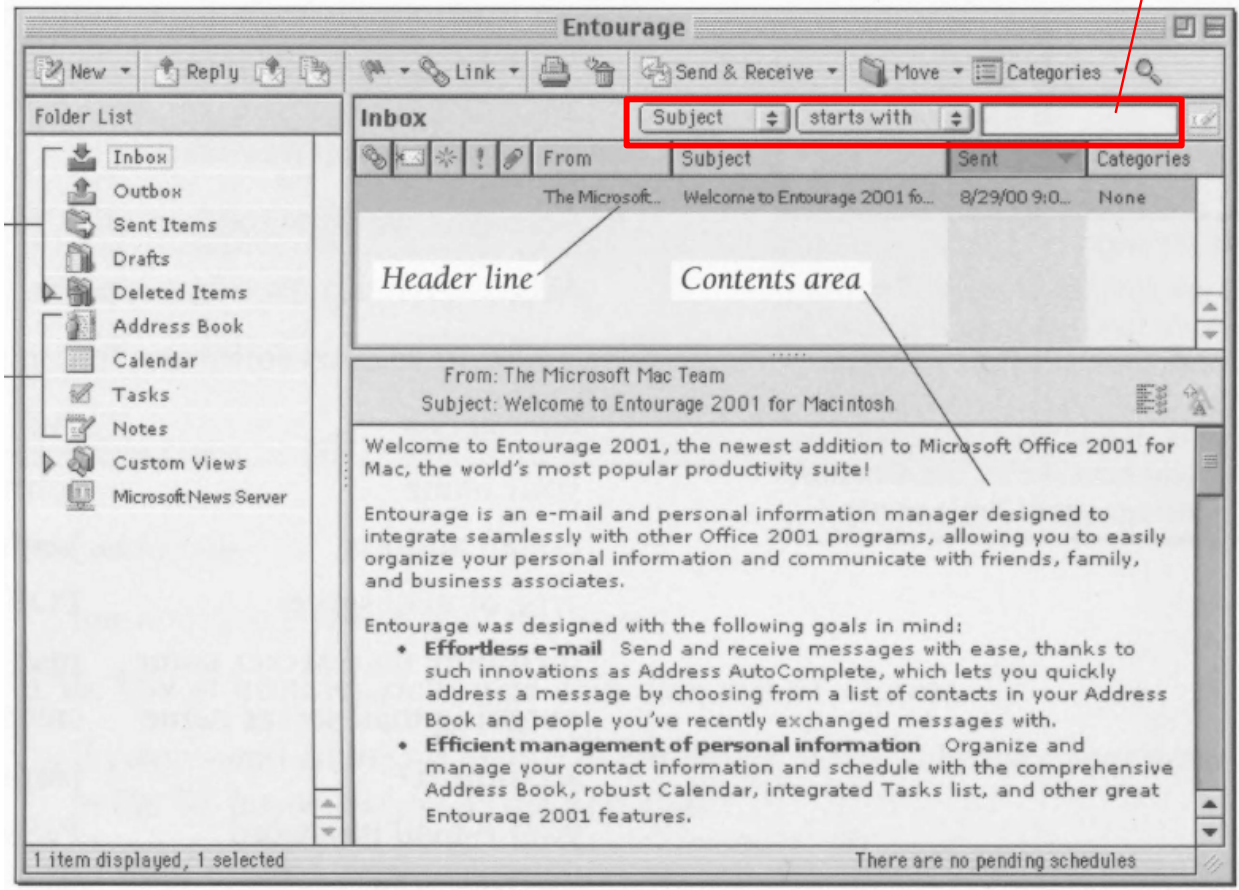
The Entourage references explain how Entourage works and how to use it. EX1004, *passim*; EX1005, 171-188; EX1006, 283-392. As these references explain, Entourage comprises six main components: an email client, newsgroup reader, address book, calendar, tasks, and notes. *E.g.*, EX1004, 2; EX1006, 281. Users can

quickly locate files (e.g., emails, news articles, tasks, contacts) containing specific text using an incremental, quick-find search, accessible at the top of a main window (e.g., the main window of an email Inbox folder). EX1004, 8-14, 38-39, 89-90, 136, 161-164, 204-205, 209; EX1005, 173; EX1006, 384-387. EX1003, ¶127.

Results are displayed “[a]s you type the filter text.” EX1004, 9. Thus, the incremental search will display results for prefixes (words that “[s]tart[] [w]ith” the user-entered search string), such that the user does not have to type out the entire word. *E.g.*, EX1004, 161; EX1005, 173; EX1006, 384-386. The incremental, quick-find search also allows a user to search within specific email headers (To, From, Subject, Category). *E.g.*, EX1004, 8, 161; EX1005, 173.

## Entourage's Incremental, Quick-Find Search

Search Field



EX1005, 173 (red annotations added).

### Entourage's Incremental, Quick-Find Search

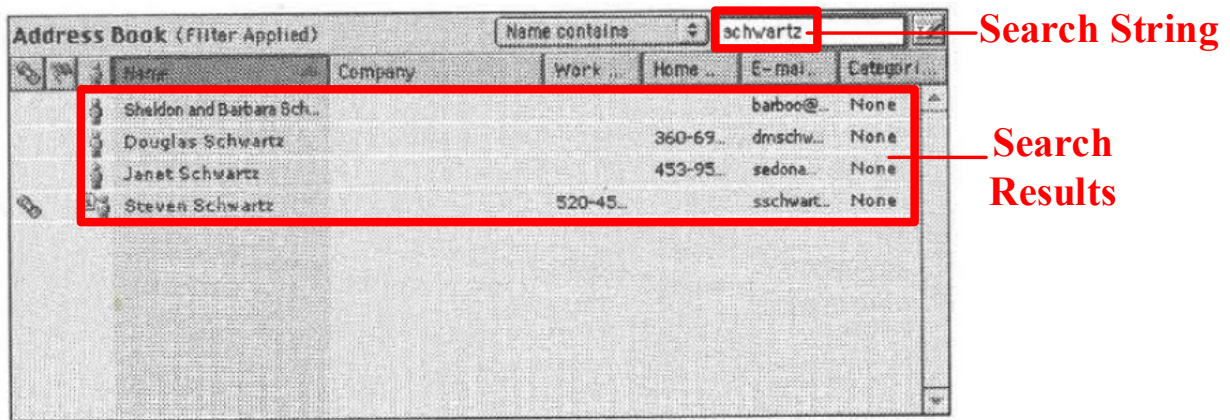


**Figure 1.12** As you type the filter text, only matches are shown in the list.

EX1004, 9 (red annotations added); *see also id.*, 38. EX1003, ¶¶128-130.

After Entourage finds a file (e.g., email) containing the search string(s), Entourage scrolls to the portion of the file 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.

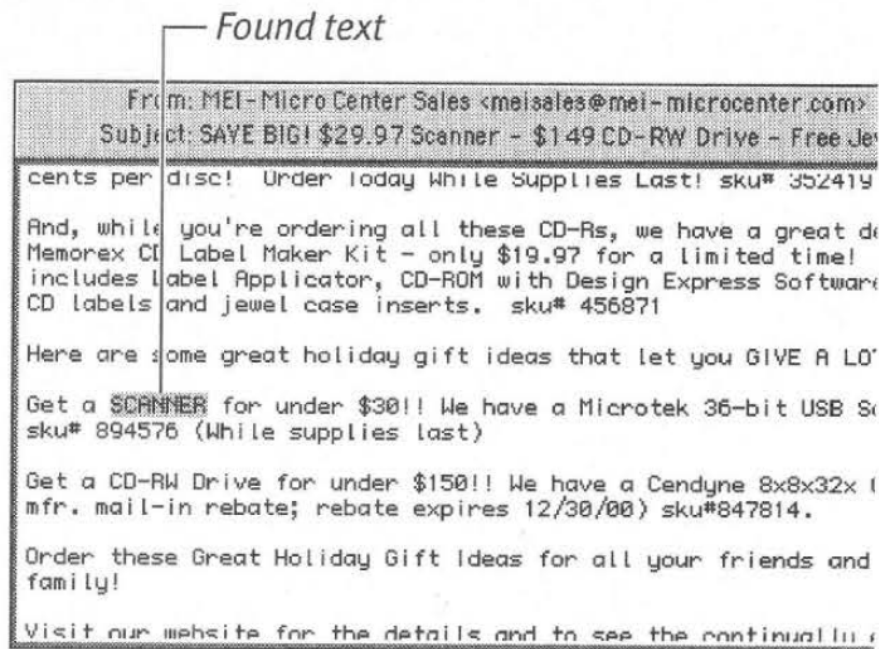


**Figure 11.20** The first instance of the found text—if there is one—is highlighted for you.

EX1004, 163. EX1003, ¶131.

## **B. Combining The Entourage References**

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. 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, ¶132.

A POSITA also would have reasonably expected to succeed in combining the teachings of these three references at least because they describe the same version of Entourage (Entourage 2001) for the same computing system (Macintosh). *E.g.*, EX1004, (i); EX1005, (a)-(c), 171-188; EX1006, (i), 1-3, 5, 9, 283-392. EX1003, ¶133.

**C. Overview Of The True (EX1009)  
And Baeza-Yates (EX1023) References**

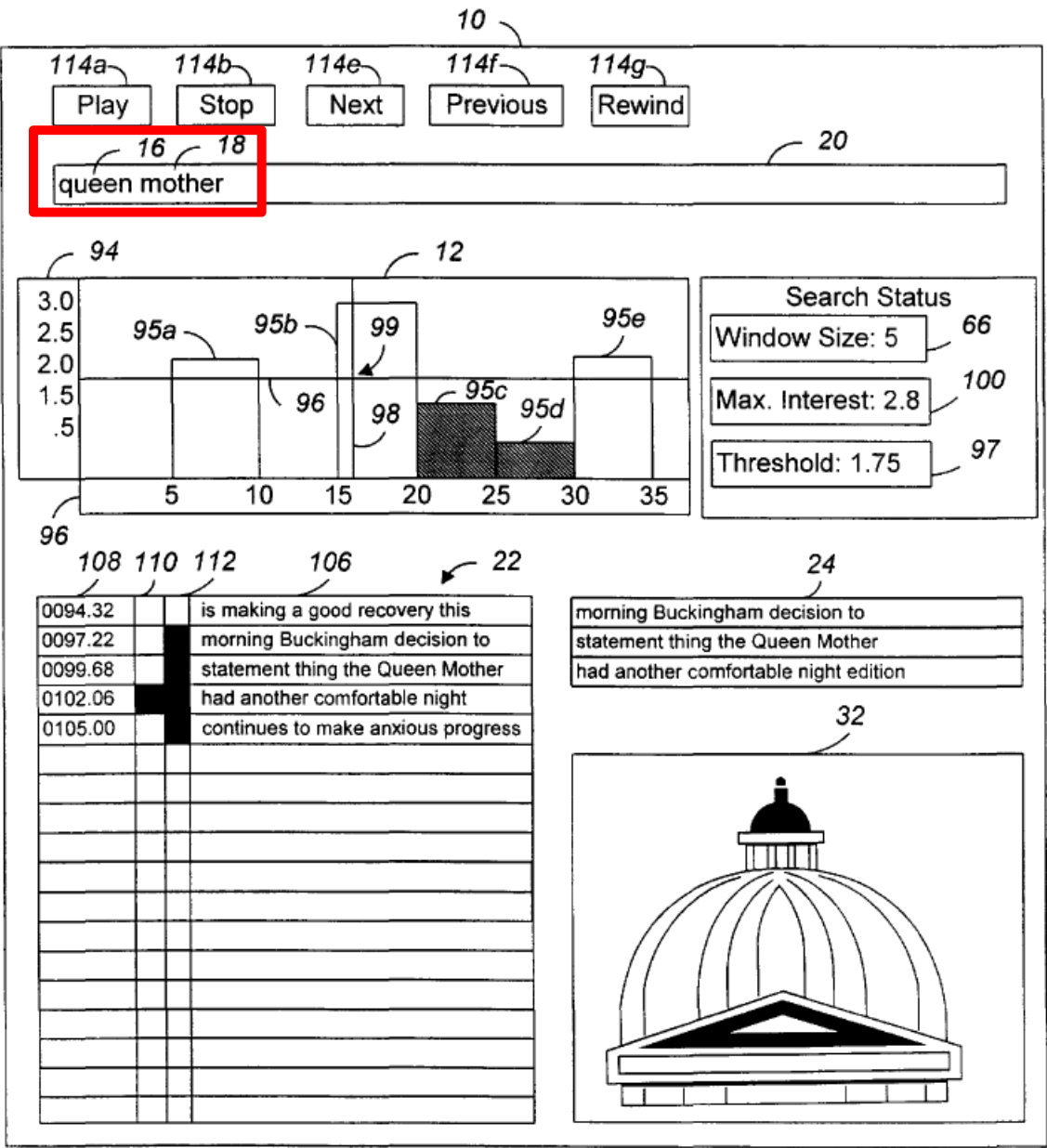
True and Baeza-Yates are prior art under at least Sections 102(a) and 102(b) (pre-AIA) because they published and were publicly accessible more than one year before the '490 patent's earliest alleged priority date (September 3, 2002). EX1009, Cover; EX1023, (iv) (copyright date); EX1060, ¶¶90-97, 135.

True and Baeza-Yates are analogous art to the '490 patent because they are from the same field of endeavor: systems and methods for providing a user interface through which a user can enter search strings, and locating files and/or portions of files containing the search strings. *E.g.*, EX1001, 1:38-39, 2:20-32, 3:46-4:7, 4:41-

48; FIGS. 3A, 3G-3I; EX1009, Abstract, 1:37-2:61, 3:19-5:51, FIGS. 1-2; EX1023, (i), (v), 11-14. EX1003, ¶134.

True is an issued U.S. patent that discloses incrementally searching for, and locating, portions of an audio transcript containing multiple search strings (e.g., “queen” and “mother”) entered in a single search field and displays “query results ... as the query is being input” (EX1009, 2:41-42). EX1009, 2:26-48, 4:38-5:6, 5:41-50, FIG. 2.

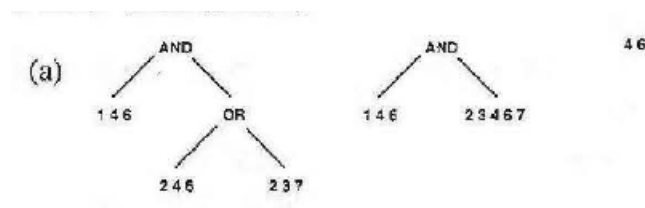
FIG. 2



EX1009, FIG. 2 (red annotation added). EX1003, ¶135.

Baeza-Yates is a textbook published in 1999, primarily intended for undergraduate information retrieval (IR) students (EX1023, (v)). Among many other

IR topics, Baeza-Yates discloses using indexes (e.g., EX1023, 6, 191) to locate files containing two search strings by locating a first set of files containing at least one word that begins with the first string and locating a second set of files containing at least one word that begins with the second string and then combining the results (i.e., identifying the files present in the first and second sets of files), as was well known in the art. EX1023, 101-103, 195, (“[P]hrases and proximity queries are split into single words. ... The lists of the occurrences of all the words found are retrieved. ... The occurrences are processed to solve ... Boolean operations.”), 196 (“Each element must be searched separately and a list ... generated for each one. Then, the lists of all elements are traversed in synchronization to find places where all the words ... appear close enough (for proximity).”), 207-208; *see also supra* Section IV.B.2 (citing, e.g., EX1019, 6:52-62, 9:11-12, 9:24-32; EX1024, 17; EX1025, 109-110).



EX1023, 208 (showing identifying a first set of files (1, 4, and 6) containing a first search string and identifying a second set of files (2, 3, 4, 6, and 7) containing a second search string and then identifying the files (4 and 6) that appear in both sets of files). EX1003, ¶136.

**D. Combining Entourage With The True And Baeza-Yates References**

The Entourage+True+Baeza-Yates combination combines the Entourage references' teachings to incrementally locate files, including emails, containing at least one search string entered in a search field with True's teachings to do so for multiple search strings entered in the search field and Baeza-Yates teachings to do so by searching each string separately in an index data structure and then intersecting the results. EX1003, ¶137.

A POSITA would have been motivated to combine the Entourage references' teachings to incrementally search at least one search string entered in a single search field with True's express teachings to do so for multiple search strings entered in the search field at least because a POSITA knew that such a multi-string search would return more relevant results than a single-string query. *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, ¶138.

Further, when incorporating True's teachings to search for multiple search strings, a POSITA would have been motivated to include Baeza-Yates's teachings to locate the files containing the two search strings by searching each string separately and then intersecting the results at least because this was one of only a

limited number (a few at most) of known ways to locate files containing multiple search strings, was simple and easy to implement, and was especially well suited for an incremental search. EX1023, 25-26, 101-105, 195-196, 207-208. EX1003, ¶139.

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, ¶140.

A POSITA also would have been motivated with a reasonable expectation of success to use an index data structure to speed up the search process, as disclosed by Baeza-Yates (*e.g.*, EX1023, 6, 191) and well known in the art (*supra* Section IV.B.3). EX1003, ¶141.

**E.     The Entourage+True+Baeza-Yates  
Combination Teaches Every Claim Element**

**1.     Claim 1**

**a)     Element [1.1]**

The Entourage references teach a computing system (a Macintosh computer) on to which a user installs/downloads the Entourage software. EX1004, (i) (“ENTOURAGE FOR MACINTOSH”), 23, EX1006, 1, 5, 9, 613 (“There are two

ways to install Microsoft Office 2001 on your Mac”), 614-618; *see also* EX1043, 573-576. EX1003, ¶142.

**b) Elements [1.2-1.4]**

The Entourage references teach these claim elements: a processor (e.g., a chip), a display device (e.g., a monitor), and a computer readable storage medium (e.g., a memory, hard drive, disk). EX1004, 23, 30, 35-36, 54-55; EX1006, 1, 5, 9, 613-618; *see also* EX1043, 573-576. EX1003, ¶¶143-144.

Moreover, a POSITA would have found it obvious that the computing system comprises a processor, memory, and a display at least because it was well known that these were necessary components of Macintosh and other personal computing systems. EX1003, ¶145.

**c) Element [1.5]**

Although the Entourage references do not expressly disclose indexing the computing system’s files in 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, Baeza-Yates does. EX1023, 5-27, 191-228. A POSITA would have been motivated to incorporate Baeza-Yates’ teachings to implement an index data structure to expedite the search process, as taught by Baeza-Yates and was well known in the art. *Supra* Section IV.B.3. EX1003, ¶146.

“[I]ndexes are at the core of every modern information retrieval system. They provide faster access to the data and allow the query processing task to be speeded up.” EX1023, 6. *See also, e.g.*, EX1054, 1:19-22 (“For some time now, existing systems have provided the capability to index the contents of documents. These systems typically store indexing data in a data structure so that it can be accessed and manipulated to answer user-generated queries efficiently.”); EX1055, 241 (“Just as the index of a book helps to locate words in the book, an index enables data to be retrieved more quickly than by searching the whole database file from the beginning ... .”); EX1059, 159 (“For a database to support full-text searches, it must have a current *full-text index*.”), 160, 169-171; EX1061, 135-139. EX1003, ¶147.

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 been motivated to create an index data structure comprising associations between each of a plurality of character strings to associated files in which the character strings are included to expedite the search process. EX1003, ¶148.

A POSITA would have reasonably expected to success in implementing Baeza-Yates’ indexing teachings in Entourage’s search tool at least because a POSITA would have understood that Entourage’s search tool already includes such indexing functionality and because implementing such functionality would have

required only simple and straightforward software instructions. *See Keynetik*, WL 2003932 at \*2 (Fed. Cir. Feb. 15, 2023). EX1003, ¶149.

**d) Element [1.6]**

The Entourage references teach this claim element at least because they disclose software instructions (the software that runs Microsoft Office 2001, including Entourage) that are downloaded to the computing system's memory. EX1004, 23 (“Installing the software from the Microsoft Office 2001 CD-ROM” to “[s]et[] up Entourage.”), 30, 35-36, 54-55; EX1006, 1, 5, 9, 613-618; *see also* EX1043, 573-576. A POSITA would have understood or at least found it obvious that the software is executed by the processor (chip) of the computing device (Macintosh computer) to perform one or more operations (e.g., the claimed operations in elements [1.7-1.13] below) at least because this was a basic and fundamental operating principle of computers that a POSITA was well aware of. EX1003, ¶150.

**e) Element [1.7]**

The Entourage references teach displaying an interactive user interface that includes at least a first search field on the display device. EX1004, 2-3, 7-14, 38-39, 89-90, 136, 161-164, 204-205, 209; EX1005, 173; EX1006, 384-387; *see also* EX1043, 83-84, 337-338, 343-344, 355, 385-386, 404. EX1003, ¶151.

For example, the Entourage reference disclose interactive user interfaces including at least a first search field for searching emails and contacts:

## Searching Emails

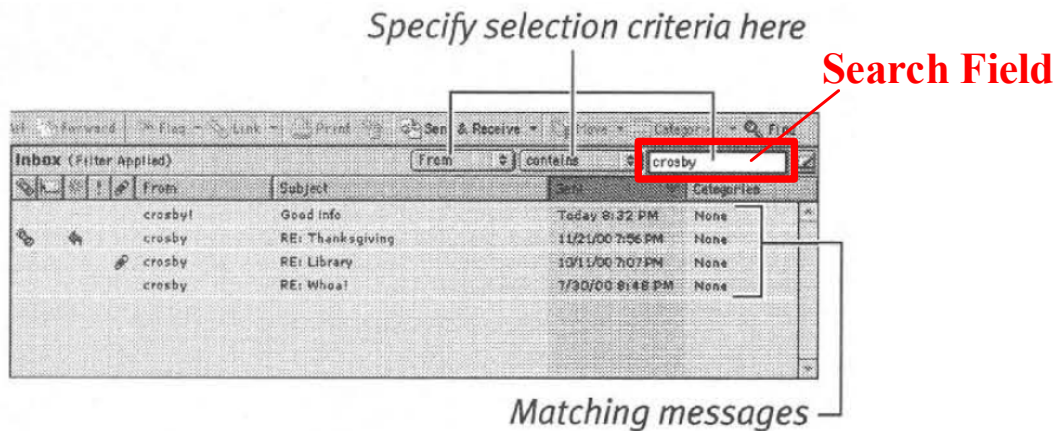


**Figure 11.15** To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161 (red annotations added).

## Searching Contacts

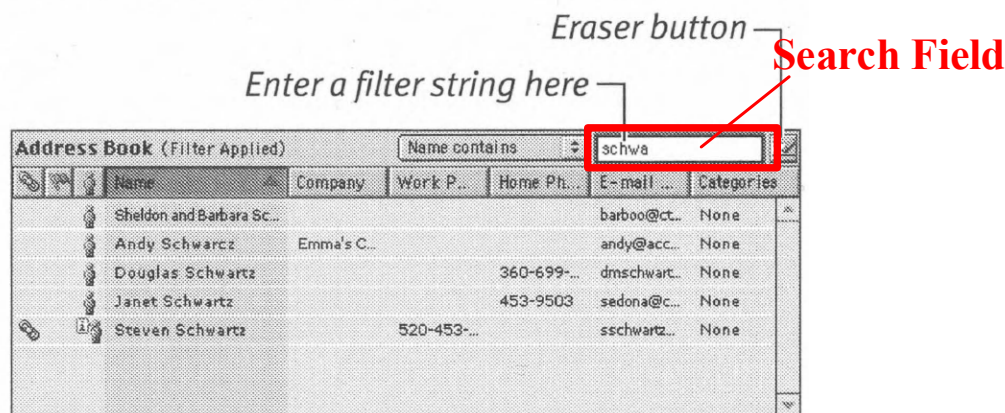


**Figure 3.4** You can limit the display to records that contain a particular text string.

EX1004, 38 (red annotations added). EX1003, ¶152.

**f) Elements [1.8-1.10]**

The Entourage references in combination with True teach these claim elements, especially in view of the knowledge of a POSITA reflected in the numerous background references described above in Section IV.B.2. EX1004, 2-3, 7-14, 38-39, 89-90, 136, 161-164, 204-205, 209; EX1005, 173; EX1006, 384-387; EX1009, 2:26-31, 2:39-48, 4:38-43, 4:65-5:6, 5:41-50, FIG. 2; *supra* Section IV.B.2. EX1003, ¶153.

For example, the Entourage references expressly disclose receiving at least one search string (e.g., the start or all of a person’s last name, such as “schwa” or “schwartz”) (EX1004, 9, 38) in a search field, and at least suggest receiving two search strings (e.g., a person’s first and last names, such as “Orson Welles”)<sup>8</sup> (e.g., EX1006, 387) in the search field separated by a string separator character (e.g., a space)<sup>9</sup> (EX1006, 387). *See also* EX1004, 267; EX1043, 327, 338, 343-344, 379-380, 385-389, 402-409. The Entourage references also expressly disclose “searching for text strings” (plural), at least suggesting to a POSITA to receive multiple search

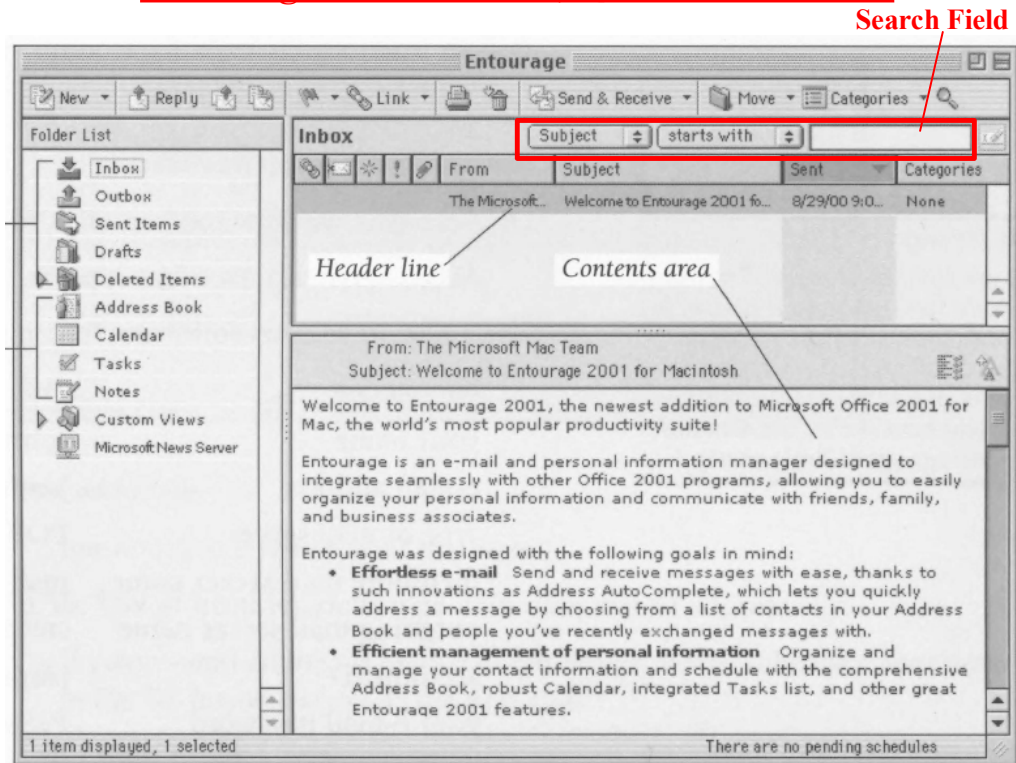
---

<sup>8</sup> *See* fn. 6 (explaining that two words separated by a string separator character (e.g., a space) are two search strings).

<sup>9</sup> *See* fn. 7 (explaining that a string separator character separates two strings).

strings in the single search field of Entourage's incremental Quick Find search.  
EX1004, 267, 269 (emphasis added).

### Entourage's Incremental, Quick-Find Search



EX1005, 173 (red annotations added).

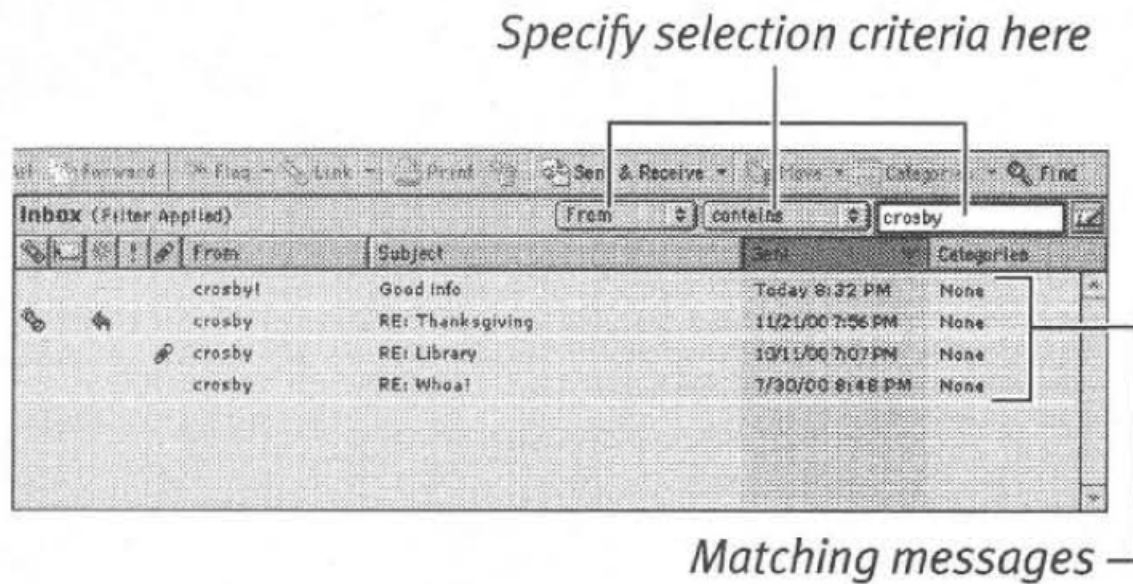


**Figure 11.15** To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161. EX1003, ¶154.

Moreover, a POSITA would have been motivated to incorporate True's express teachings to receive two search strings in the search field (EX1009, 2:26-31, 2:39-48, 4:38-43, 4:65-5:6, 5:41-50, FIG. 2) to provide more targeted searching, as was common sense and well known in the art. *Supra* Section IV.B.2; 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."). A POSITA would have reasonably expected to succeed in incorporating this functionality from True in the Entourage search tool at least because the Entourage search tool already includes the same or

substantially similar functionality and would have required only simple and straightforward software modifications. *See, e.g., Keynetik*, WL 2003932 at \*2 (Fed. Cir. Feb. 15, 2023). EX1003, ¶155.

A POSITA also would have found it obvious to treat Entourage’s space (e.g., the space between “Orson” and “Welles”) as a string separator character—and thus treat the two keywords separated by the space (e.g., “Orson” and “Welles”) as separate search strings and search them separately—at least because it was common to do so. *E.g.*, EX1007, 195; EX1019, 6:52-62, EX1020, 2:23-3:10 (“[T]he user may enter the phrase ‘merced microprocessor.’ ... [T]he phrase ... may be parsed such that the word ‘merced’ creates a first set of hits while the word ‘microprocessor’ creates a second set of hits. In the absence of any Boolean operator, as is in the case of FIG. 1, it may be assumed, for example, that the user wishes to find all web pages or web sites that contain both the term ‘merced’ and the term ‘microprocessor.’”); EX1024, 17 (“If the user types multiple keywords in the query, the search engine searches each of the keywords independently and combine [sic] all the results to generate the final answer.”); EX1057, 27 (“[S]ome search engines default to doing an AND search . . .”), 82 (“Google defaults to doing an AND search ... so plus signs (+) aren’t necessary.”), 89. EX1003, ¶156.

Moreover, it was well known to use various other types of string separator characters, including a Boolean AND operator, a comma, and a plus sign, to instruct

the search tool to treat two words entered in a search field as separate search strings (and thus search them separately), and to only locate files containing both words (whether they appear adjacent or non-adjacent in the files). *E.g.*, EX1009, 1:6-10, 4:38-42, FIG. 2; EX1019, 6:51-66, 8:66-11:32, 13:26-30 (claim 5), FIG. 6; EX1023, 102-103 (“The oldest (and still heavily used) form of combining keyword queries is to use Boolean operators. ... The operators most commonly used, given two basic queries or Boolean subexpressions  $e_1$  and  $e_2$  are: ... **AND** The query ( $e_1$  AND  $e_2$ ) selects all documents which satisfy both  $e_1$  and  $e_2$ .”); EX1057, 60 (“Use a plus sign (+) in front of each word or phrase that must appear in the results”), 129 (same), 109 (same), 151 (To search “multiple words and phrases, each of which must be present ... [y]ou can also use plus signs (+) or AND operator.”), 166; EX1059, 161-162, 163 (“[T]o search for documents that contain all of several words, type them all in this box, separated by commas.”). EX1003, ¶157.

Thus, even if the Entourage references do not expressly disclose receiving two search strings in the same search field (which Petitioner does not acquiesce is appropriate), a POSITA still would have found it obvious to include this basic and well-known functionality—described, for example, by True—to allow a user to enter more targeted search queries and thereby receive more relevant search results. EX1003, ¶158.

**g) Elements [1.11-1.13]**

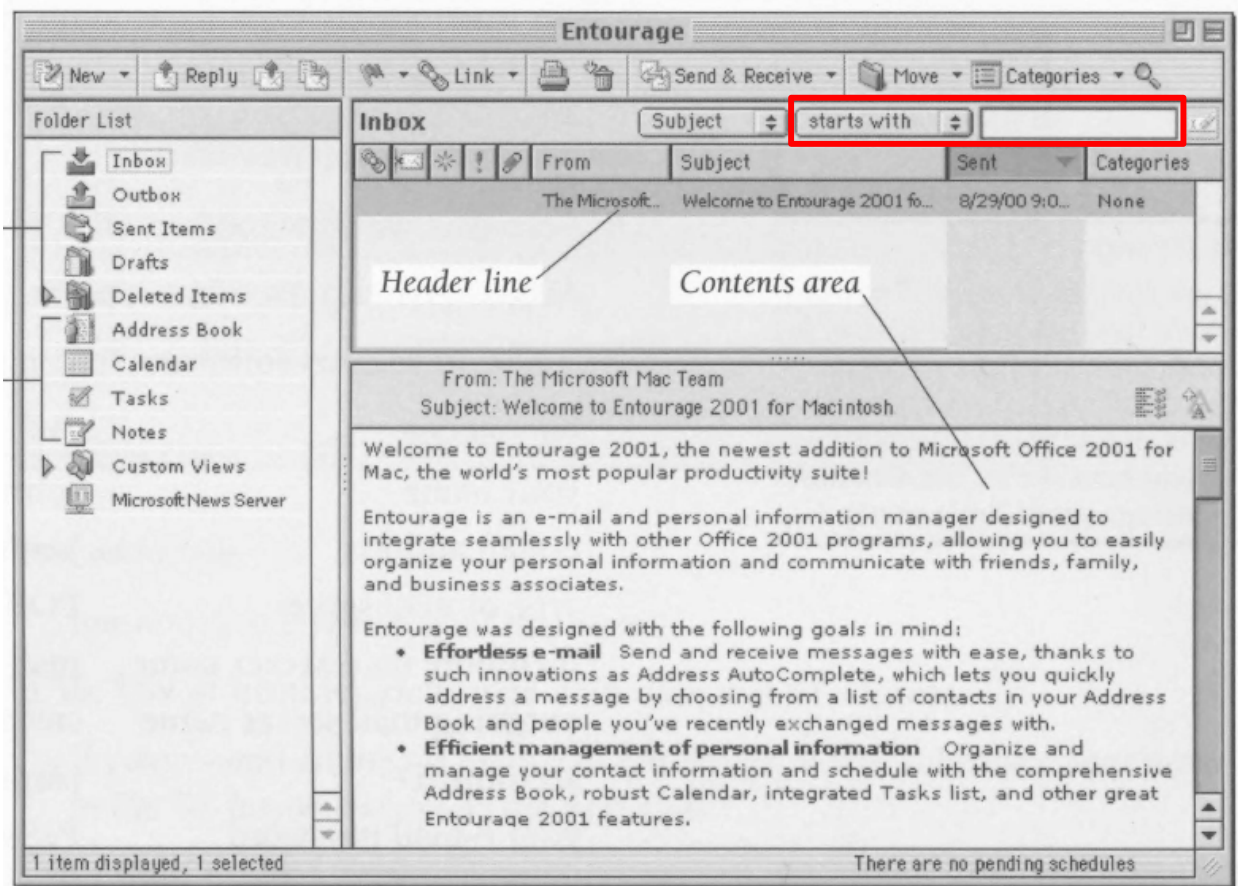
The Entourage references in combination with True and Baeza-Yates teach these claim elements, especially in view of the knowledge of a POSITA reflected in the numerous background references described above in Sections IV.B.1-IV.B.3. EX1004, 2-3, 7-14, 38-39, 89-90, 161-164, 204-205, 209; EX1005, 173; EX1006, 358-384; EX1009, 2:26-31, 2:39-48, 4:38-43, 4:65-5:6, 5:41-50, FIG. 2; EX1023, 25-26, 101-105, 195-196, 207-208; *supra* Sections IV.B.1-IV.B.3; *see also* EX1043, 83-84, 337-338, 343-344, 355, 385-386, 404. EX1003, ¶159.

**Updating A Set Of Matching Files As Each Character Is Received  
(Performing An Incremental Search)**

The Entourage references teach what was generally well known in the art: incrementally locating and displaying files<sup>10</sup> (e.g., contacts, emails, news articles, tasks, etc.) containing partial search string(s) as each successive character of the partial search strings is received. EX1004, 8, 9 (results are shown “[a]s you type the filter text”), 10, 38-39, 89, 161, 204-205, 209; EX1005, 173; EX1006, 358, 384; *supra* Section IV.B.1; EX1079 at 1007:20-22; *see also* EX1043, 83-84, 337-338, 343-344, 355, 385-386, 404.

---

<sup>10</sup> Claim 13 of the ’490 patent recites that “the files comprise emails, text documents, contact information, appointment information, [and] task information ... .” EX1001, 31:18-21.



EX1005, 173 (red annotation added).

Address Book (Filter Applied) Name contains schartz

| Name                       | Company | Work      | Home      | E-mail      | Category |
|----------------------------|---------|-----------|-----------|-------------|----------|
| Sheldon and Barbara Sch... |         |           |           | barboo@...  | None     |
| Douglas Schwartz           |         |           | 360-69... | dmschw...   | None     |
| Janet Schwartz             |         |           | 453-95... | sedona...   | None     |
| Steven Schwartz            |         | 520-45... |           | sschwart... | None     |

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

EX1004, 9.

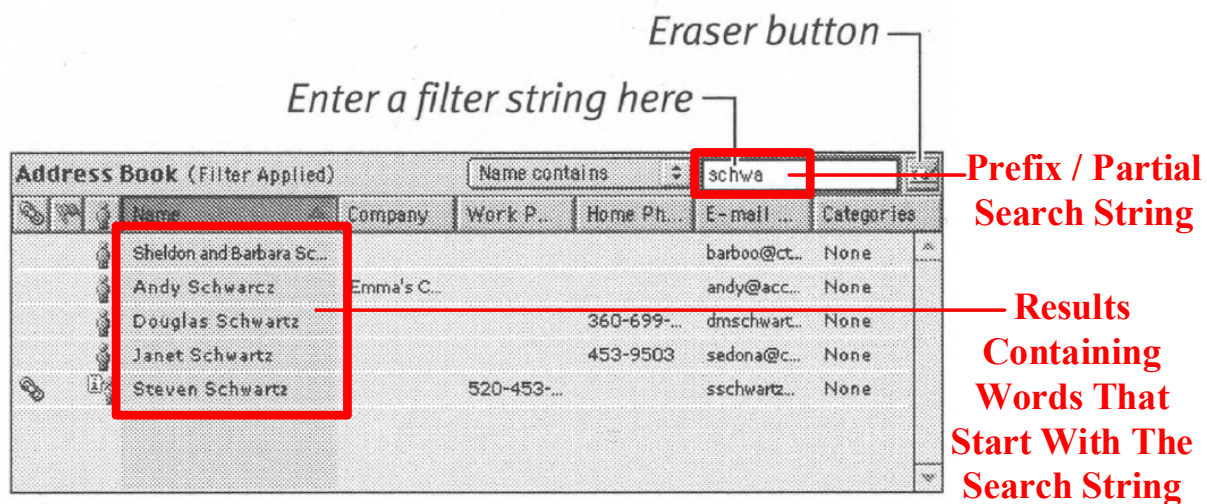


**Figure 3.4** You can limit the display to records that contain a particular text string.

EX1004, 38 (red annotations added). EX1003, ¶160.

### Searching For Multiple Strings

The Entourage references also disclose “searching for text strings,” (EX1004, 267, 269; *see also* EX1006, 384, 387) plural, which at least suggests incrementally searching for files containing two partial search strings entered in a single search field in view of the Entourage references’ express teachings to perform an incremental search. EX1004, 8-10, 38-39, 89, 161, 204-205, 209; EX1005, 173; EX1006, 358, 384; EX1043, 83-84, 337-338, 343-344, 355, 385-386, 404; *see also* EX1006, 384, 387 (disclosing performing a search on a query comprising multiple words entered in a single search field). Moreover, it would have been obvious to a POSITA to search for multiple partial search strings entered in a single search field

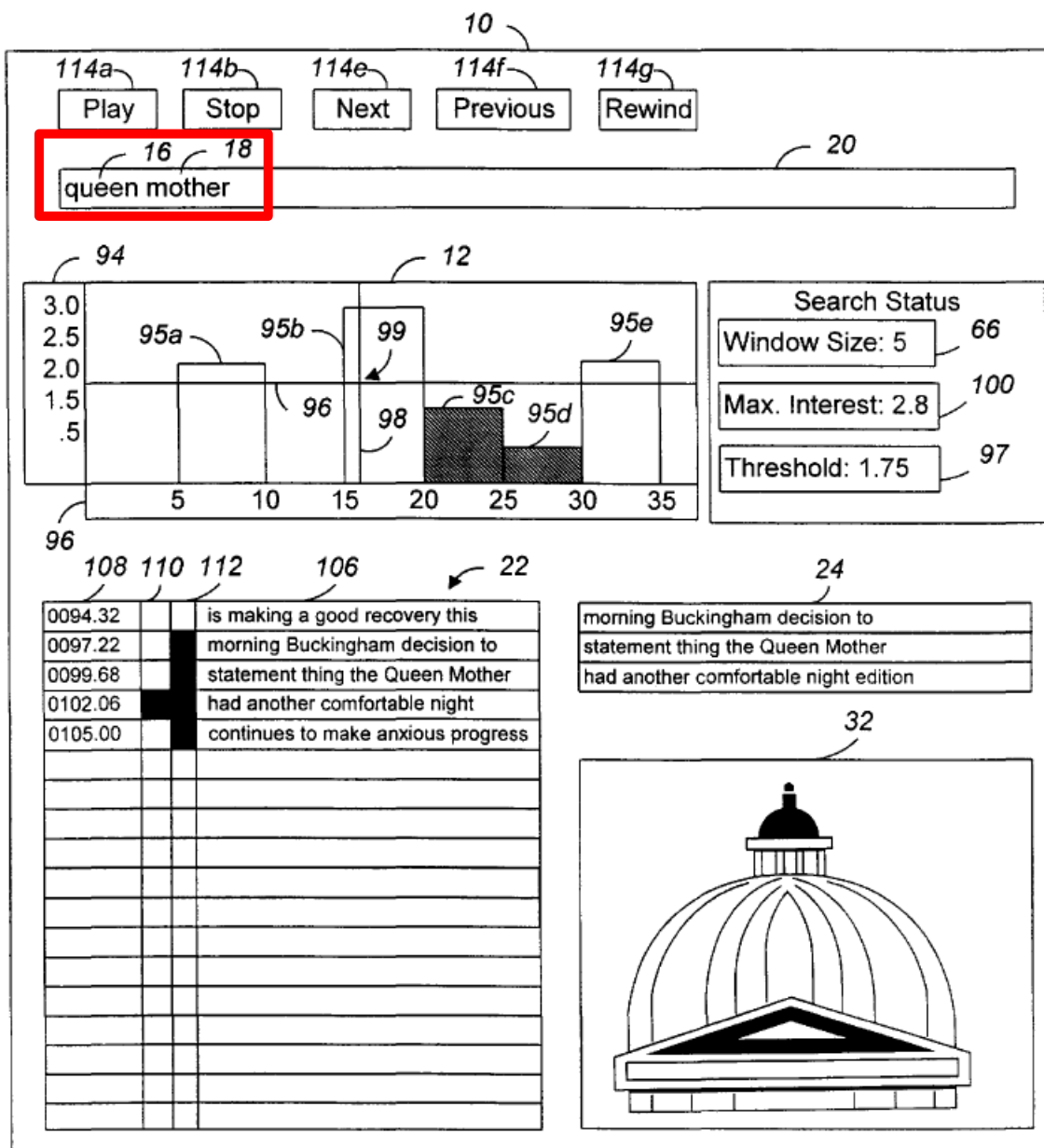
at least because it was within a POSITA's knowledge to do so, as reflected in the numerous background references cited above in Sections IV.B.2 (state of the art) and VIII.C.1.f (elements [1.8-1.10]). EX1003, ¶161.

Moreover, a POSITA would have been motivated to incorporate True's teachings to incrementally locate (i.e., locate as the user types or modifies their search query) files containing multiple search strings entered in the same search field (e.g., files containing the strings "queen" and "mother") to provide more targeted searching, as was common sense and well known in the art. EX1009, 2:26-31, 2:39-48, 4:38-43, 4:65-5:6, 5:41-50, FIG. 2; EX1057, 27 ("AND Searches [] 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."); *supra* Sections IV.B.2, VIII.E.1.b. EX1003, ¶162.

A POSITA would have reasonably expected to succeed in incorporating this functionality from True in Entourage's search tool at least because Entourage's search tool already includes the same or substantially similar functionality, and because it would have required only simple and straightforward software modifications. *See, e.g., Keynetik*, WL 2003932 at \*2 (Fed. Cir. Feb. 15, 2023). EX1003, ¶163.

Specifically, True discloses that the “user types a query, consisting of a combination of words of interest” in a single search field (a “query window”), as shown in True’s annotated FIG. 2 below. EX1009, 3:36-37.

**FIG. 2**



EX1009, FIG. 2 (red annotation added). EX1003, ¶164.

The search tool separately searches each search string (e.g., “queen” and “mother”) and displays “query results ... as the query is being input” (EX1009, 2:41-42) according to the number of occurrences of the search strings in each file (transcript window) and the system’s confidence that the strings were correctly transcribed by the speech recognizer. EX1009, 4:21-5:15, 5:41-50 (“A window's interest level may be calculated, for example, as the sum of the confidences of each word in the transcript window matching a word in the query. For example, if the query is ‘queen mother,’ and a transcript window contains an occurrence of the word ‘queen’ with a confidence level of 0.27, another occurrence of the word ‘queen’ with a confidence level of 0.75, and an occurrence of the word ‘mother’ with a confidence level of 0.82, then the window's interest level is 1.84 (0.27+0.75+0.82). Thus, multiple occurrences of a search word are all given weight in the interest level computation.”), FIG. 2. EX1003, ¶165.

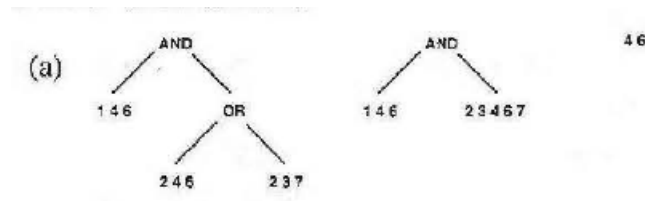
Thus, it would have been obvious to a POSITA to incrementally locate files containing two strings (as opposed to just one string) to provide the user with more relevant results, as is suggested by the Entourage references and was known to a POSITA. *Supra* Section IV.B.2; EX1057, 27. EX1003, ¶166.

**Identifying Files Containing Both Search Strings By Separately Searching Each String And Identifying Files Present In Both Sets Of Search Results**

When implementing True's teachings to search two search strings, a POSITA would have been motivated with a reasonable expectation of success to incorporate Baeza-Yates' teachings to locate files containing the two search strings by independently searching each string and then taking the intersection of those results (i.e., locating a first set of files that each includes at least one string that corresponds to the first partial search string, locating a second set of files that each includes at least one string that corresponds to the second partial search string, and identifying one or more files that are included in both the first and second sets of files) at least because this was one of only a limited number (a few at most) of known ways to identify files containing more than one search string, it was simple, predictable and easy to implement, and was especially well suited for Entourage's incremental search. EX1023, 101-103, 195-196, 207-208; *supra* Section IV.B.2-IV.B.3. EX1003, ¶167.

For example, Baeza-Yates describes how "phrases and proximity queries are split into single words. ... The lists of the occurrences of all the words found are retrieved. ... The occurrences are processed to solve ... Boolean operations."). EX1023, 195; *see also id.*, 101-103, 196 ("Each element must be searched separately and a list ... generated for each one. Then, the lists of all elements are traversed in

synchronization to find places where all the words ... appear close enough (for proximity).”), 207-208. Figure 8.12 of Baeza-Yates, for example, shows how a simple Boolean search for “[string 1] AND [string 2] OR [string 3]” identifies a first set of files (1, 4, and 6) containing the first search string, identifies a second set of files (2, 3, 4, 6, and 7) containing either the second string or the third string, and then identifies the files (4 and 6) that appear in both sets of files.



EX1023, 208. EX1003, ¶168.

Many others also provided similar descriptions of this classical approach. EX1019, 6:52-62 (“In this example, a user is searching for documents and presents a search request with a query string ‘The game of Go’. Search engine [] looks up each of the terms in word index. ... Search engine [] 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.”), 9:11-12, 9:24-32; EX1024, 17 (“If the user types multiple keywords in the query, the search engine searches each of the keywords independently and combine [sic] all the results to generate the final answer.”); EX1025, 109-110. EX1003, ¶169.

Thus, it would have been obvious to a POSITA to identify the files containing strings corresponding to both partial search strings by incrementally searching each search string separately and then identifying the files that appear in both sets of search results at least because this was one of only a limited number (a few at most) of well-known techniques that was predictable, simple, and easy to implement. Moreover, this intersection approach would have been a commonsense approach for locating files containing both search strings when performing an incremental search because a POSITA would have recognized that this approach's separate treatment of each string is well suited for an incremental search, as explained above in Section VII.E.1.c. EX1003, ¶¶170-172.

### **Using An Index For The Search**

As explained above for element [1.5], when performing the search, a POSITA would have understood or at least found it obvious that the computing system searches the index data structure, rather than every word of every file, because the POSITA knew that this approach is much faster and more efficient. *Supra* Sections IV.B.3, VIII.C.1.c. Moreover, it would be nonsensical to create the index data structure but not use it to expedite the search. EX1003, ¶173.

### **Including Files In Which The Search Strings Are Non-Adjacent**

Finally, a POSITA would have understood or at least found it obvious that the partial search strings are non-adjacent in at least one of the matching files at least

because this is the entire point of treating the multi-word query as two separate search strings that are searched independently. EX1003, ¶174.

Moreover, it would have been obvious to include files in which the search strings are non-adjacent at least because it was common/well known to do so and was an inherent part of Boolean AND queries. *Supra* Section IV.B.2; EX1017, 1:66-2:2; EX1023, 24-27, 101-103, 192-196; EX1059, 160. EX1003, ¶175.

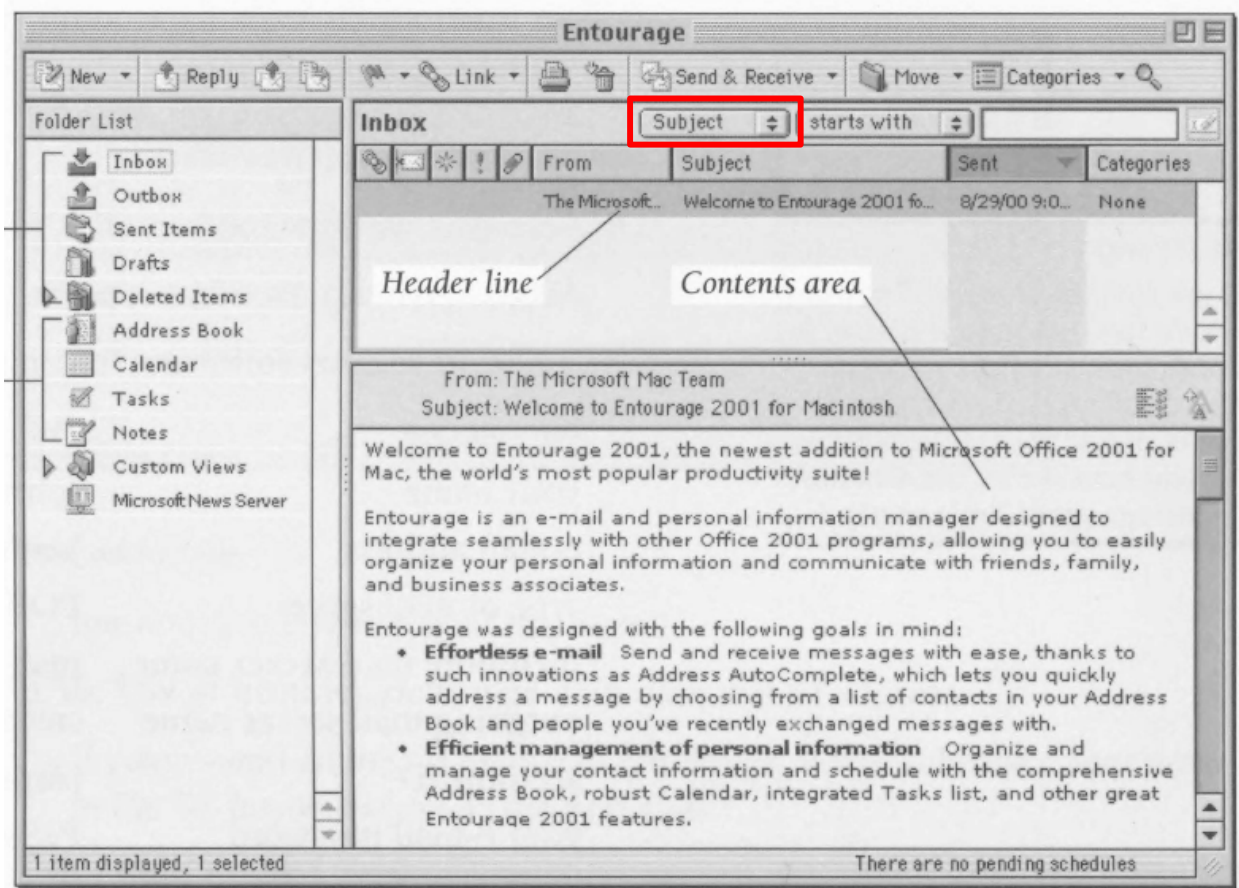
## **2. Claims 2-3**

The Entourage+True+Baeza-Yates combination renders these claims obvious at least because a POSITA would have understood that the Entourage references' disclosed search is order-agnostic. Specifically, as explained above for elements [1.11-1.13], a POSITA would have understood or at least found it obvious that the search strings are searched independently as separate strings (not as an exact phrase) at least because it was well known that this is how search tools performed multi-string searches. *Supra* Sections IV.B.2, VII.C.1.g. EX1003, ¶176.

Indeed, a POSITA knew that search tools typically would care about the order of the search strings only when a user expressly indicated that the order mattered, such as when the user indicated that they were searching for a specific phrase (e.g., by including quotation marks around the phrase). EX1026, 40 ("Use quotes for phrases... ."); EX1057, 19-21, 60-61, 89, 129, 146, 151, 163 ("Phrases should be enclosed in double quotes."), 166. EX1003, ¶¶177-178.

### 3. Claim 4

The Entourage+True+Baeza-Yates combination renders this claim obvious at least because the Entourage references disclose searching within specific attribute fields of the files, such as the To, From, and Subject fields of email files. EX1004, 2-3, 8-14, 38, 89, 161 (“Click the leftmost pop-up menu above the message list, and choose Subject, From, To, or Category. ... The message list is filtered to show only the messages that have been assigned the chosen category. ... Type a search string in the text box. Matching headers are displayed in the message list.”), 162, 205, 207-209; EX1005, 173; EX1006, 384-387; *see also infra* Section VII.E.10 (claim 14).



EX1005, 173 (red annotation added showing searching only within the “Subject” header).

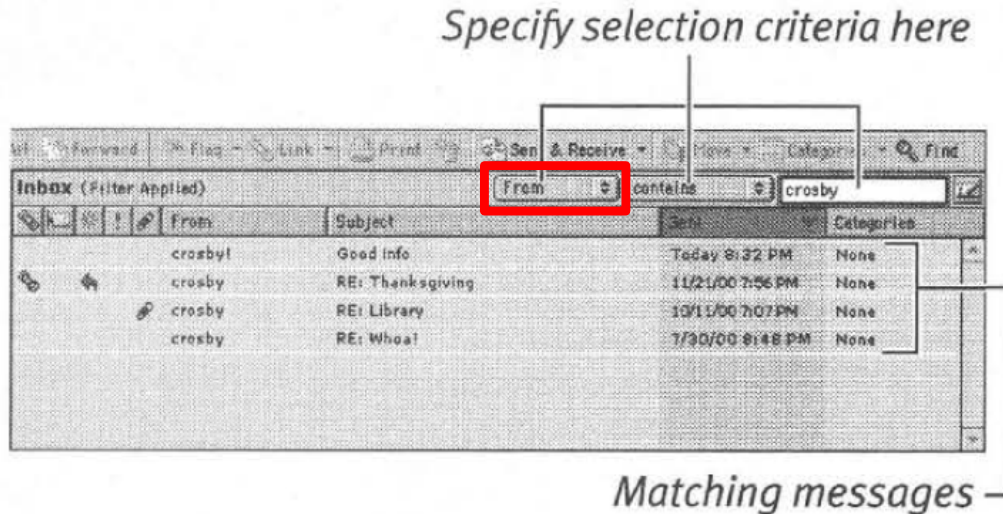
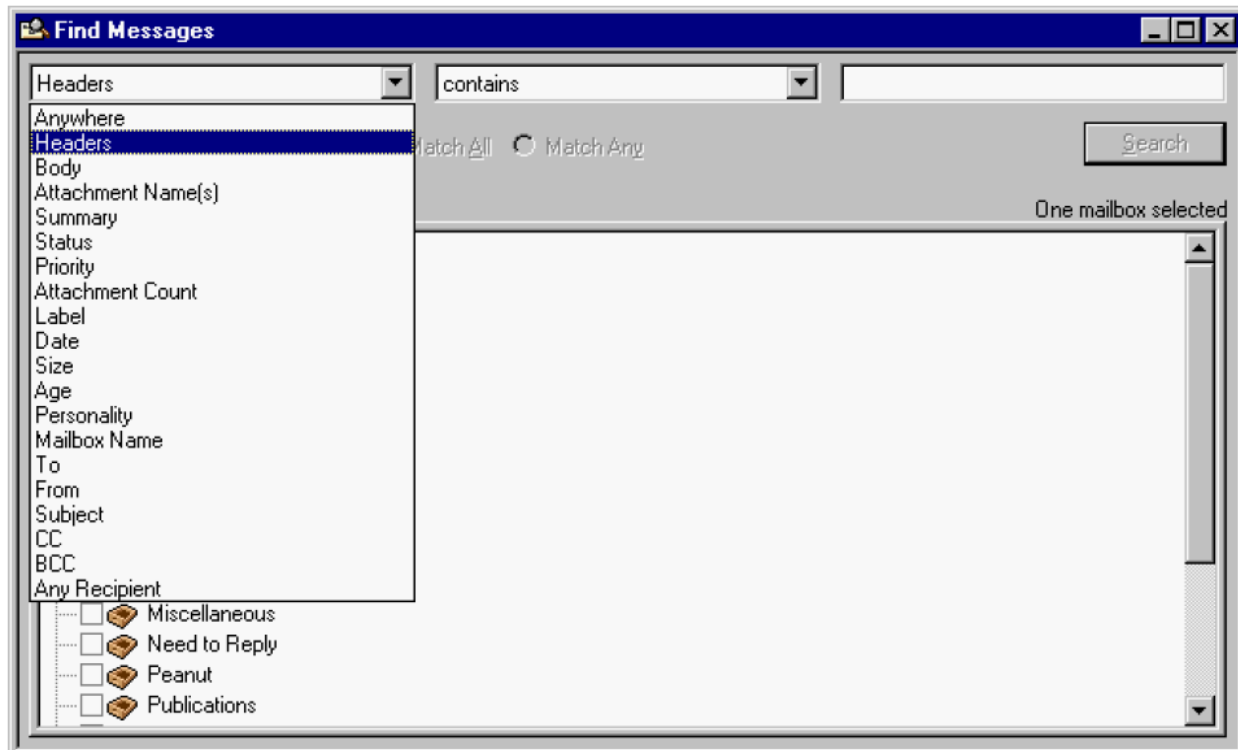


**Figure 11.15** To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161 (red annotations added showing searching only within the “From” header). EX1003, ¶179.

Moreover, it was well known to perform this kind of attribute field-specific search. *Supra* Section IV.B.5.

*Find Messages drop-down and conjunction options lists*



EX1007, 129, EX1003, ¶180.

To perform this kind of attribute field-specific searching, a POSITA would have understood or at least found it obvious that the index data structure would need to include indications of character strings associated with respective attribute fields of the files. EX1003, ¶181.

**4. Claims 5-7**

As explained above for claim 1, the Entourage+True+Baeza-Yates combination renders obvious generating an email index and including an email search mode interface having email-specific attribute search fields including at least a date field, a from field, and a sender field. *Supra* Section VIII.E.1.c. Although the

Entourage references do not expressly disclose the same kind of search capability for documents and email attachments, it would have been obvious to a POSITA to do so, especially since the Entourage references disclose searching for documents (EX1006, 16-18) and searching for emails based on whether or not they include an attachment (EX1004, 162; EX1006, 386-387). EX1003, ¶182.

A POSITA would have been motivated to extend Entourage's incremental email search capabilities to documents and email attachments so that a user could also search for and more quickly locate these other types of files. EX1003, ¶183.

A POSITA would have reasonably expected to succeed in modifying the existing Entourage programming and user interface techniques to also search documents and email attachments because it would have required only trivial software modifications that were well within the skill of a POSITA, as evidenced by the fact that Entourage's own Advanced Find tool allows users to search for multiple types of files (e.g., emails, contacts, events, tasks, and notes) simultaneously. EX1004, 11. Indeed, well before the priority date of the '490 patent, searching through any type of file—e.g., databases, text documents, web pages, etc.—was so well known that it was literally textbook knowledge. EX1021, 124-132; EX1023, 108, 141, 142 (a retrieved document includes “a file, an email, [and] a World Wide Web (or just Web) page”), 144 (“There is no single format for a text document, and

an IR system should be able to retrieve information from many of them.”), 367; EX1024, 1-5; EX1059, 159-171; EX1061, 133-148. EX1003, ¶¶184-185.

## 5. Claim 8

As explained above for claim 1, the Entourage references disclose that the partial search string comprises characters corresponding to at least a portion of a first string, such as “schwa” for “schwartz.” *Supra* Section VIII.E.1.f.

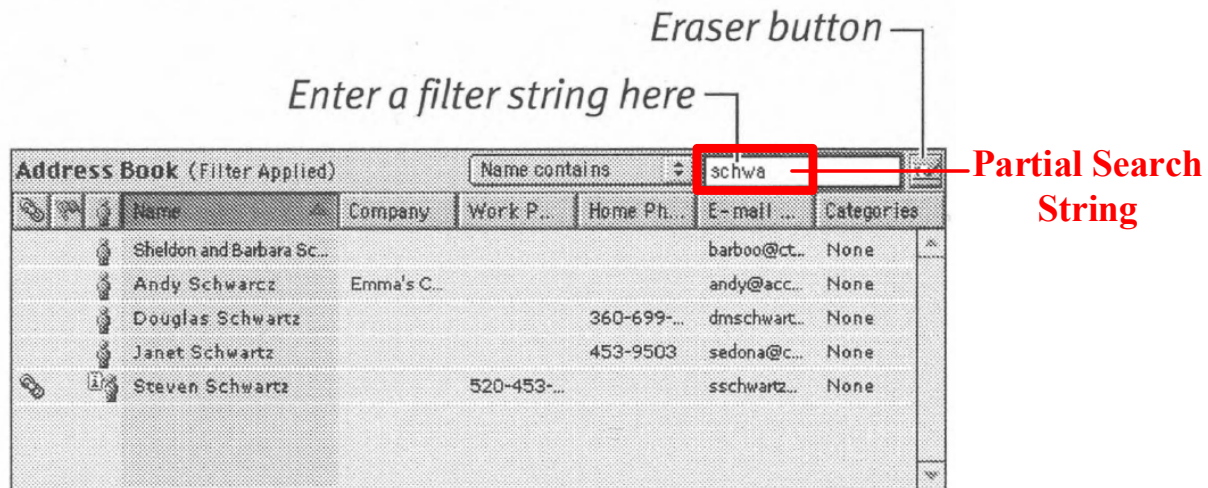


**Figure 3.4** You can limit the display to records that contain a particular text string.

EX1004, 38 (red annotations added). *See also id.*, 8, 9 (results are shown “[a]s you type the filter text”), 10, 38-39, 89, 161, 204-205, 209; EX1005, 173; EX1006, 358, 384; *supra* Section IV.B.1; EX1043, 83-84, 337-338, 343-344, 355, 385-386, 404. Thus, this claim is obvious for at least the reasons claim 1 is obvious. EX1003, ¶186.

6. Claim 9

As explained above for claim 1, the Entourage references disclose that the computing system displays the set of matching files in the interactive user interface.

*Supra* Sections VIII.C.1.e, VIII.C.1.g.

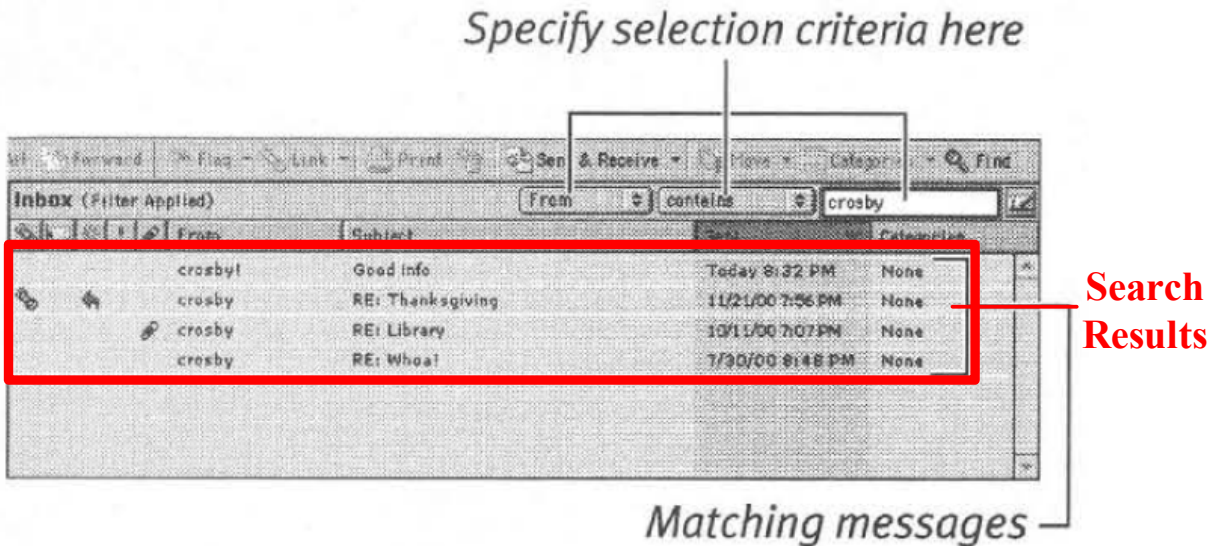


**Figure 11.15** To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161 (red annotations added).

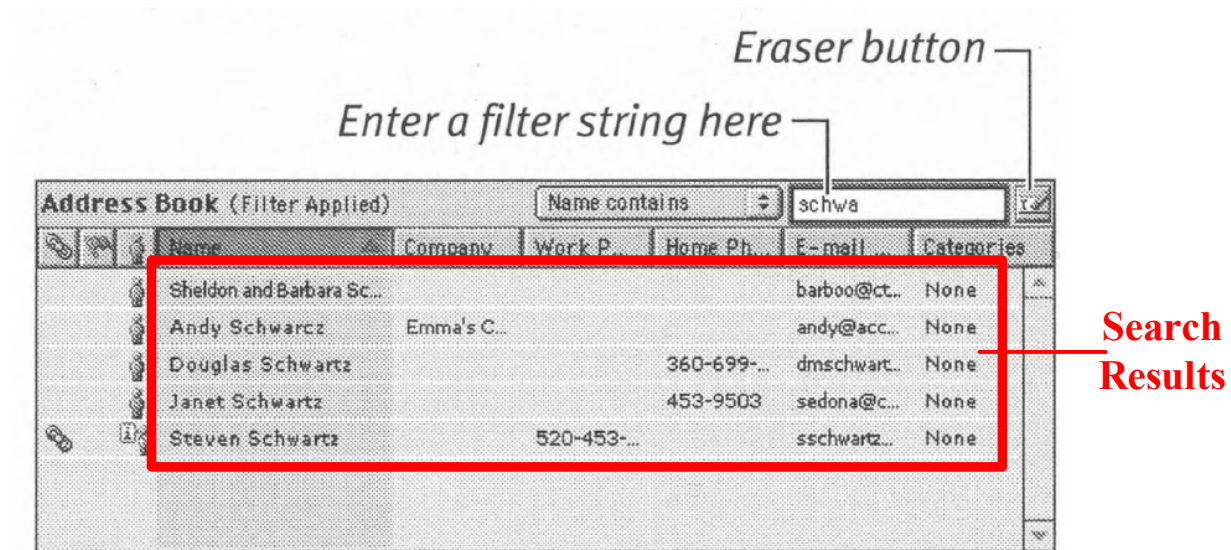


**Figure 3.4** You can limit the display to records that contain a particular text string.

EX1004, 38 (red annotations added). Thus, this claim is obvious for at least the reasons claim 1 is obvious. EX1003, ¶187.

## 7. Claims 10-11

The Entourage+True+Baeza-Yates combination renders these claims obvious at least because the Entourage references disclose displaying a portion of a selected file and highlighting occurrences of only the searched string. EX1004, 2-3, 6-13, 38-39, 89, 161-163, 204-205; EX1006, 384-387. EX1003, ¶188.

For example, a user can select a particular file (e.g., email, contact) and a portion of the file is displayed in the preview pane. EX1004, 2-3, 6-7, 38, 162 (“To examine any of the matches, click its header and read the message in the preview pane.”), 205.

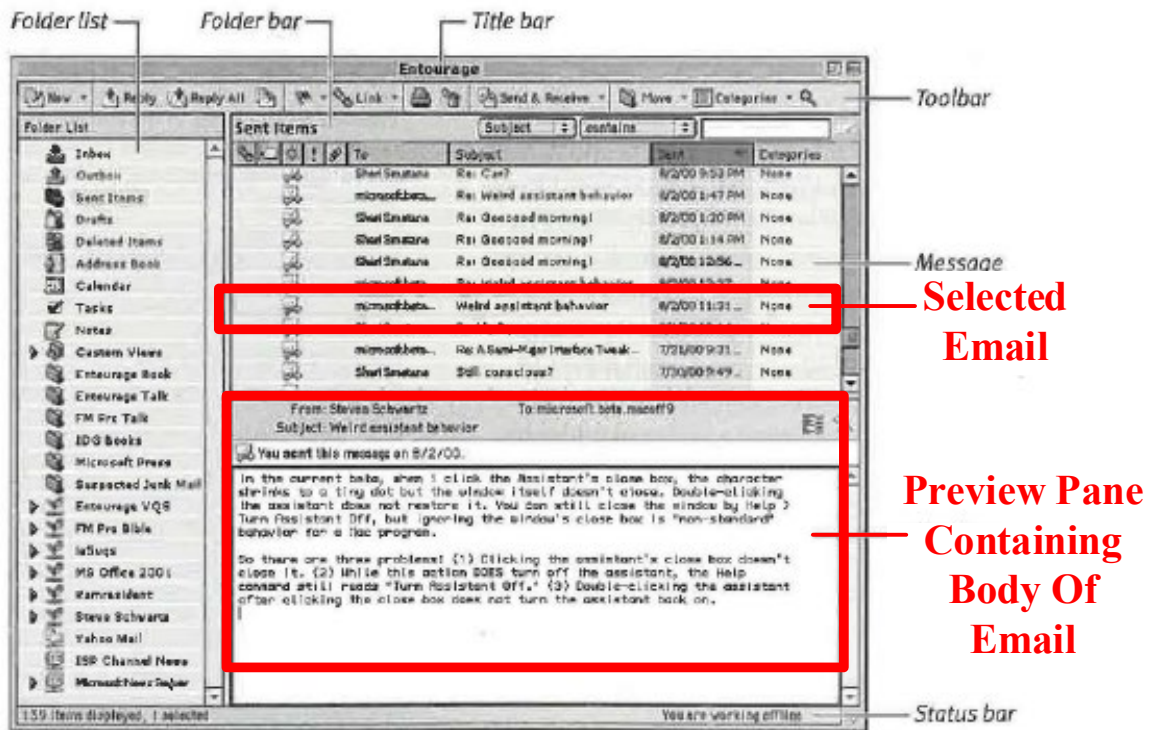
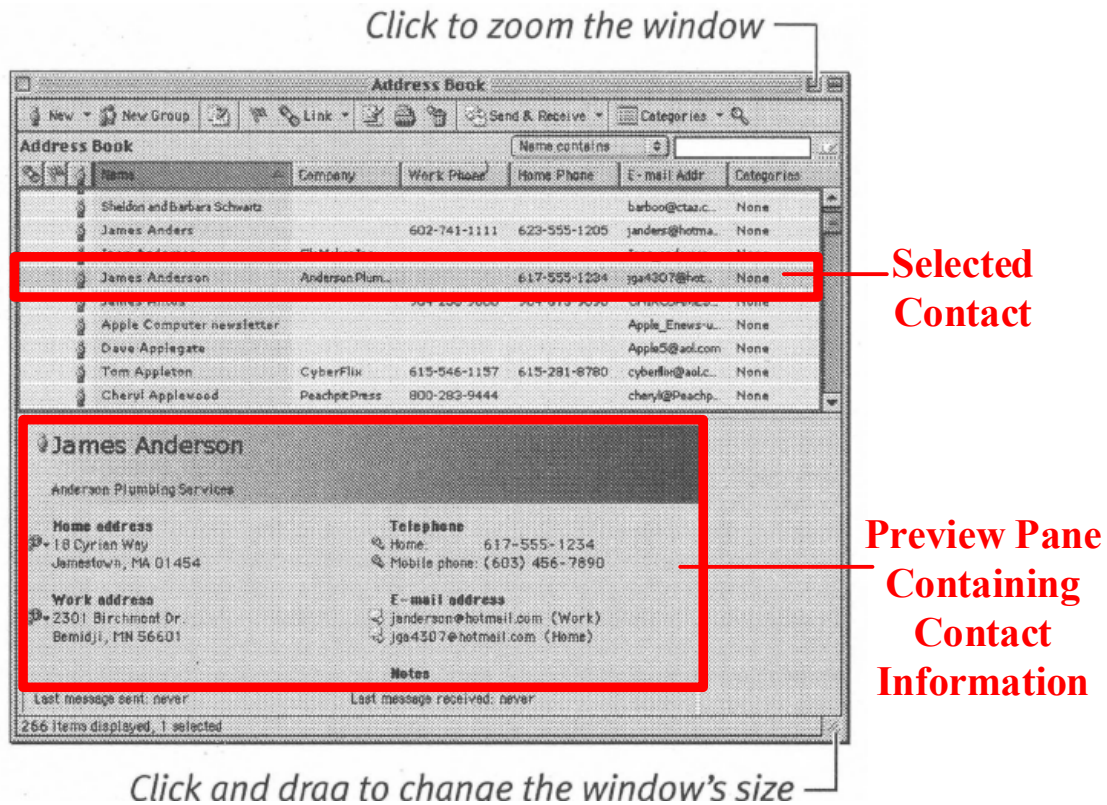


Figure 1.4 The main Entourage window.

EX1004, 3 (red annotations added).



EX1004, 38 (red annotations added). EX1003, ¶¶189-190.

The Entourage references also disclose highlighting a search term within a given file. EX1004, 11, 163; *see also* EX1043, 386.

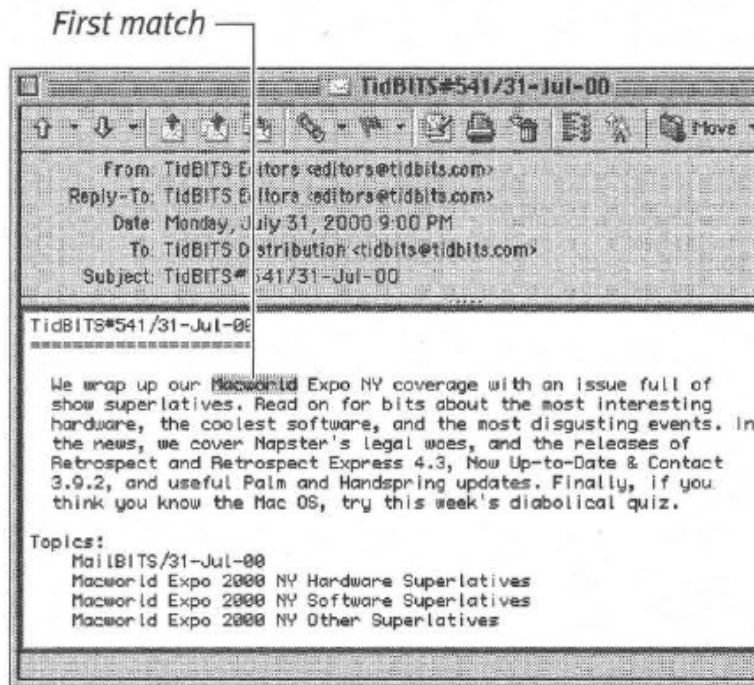


**Figure 1.16** When you search for matching text within an item, Entourage highlights the first match, scrolling as necessary to display it.

EX1004, 11.

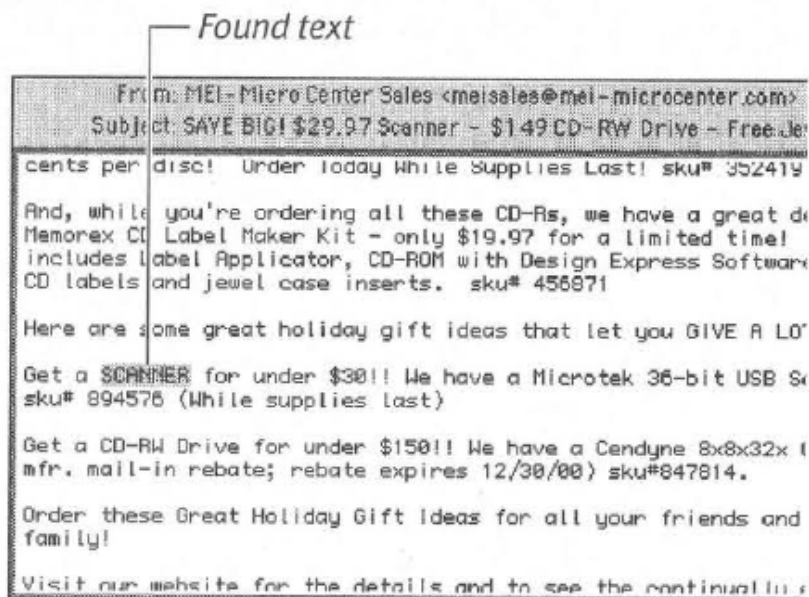


**Figure 11.20** The first instance of the found text—if there is one—is highlighted for you.

EX1004, 163. EX1003, ¶191.

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 file, as was well known in the art. *See supra* Section IV.B.4. EX1003, ¶¶192-194.

**8. Claim 12**

The Entourage+True+Baeza-Yates combination renders this claim obvious at least because, as explained above for claim element [1.10], the Entourage references disclose that the string separator character is a space. *Supra* Section VIII.E.1.f. Moreover, it was well known to use a space or a plus sign (+) as a string separator character, including to serve as a Boolean AND operator. *Id.* EX1003, ¶195.

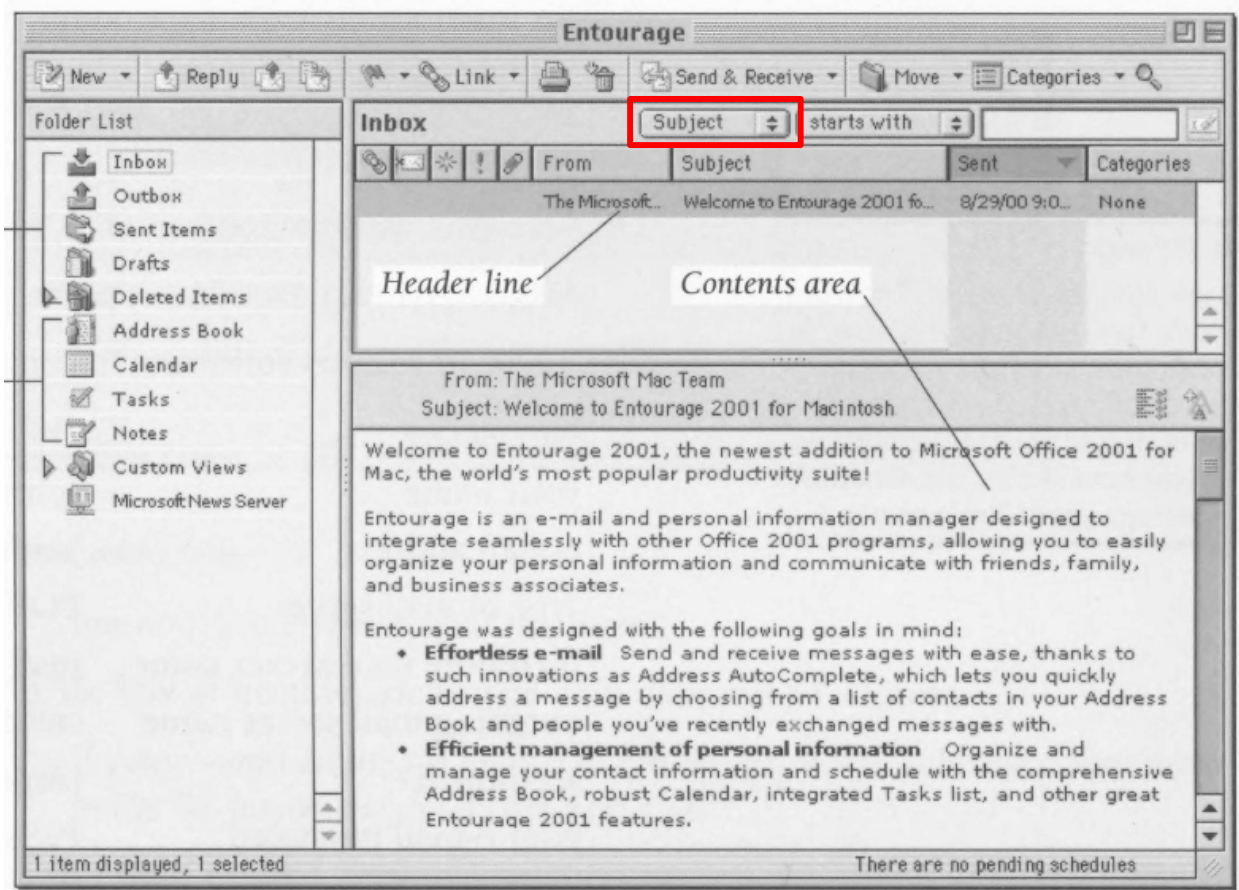
**9. Claim 13**

The Entourage+True+Baeza-Yates combination renders this claim obvious at least because, as explained above for elements [1.11-1.13], the Entourage references disclose that the files comprise emails, contact information, task information, and text documents (news articles). *Supra* Section VIII.E.1.g. EX1003, ¶196.

**10. Claims 14-15**

The Entourage+True+Baeza-Yates combination renders these claims obvious at least because, as explained above for claim 1, the Entourage references teach that the single search field corresponds with a single field of the files and only that field

of the files is incrementally searched. *Supra* Section VIII.E.1. Specifically, Entourage includes a drop-down menu where a user can search within a specific field (e.g., “To,” “From,” “Subject,” etc.) and the incremental search is performed only within the specified field. *Supra* Section VIII.A; EX1004, 2-3, 8-14, 38, 89, 161, 162, 205, 207-209; EX1005, 173; EX1006, 384-387.



EX1005, 173 (red annotation added showing searching only within the “Subject” header).

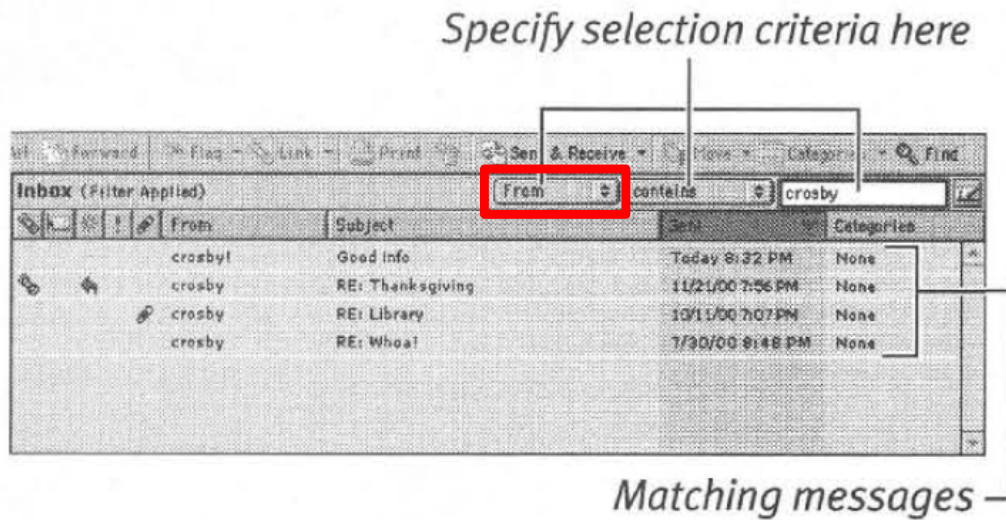
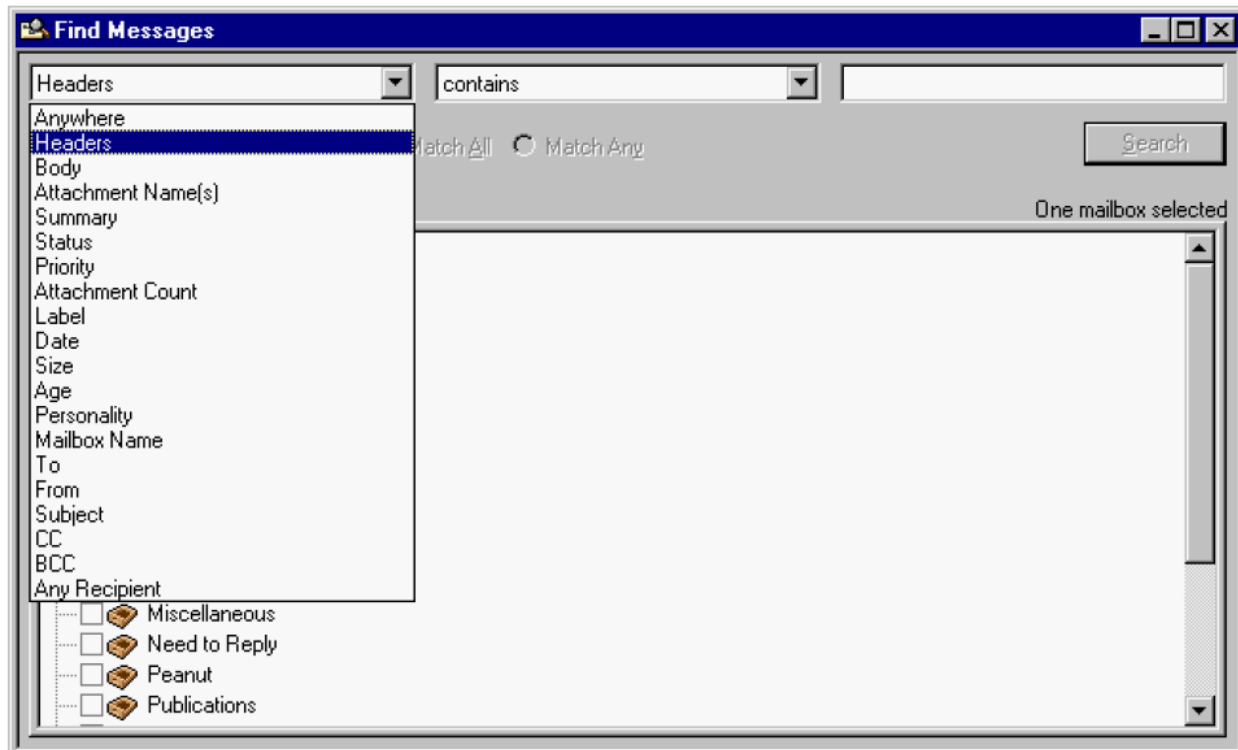


**Figure 11.15** To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161 (red annotations added showing searching only within the “From” header). Moreover, it was well known in the art to search within these specific email headers, among others. *Supra* Section IV.B.5.

*Find Messages drop-down and conjunction options lists*



EX1007, 129. EX1003, ¶197.

**11. Claim 16**

The Entourage+True+Baeza-Yates combination renders this claim obvious at least because the Entourage references disclose “search[ing] for a document based on its name, file type, and location.” EX1006, 18; *see also id.*, 17. Although the Entourage references do not expressly disclose that this document-specific search is incremental, a POSITA would have found it obvious to implement it 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.g. Moreover, as

explained for claims 5-7, modifying existing Entourage searching of email messages on email-specific attributes to enable searching of documents on document-specific attributes was trivial and well within the ability of a POSITA and would have been obvious to try since searching documents based on various attributes of documents was commonplace and well known prior to the priority date of the '490 patent. *Supra* Section VIII.E.4. EX1003, ¶198.

## 12. Claim 17

The Entourage+True+Baeza-Yates combination renders this claim obvious at least because the Entourage references disclose that the single field comprises one of a “contact name, company, or assigned category” of contacts files in an address book. EX1004, 38.

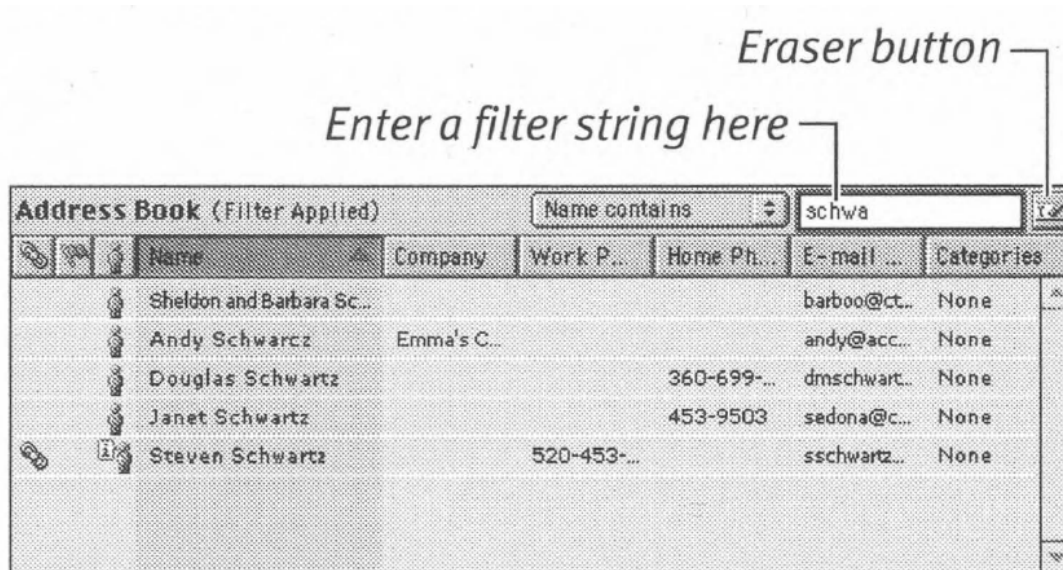


**Figure 3.4** You can limit the display to records that contain a particular text string.

EX1004, 38. EX1003, ¶199.

**13. Claim 18**

The Entourage+True+Baeza-Yates combination renders this claim obvious at least because the Entourage references disclose that the single field comprises the subject and/or content fields of tasks (EX1004, 10, 86, 89), which the Entourage references disclose is a type of calendar file (EX1004, 66, 73) and which includes typical calendar file characteristics, including a specific date, time, and reminders (EX1004, 83-89). Moreover, the Entourage references expressly disclose the ability to search calendar files (EX1004, 11), albeit in the non-incremental advanced find dialog box, but it would have been obvious to a POSITA to implement this 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.g. Moreover, as explained for claims 5-7, modifying existing Entourage searching of email messages on email-specific attributes to enable searching of other types of files on attributes specific to those files was trivial and well within the ability of a POSITA and would have been obvious to try since searching any type of file based on attributes appropriate for a given file type was commonplace and well known prior to the priority date of the '490 patent. *Supra* Section VIII.E.4. EX1003, ¶200.

**14. Claim 19**

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. Thus, the Entourage+True+Baeza-Yates combination renders this claim obvious. EX1003, ¶201.

**15. Claim 20**

The Entourage+True+Baeza-Yates combination renders this claim obvious at least because Baeza-Yates discloses including Boolean operators in a single search field (e.g., EX1023, 102-103) and a POSITA would have been motivated with a reasonable expectation of success to allow the user to apply at least the AND

operator to the search strings to provide more precise search results, as was well known in the art. *Supra* Sections IV.B.2, VIII.C.1.g. EX1003, ¶202.

#### **IX. NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS**

While Patent Owner has argued secondary considerations in the ITC proceeding, they are based on substantively different claims in different patents, and even if given any weight, such considerations cannot overcome the Petition's strong showing of obviousness. EX1003, ¶203.

Patent Owner cannot show a nexus between the claims and the alleged objective indicia. Patent Owner cannot show a presumption of nexus because the product that allegedly embodies the claim is not coextensive with the claims at least because it includes an unclaimed feature—the ability to enter a search command in the search field—that is recited by the patents involved in the ITC proceeding and materially impacts the product's functionality. *Fox Factory, Inc. v. SRAM, LLC*, 944 F.3d 1366, 1375 (Fed. Cir. 2019) (“A patent claim is not coextensive with a product that includes a ‘critical’ unclaimed feature that is claimed by a different patent and that materially impacts the product's functionality ...”).

Patent Owner cannot otherwise demonstrate nexus at least because Patent Owner appears to tie any secondary considerations to either unclaimed features (e.g., specific indexing approaches that allow for faster search results) or claimed features (e.g., incremental searching and highlighting of search terms) that were known in

the prior art (*supra* Sections IV.B.1, IV.B.4, and VII-VIII). *Novartis AG v. Torrent Pharms. Ltd.*, 853 F.3d 1316, 1331 (Fed. Cir. 2017) (“[T]he identified objective indicia must be directed to what was not known in the prior art.”). EX1003, ¶203. Indeed, one of Patent Owner’s own experts admitted in the ITC proceeding that incremental searching was well known. EX1079 at 1007:20-22. While Patent Owner has provided some generic praise of X1’s products, it has not demonstrated that the alleged objective indicia are the direct result of novel claimed features of the challenged patent

Further, the supposed evidence submitted at the ITC (*see, e.g.*, EX1063-EX1074) cuts against any secondary considerations supporting non-obviousness, as it shows that Patent Owner’s product “isn’t that much different than others” (EX1072, 1-2) and is inferior to other products because it did not allow for Boolean and phrase searches (EX1063, 1).

## **X. CONCLUSION**

Petitioner requests institution and cancellation of claims 1-20 of the ’490 patent.

Respectfully submitted,

Dated: December 20, 2024

By: /Todd M. Siegel/  
Todd M. Siegel, Reg. No. 73,232  
todd.siegel@klarquist.com  
KLARQUIST SPARKMAN, LLP  
One World Trade Center, Suite 1600

121 S.W. Salmon Street  
Portland, Oregon 97204  
Tel: 503-595-5300  
Fax: 503-595-5301

Counsel for Petitioner

**CERTIFICATE OF COMPLIANCE WITH  
TYPE-VOLUME LIMITATION PURSUANT TO 37 C.F.R. § 42.24**

This brief complies with the type-volume limitation of 37 C.F.R. § 42.24(a)(1)(i).

The brief contains 13,957 words, excluding the parts of the brief exempted by 37 C.F.R. § 42.24(a).

The brief has been prepared in a proportionally spaced typeface using Microsoft Word for O365 in a 14-point Times New Roman font.

Dated: December 20, 2024

By: /Todd M. Siegel/  
Todd M. Siegel, Reg. No. 73,232  
todd.siegel@klarquist.com  
KLARQUIST SPARKMAN, LLP  
One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Tel: 503-595-5300  
Fax: 503-595-5301

Counsel for Petitioner

**CERTIFICATE OF SERVICE**  
**IN COMPLIANCE WITH 37 C.F.R. § 42.6(e)(4)**

The undersigned certifies that the **Petition for *Inter Partes* Review of U.S. Patent No. 10,552,490 and Exhibits 1001-1044, 1052-1076, and 1079** were served on December 20, 2024, via **Express Mail** on the Patent Owner at the following address of record as listed on the USPTO Patent Center:

Knobbe, Martens, Olson & Bear, LLP  
2040 Main Street, Fourteenth Floor  
Irvine, CA 92614

The undersigned certifies that the Petition only was served on counsel of record in ITC 337-TA-1389 and District Court case no. 8:23-cv-02415 as follows:

Patrick J. McCarthy - PMcCarthy@goodwinlaw.com  
Christie L. Larochelle - CLarochelle@goodwinlaw.com  
Amadou Kilkenney Diaw - ADiaw@goodwinlaw.com  
I. Neel Chatterjee - NChatterjee@goodwinlaw.com

By: /Todd M. Siegel/  
Todd M. Siegel, Reg. No. 73,232  
todd.siegel@klarquist.com  
KLARQUIST SPARKMAN, LLP  
One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Tel: 503-595-5300  
Fax: 503-595-5301

Counsel for Petitioner