UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

MICROSOFT CORPORATION,
Petitioner,

v.

X1 Discovery, Inc.,
Patent Owner.

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

DECLARATION OF LOREN G. TERVEEN, PH.D.

MICROSOFT
EXHIBIT 1003

TABLE OF CONTENTS

Page

I. INTRODUCTION AND ENGAGEMENT 6

II. BACKGROUND AND QUALIFICATIONS.....6

III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '490 PATENT.....9

IV. UNDERSTANDING OF PATENT LAW 14

V. THE '490 PATENT.....18

VI. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART.....21

A. Person Of Ordinary Skill In The Art.....21

B. State Of The Art 22

1. Incrementally (i.e., Character-By-Character) Locating Documents Containing Partial Strings (e.g., Prefixes).....22

2. Locating Documents Containing Two Non-Adjacent Search Strings Entered In The Same Search Field (e.g., Boolean AND Searching).....26

3. Using Indexes To Expedite The Searching Process 31

4. Displaying Portion(s) Of A Document Containing The Search Term(s) At Non-Adjacent Locations And Highlighting The Non-Adjacent Search Terms.....34

5. Email-Specific Searching.....42

VII. CLAIM CONSTRUCTION 43

VIII. GROUND 1: CLAIMS 1-20 ARE OBVIOUS OVER LOTUS NOTES (EX1059, EX1061), RASKIN (EX1021), AND WU (EX1019) 43

A. Overview Of The Lotus Notes References (EX1059, EX1061).....44

B. Overview Of Raskin (EX1021) 48

C. Overview Of Wu (EX1019) 50

D. Combining The Lotus Notes References, Raskin, And Wu 51

E. The Lotus Notes+Raskin+Wu Combination Teaches Every Claim Element 54

1. Claim 1 54

a) Element [1.1] 54

b) Elements [1.2-1.4] 54

c) Element [1.5] 55

d) Element [1.6] 56

e) Element [1.7] 56

f) Elements [1.8-1.10] 57

g) Elements [1.11-1.13] 60

2. Claims 2-3 65

3. Claim 4 66

4. Claims 5-7 68

5. Claim 8 70

6. Claim 9 70

7. Claims 10-11 71

8. Claim 12 72

9. Claim 13 73

10. Claims 14-15 73

11. Claim 16 76

12. Claim 17 77

13. Claim 18 77

14. Claim 19 78

15. Claim 20 79

IX. GROUND 2: CLAIMS 1-20 ARE OBVIOUS OVER THE ENTOURAGE REFERENCES (EX1004-EX1006), TRUE (EX1009), AND BAEZA-YATES (EX1023) .. 80

A. Overview Of The Entourage References (EX1004-EX1006) 81

B. Combining The Entourage References 87

C. Overview Of The True (EX1009) And Baeza-Yates (EX1023) References 88

D. Combining Entourage With The True And Baeza-Yates References 91

E. The Entourage+True+Baeza-Yates Combination Teaches Every Claim Element 93

1. Claim 1 93

a) Element [1.1] 93

b) Elements [1.2-1.4] 93

c) Element [1.5] 94

d) Element [1.6] 97

e) Element [1.7] 98

f) Elements [1.8-1.10] 99

g) Elements [1.11-1.13] 105

2. Claims 2-3 118

3. Claim 4 119

4. Claims 5-7 122

5. Claim 8 125

6. Claim 9 126

7. Claims 10-11 128

8. Claim 12 138

9. Claim 13 138

10. Claims 14-15 139

11. Claim 16 142

12. Claim 17 143

13. Claim 18 144

14. Claim 19 145

15. Claim 20 146

X. NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS 147

I, Loren G. Terveen, Ph.D., do hereby declare as follows:

I. INTRODUCTION AND ENGAGEMENT

1. I have been retained as an independent expert on behalf of Microsoft Corporation in connection with the above-captioned Petition for *Inter Partes* Review ("IPR") to provide my analyses and opinions on certain technical issues related to U.S. Patent No. 10,552,490 (hereinafter "the '490 patent").

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

3. Specifically, I have been asked to provide my opinions regarding whether claims 1-20 (each a "challenged claim" and collectively the "challenged claims") of the '490 patent would have been obvious to a person having ordinary skill in the art ("POSITA") as of September 3, 2002. It is my opinion that each challenged claim would have been obvious to a POSITA after reviewing the prior art discussed herein.

II. BACKGROUND AND QUALIFICATIONS

4. I am a Professor of Computer Science & Engineering at the University of Minnesota.

5. I received a B.A. in Computer Science, Mathematics, and History from the University of South Dakota in 1984, a M.S. in Computer Science from the

University of Texas in 1988, and a Ph.D. from the University of Texas in Computer Science in 1991.

6. I am a member of the Association for Computing Machinery (ACM), the oldest, largest, and most prestigious computing society in the world. From 2015-2018 I served as President of ACM's Special Interest Group on Computer-Human Interaction, one of its largest and most active special interest groups. I also am a member of the ACM Council, the highest governing body of the ACM. I received the ACM Distinguished Scientist Award in 2009.

7. My research and teaching focus on human-computer interaction, user interface design, and social computing. I have several decades of experience in these specialties of computer science in both industry and academia. I worked for AT&T Laboratories from 1991 through 2002, during which time I conducted research and developed systems that solved problems in software engineering, web information seeking and organization, and recommender systems. As part of this research, I developed a number of systems on the World Wide Web, which involved, among other things, storing information in databases, generating HTML pages in response to user requests, and providing interactive search and browsing interfaces to users through the Common Gateway Interface (CGI) and other technologies. In all my research, I designed, implemented, and tested graphical user interfaces. I have been employed full-time as a professor at the University of Minnesota since 2002, during

which time I have taught classes in computer science, human-computer interaction and social computing, and have conducted, supervised, and published research in the field, much of which has involved information organization and seeking systems. My research has been published in numerous journal and conference papers, as well as in a book I co-authored entitled "Foundational Issues in Artificial Intelligence and Cognitive Science: Impasse and Solution." I have served on the editorial board of ACM's Transactions on Human-Computer Interaction and the Communications of the ACM, and have served as a reviewer for numerous journals, including ACM Computing Surveys, IEEE Transactions on Data and Knowledge Engineering, the International Journal of Human-Computer Studies, and the Journal of Computer-Supported Cooperative Work.

8. I am a listed inventor on ten patents, including several related to graphical interfaces, web-based systems, and information organization and search.

9. I have consulted on over 15 intellectual property cases that have dealt with topics in user interface design, recommender systems, web information systems, and information seeking systems. During these cases, I have testified before judges, been deposed, and written multiple expert reports.

10. I have attached a more detailed list of my qualifications as Appendix A. Based on my academic and professional experiences, I believe that I am

qualified in the technology fields pertaining to this proceeding (and described below), at least as a person of ordinary skill in the art.

11. I have no financial interest in the outcome of this proceeding, and will be paid for my time regardless of whether I testify at the proceedings or whether the outcome is in accordance with my conclusions.

III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '490 PATENT

12. In preparing this declaration, I have reviewed the following materials bearing Exhibit Nos. that I understand are being referenced in the IPR to which my declaration accompanies:

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 10,552,490 ("**'490 patent**")
1002	File History of U.S. Patent No. 10,552,490
1004	Steve Schwartz, *Visual QuickStart Guide Entourage 2001 for Macintosh*, Peachpit Press (2001) ("**Entourage-1**")
1005	Bonita Sebastian et al., *Microsoft Office 2001 A Hands-On Guide for Macintosh*, Computer Literacy Press (excerpts) (2000) ("**Entourage-2**")
1006	Nan Barber & David Reynolds, *Office 2001 for Macintosh: The Missing Manual*, Pogue Press/O'Reilly (1st ed.) (Apr. 2001) ("**Entourage-3**")
1007	*Eudora Email User Manual for Windows Version 5.0*, QUALCOMM (Aug. 2000) ("**Eudora**")
1008	Alan R. Neibauer, *Running Microsoft Outlook 2000*, Microsoft Press (excerpts) (1999) ("**Outlook3**)

No.	Description
1009	U.S. Patent No. 6,112,172 to True et al. ("**True**")
1010	U.S. Patent No. 5,444,823 to Trung D. Nguyen ("**Nguyen**")
1011	U.S. Patent No. 7,039,635 to Morgan et al. ("**Morgan**")
1012	U.S. Patent Application Publication No. 2002/0055981 to Spaey et al. ("**Spaey**")
1013	U.S. Patent No. 5,845,278 to Kirsch et al. ("**Kirsch**")
1014	U.S. Patent No. 4,255,796 to Gabbe et al. ("**Gabbe**")
1015	U.S. Patent No. 5,692,173 to Chee Heng Chew ("**Chew**")
1016	U.S. Patent No. 6,049,796 to Siitonen et al. ("**Siitonen**")
1017	U.S. Patent No. 5,303,361 to Colwell et al. ("**Colwell**")
1018	U.S. Patent No. 6,741,985 to Robin A. R. Green ("**Green**")
1019	U.S. Patent No. 5,991,756 to Jiong Wu ("**Wu**")
1020	International Patent Application Publication No. WO 00/26762 to Mark Richanbach ("**Richanbach**")
1021	Jef Raskin, *The Humane Interface: New Direction for Designing Interactive Systems*, ACM Press (2000) ("**Raskin**")
1022	Richard M. Stallman, *GNU Emacs Manual* (1st ed.) (Jun. 1985) ("**GNU Emacs**")
1023	Ricardo Baeza-Yates & Berthier Ribeiro-Neto, *Modern Information Retrieval*, ACM Press (1999) ("**Baeza-Yates**")
1024	Sunny Lam, *The Overview of Web Search Engines* (Feb. 9, 2001) ("**Lam**")
1025	Oscar Vallarino, *On The Use Of Bit Maps For Multiple Key Retrieval*, Proceedings of the 1976 Conference on Data: Abstraction, Definition And Structure, 108-114 (Mar. 1976) ("**Vallarino**")

No.	Description
1026	Steve Lawrence & C. Lee Giles, *Context And Page Analysis For Improved Web Search*, IEEE Internet Computing, 38-46 (Jul.-Aug. 1998) ("**Lawrence**")
1027	Rawley D. Johnson et al., *Technical Reference Document Database System (TDOCS) Requirements Definition* (Aug. 1993) ("**Johnson**")
1028	Lev Finkelstein et al., *Placing Search in Context: The Concept Revisited*, ACM Transactions on Info. Sys., Vol. 20, No. 1, 116-131 (Jan. 2002) ("**Finkelstein**")
1029	U.S. Patent No. 6,952,805 to Tafoya et al. ("**Tafoya**")
1030	U.S. Patent No. 6,009,459 to Belfiore et al. ("**Belfiore**")
1031	U.S. Patent No. 5,220,625 to Hatakeyama et al. ("**Hatakeyama**")
1032	U.S. Patent No. 6,185,551 to Birrell et al. ("**Birrell**")
1033	U.S. Patent No. 6,839,702 to Patel et al. ("**Patel**")
1034	U.S. Patent No. 6,973,450 to Silverbrook et al. ("**Silverbrook**")
1035	U.S. Patent No. 6,834,276 to Jensen et al. ("**Jensen**")
1036	U.S. Patent No. 5,946,678 to Ijsbrand Jan Aalbersberg ("**Aalbersberg**")
1037	European Patent Application Publication No. EP 1 209 866 to Baldonado et al. ("**Baldonado**")
1038	Declaration of Delainey Tanner, dated November 18, 2024
1039	Kaitlin Duck Sherwood, *Overcome Email Overload With Eudora 5*, World Wide Webfoot Press (excerpts) (1st ed.) (2001)
1040	Claims Listing
1041	Donald E. Knuth, *The Art Of Computer Programming Vol. 3 Sorting And Searching*, Addison Wesley Longman (excerpts) (2nd ed.) (May 1998)
1042	N/A

No.	Description
1043	Steve Schwartz & Robert Correll, *Running Microsoft Office 2001 for Mac*, Microsoft Press (excerpts) (2001)
1044	Ben Shneiderman et al., *Clarifying Search: A User-Interface Framework For Text Searches*, 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
1052	Ben Shneiderman, *Designing The User Interface: Strategies For Effective Human-Computer Interaction*, Addison Wesley Longman (excerpts) (3rd ed.) (Mar. 1998)
1053	Claudine Badue et al., *Distributed Query Processing Using Partitioned Inverted Files*, 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. ("**Peltonen**")
1055	Dick Pountain, *The New Penguin Dictionary of Computing*, Penguin Group (excerpts) (1st ed.) (2001)
1056	File History of U.S. Patent No. 7,370,035
1057	Alfred and Emily Glossbrenner, *Search Engines For The World Wide Web*, PeachPit Press (3rd ed.) (2001)
1058	U.S. Patent No. 7,370,035 to Gross et al. ("**the '035 patent**")
1059	Adam A. Wilcox, *PC Learning Labs Teaches Lotus Notes 3.0*, Ziff-Davis Press (1993)
1060	N/A

No.	Description
1061	Stephen Londergan & Pat Freeland, *Lotus Notes R5 For Dummies Quick Reference*, IDG Books Worldwide, Inc. (1999)
1062	N/A
1063	Exhibit CX-0196 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1064	Exhibit CX-0198 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1065	Exhibit CX-0199 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1066	Exhibit CX-0203 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1067	Exhibit CX-0204 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1068	Exhibit CX-0206 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1069	Exhibit CX-0208 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1070	Exhibit CX-0209 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1071	Exhibit CX-0211 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)

No.	Description
1072	Exhibit CX-0213 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1073	Exhibit CX-0214 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1074	Exhibit CX-0215 to Complainant's Pre-Trial Brief in the Matter of *Certain Computing Devices Utilizing Indexed Search Systems and Components Thereof*, ITC-337-TA-1389 (August 29, 2024)
1075	U.S. Patent No. 9,663,139 ("**'139 patent**")
1076	File History of U.S. Patent No. 9,663,139

IV. UNDERSTANDING OF PATENT LAW

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

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

15. I understand that the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the term appears, but in the context of the entire patent, including the specification. I further understand that the principal considerations regarding the scope and

meaning of the claims are the plain language of the claim (including the surrounding claim language and context), the patent specification, and the prosecution history. I understand that while a claim is to be read in light of the specification, one must generally avoid importing limitations into the claim from the specification. I am also informed that the prosecution history can often inform the meaning of the claim by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be. I applied these understandings when considering the scope and meaning of the claims of the ’490 patent.

16. I understand that a prior art reference anticipates an asserted claim, and thus renders the claim unpatentable, if all elements of the claim are disclosed in that prior art reference, either explicitly or inherently (i.e., necessarily present or implied).

17. I further understand that a claim is unpatentable if it would have been obvious. Obviousness of a claim requires that the claim would have been obvious from the perspective of a POSITA at the time the alleged invention was made. I understand that a claim could have been obvious from a single prior art reference or from a combination of two or more prior art references.

18. I understand that an obviousness analysis requires an understanding of the scope and content of the prior art, any differences between the alleged invention and the prior art, and the level of ordinary skill in evaluating the pertinent art.

19. I further understand that a claim would have been obvious if it unites old elements with no change to their respective functions, or merely substitutes one element for another known in the field, and that combination yields predictable results. While it may be helpful to identify a reason for this combination, I understand that there is no strict requirement of finding an express teaching, suggestion, or motivation to combine within the references. When a product is available, design incentives and other market forces can prompt variations of it, either in the same field or different one. If a POSITA can implement a predictable variation, obviousness likely bars its patentability. For the same reason, if a technique has been used to improve one device and a POSITA would recognize that it would improve similar devices in the same way, using the technique would have been obvious. I understand that a claim would have been obvious if common sense directs one to combine multiple prior art references or add missing features to reproduce the alleged invention recited in the claims.

20. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include,

among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus—a connection—between any such secondary considerations and the alleged invention.

21. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

22. I understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the descriptions found in the patent being considered.

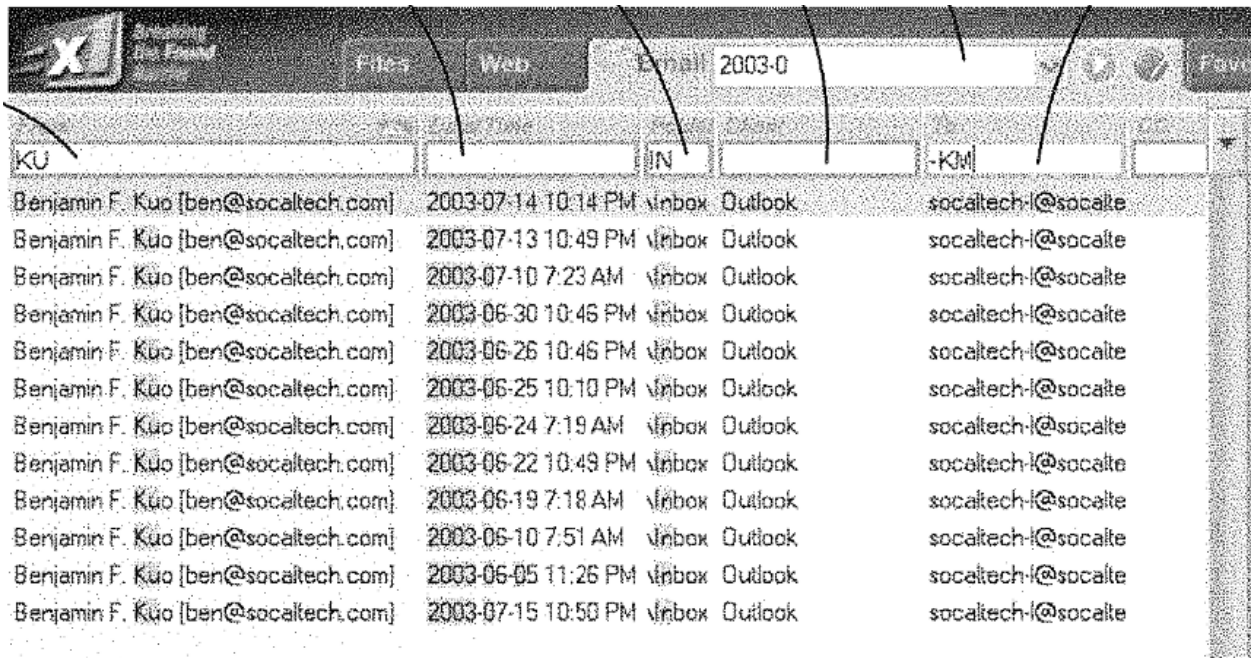
23. I understand that other challenges to the patentability of a patent, including patent ineligibility, enablement, written description, and definiteness or clarity of claim language, cannot be raised in IPR proceedings before the Board.

24. I understand that Petitioner has the burden of proving unpatentability by a preponderance of evidence, which means that the claims are more likely than not unpatentable.

25. The analysis in this declaration is in accordance with the above-stated legal principles.

V. THE ’490 PATENT

26. The ’490 patent generally relates 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. Specifically, the ’490 patent alleges that its improvement over then-existing delimited searches is the ability to provide incremental search results. 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). As an example of this incremental searching, FIG. 3G of the ’490 patent shows how when searching through emails, emails received from a user containing the entered search string are displayed as the user types their query, such that a query of “KU” returns emails from a contact with the last name of “Kuo.”



EX1001, FIG. 3G (excerpted).

27. The claims specifically recite incrementally searching two strings entered into the same search field. Independent claim 1 recites in part:

> receive a first partial search string in the first search field;
>
> receive a second partial search string in the first search field;
>
> wherein the first and second partial search strings are separated by a string separator character in the first search field;
>
> searching the index data structure for a first one or more of the plurality of files each including at least one string that corresponds to the first partial search string;
>
> searching the index data structure for a second one or more of the plurality of files each including at least

> one string that corresponds to the second partial search string; and
>
> updating a set of matching files of the plurality of files as each successive character of the first partial search string and the second partial search string is received respectively in the first search field by identifying any of the plurality of files included in both the first one or more of the plurality of files and the second one or more of the plurality of files, wherein the first partial search string and the second partial search string are non-adjacent in at least one of the matching files.

EX1001, 29:55-30:28.

28. The ’490 patent’s specification provides an illustrative example of this claim-recited functionality at 23:8-16:

> [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”).

29. Thus, the strings are partial in that they are prefixes (“Jo” for “John” and “D” for “Doe”). And the search is incremental in that the search results are

updated dynamically as the user adds additional letters to their search query. The ’490 patent’s dependent claims relate to specifics of the searching, including the types and fields of the files searched. In sum, the ’490 patent describes and claims locating indexed documents, including emails, that contain words beginning with two searched prefixes entered into the same search field, and updating the search results as the user types additional letters in the search field.

VI. LEVEL OF SKILL IN THE ART, AND STATE OF THE ART

A. Person Of Ordinary Skill In The Art

30. 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.

31. As demonstrated by the background references in Section VI.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 VI.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 VI.B.2);
- using indexes to expedite the search process (*infra* Section VI.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 VI.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 VI.B.5).

B. State Of The Art

32. 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 ground of unpatentability presented in Sections VIII-IX. These features were all well-known, widely deployed (including in commercial systems), discussed in numerous patents, and often combined with one another.

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

33. Although the ’490 patent touts that its alleged improvement over then-existing delimited searches is the ability to provide incremental search results (*e.g.*, EX1001, 1:43-53, 2:20-32), such incremental searching was well known long

before the patent's alleged 2002 priority date. For example, such incremental searching (i.e., incrementally locating and displaying search results to a user as they type their query) commonly used in, for example, text editors, search engines, digital encyclopedias, help centers, and email clients (EX1004, 8-9, 161; EX1009, 2:26-31, 2:39-48, 4:65-5:6; EX1010, 7:14-8:66; EX1011, Abstract, 2:49-3:58, 9:45-10:11; EX1012, [0026], [0051-53], [0167], [0247] (claim 1), FIGS. 8A-8C; EX1013, 10:14-28; EX1014, Abstract, 7:10-8:50; EX1015, Abstract, 1:16-21; EX1016, 2:51-67; EX1017, 9:50-10:10, FIGS. 8-11; EX1018, 5:16-6:40, 7:35-8:18, FIGS. 4-5, 8; EX1021, 125-127; EX1022, 55-57; EX1029, 12:66-15:60, FIG. 6; EX1039, 102-103; EX1043, 83-84, 337-338, 343-344, 355, 385-386, 404; EX1044, 3) at least because such incremental searching was known to provide several advantages over the other type of search, a delimited search (in which a user must type their query in full before initiating a search and receiving results), including: improved responsiveness, efficiency, and usability (EX1021, 124-127; EX1009, 2:25-48; EX1014, Abstract; EX1052, 518, 541).

34. As a first example, in 1978 Gabbe described using incremental searching to provide feedback on the volume of search results as the user typed their query. EX1014, Abstract, 7:10-8:50.

35. By 1985, at least one text editor (EMACS) employed incremental searching to locate search terms within a file. EX1021, 125-127; EX1022, (b).

36. And in 1994, the background section of at least one patent described "the incremental search method" as being known. EX1015, 1:16-21.

37. Other prior art references, including Schwartz, Sherwood, True, Nguyen, Spaey, Morgan, Kirsch, Siitonen, Shneiderman, and Colwell described using incremental searching in a variety of contexts, including to search emails (EX1004, 8-9, 161; EX1039, 102-103; EX1043, 385-389), search digitized audio files/transcripts (EX1009, 2:26-31, 2:39-48, 4:65-5:6), locate help articles (EX1010, 7:14-8:66), search digital encyclopedias (EX1011, Abstract, 2:49-3:58, 9:45-10:11), locate PDA contacts (EX1016, 2:51-67), locate locally-saved files on a hard drive (EX1017, 9:50-10:10, FIGS. 8-11), and locate Internet-based documents saved on one or more remote servers (EX1012, [0026], [0051-53], [0167], [0247] (claim 1), FIGS. 8A-8C; EX1013, 10:14-28). EX1044, 3.

38. As mentioned above, incremental searching's popularity was due at least in part to its known advantages over delimited searches, including responsiveness, effectiveness, and usability. By their very nature, incremental search tools are more responsive than delimited searches because they return results as the user types (i.e., results are updated every time the user types a new character). A POSITA knew that this immediate feedback makes incremental searches not only faster than delimited searches (thus "wast[ing] less of a user's time") (EX1021, 126), but also more effective and easier to use because users can

immediately tell whether their search query is working (i.e., returning relevant results) and can more quickly modify their search query to return more relevant results if necessary. EX1021, 126-127 ("Another advantage of incremental search is that, as the user enters the pattern, it provides constant feedback as to the results of the search. With a delimited search, [the user] does not find out whether [their] pattern is suitable or even if it was typed correctly until it is fully entered and a search attempt has been made. … Building a pattern incrementally allows the user to adjust the pattern interactively during the search, which leads the user to improve [their] search strategies through the feedback received."); EX1009, 2:39-48 ("The immediate visual feedback provided to the user can increase the speed with which the user achieves satisfactory search results. Because query results are displayed as the query is being input, the user need not wait until the full query is entered before viewing the results. This allows the user to modify a partially-entered query in response to unsatisfactory or unexpected results. Immediate display of segments of interest in response to modification of the interest level threshold increases the speed with which the user can identify segments of interest."); EX1044, 3 ("[A] very appealing alternative is that of 'dynamic queries': there is no Search button but the result set is continuously displayed and updated as phases of the search are changed. … [T]he advantages [of dynamic queries] are great: users can broaden, narrow, or refocus their search several times in as many

seconds."); EX1052, 518 ("*Dynamic queries*—in which users adjust query widgets to produce continuous updates—have proved to be effective and satisfying."), 541 (Dynamic queries employ "the use of rapid, incremental, and reversible actions" and provide "immediate display of feedback (less than 100 milliseconds). Additional benefits are the prevention of syntax errors and an encouragement of exploration."). Users also do not have to worry about "losing the search already accomplished" when modifying their search query (e.g., deleting characters). EX1021, 125-126.

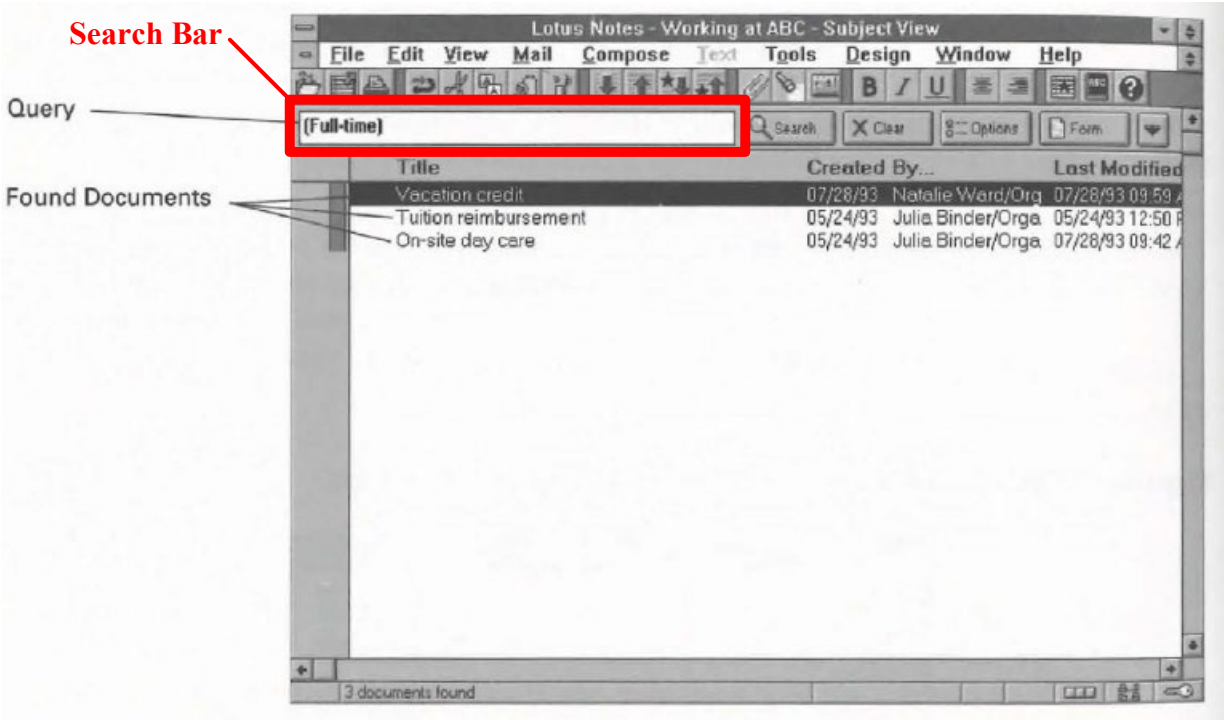
39. It also was known that the aforementioned benefits of performing an incremental search apply to multi-string Boolean searches, which I discuss in more detail below in Section VI.B.2. EX1021, 127 ("Even building a Boolean search pattern is made more effective when the early results of the pattern appear as the user adds to the pattern's specificity").

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

40. 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 in the art before 2002 and most, if not all, major search engines included such functionality. *E.g.*, EX1009, 2:26-31, 2:39-48, 4:38-43, 4:66-5:6, 5:41-50, FIG. 2; EX1019, 6:52-62,

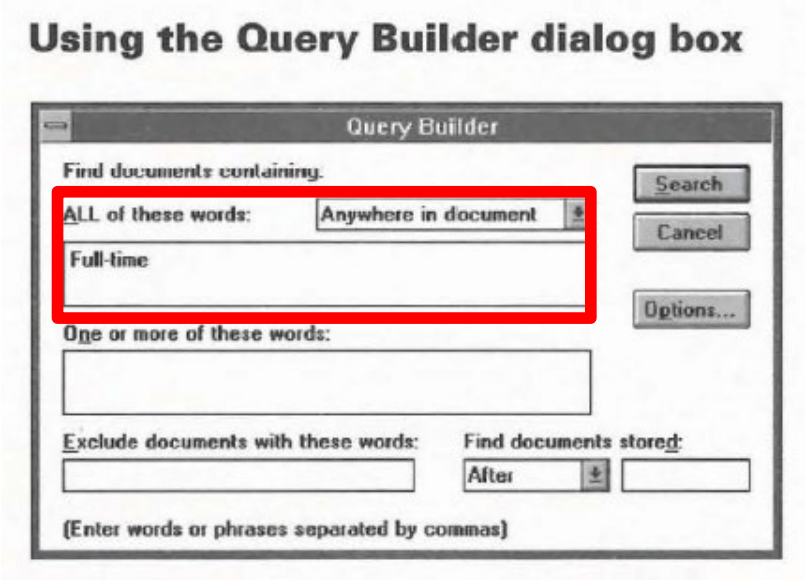
FIG. 5; EX1020, 2:22-3:16, FIG. 1; EX1057, 19-27, 58-62, 70-73, 82, 85, 89, 100, 105, 109, 121, 125, 129-130, 143, 146, 151, 166, 187, 196, 198, 215, 227; EX1059, 160-165; EX1061, 140-142.

41. For example, by at least 1993, Lotus Notes allowed users to search for mail containing multiple search strings by entering multiple search strings in a single search field, either by separating the strings with a Boolean AND operator in a simple Search bar, or by separating the strings with a comma in a Query Builder dialog box. EX1059, 160-161 ("Notes provides two tools for performing these searches, the Search bar and the Query Builder dialog box."), 162-164.



EX1059, 164 (red annotations added). "In the text box on the Search bar, enter the text for which you wish to search." EX1059, 161. "You can use the *logical*

operators AND … to find documents that contain various combinations of words and text strings." EX1059, 162.



EX1059, 163 (red annotations added). "To find documents containing all of several words, enter these words in the All Of These Words box. You should separate individual words by commas." EX1059, 162.

42. 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 search results; this was a commonly used, classical approach that dates back to at least the 1970s. 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."); EX1023, 25-26, 101-105, 195, 196 (describing how when using inverted indexes to search for documents containing multiple keywords, "[e]ach

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 in sequence (for a phrase) or appear close enough (for proximity). … [T]he most time-demanding operation on inverted indices is the merging or intersection of the lists of occurrences."), 207-208; EX1025, 109 ("In order to retrieve the data records that satisfy the conjunction, the resulting lists must then be intersected to determine the pointers to the data records desired."), 110 ("For each condition of a conjunction, we OR together the columns involved; then we AND the obtained result columns which yields the result corresponding to the conjunctions."); 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 ("FIG. 6 is a flow chart of a process of 'AND'ing two or more match lists to generate a new match list."), 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 . . .".); EX1026, 40; EX1033, 5:54-56; EX1041, 562; EX1044, 1-2; EX1052, 515; EX1053, 11; EX1057, 19-26, 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."), 58-62, 70-73, 82, 85, 89, 100, 105, 109, 121, 125, 129-130, 143, 146, 151, 166, 187, 196, 198, 215, 227; *see also* EX1007, 195.

43. Indeed, this was a commonly used, classical approach that dates back to at least the 1970s. By 2002, this classical approach was so well known, others did not even bother describing it in great detail. EX1020, 3:3-10 ("In the example of FIG. 1, the phrase 'merced microprocessor' in block 102 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.' Using Boolean operators in search queries is conventional and will not be discussed in great detail here for brevity's sake."); 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.”).

44. 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, it was common to determine not only where the search terms are located in a given document, but also their proximity to one another, and many search tools allowed users to specify an allowable distance between search terms (a so-called “proximity search”). EX1023, 101-103, 195-196; EX1017, 1:66-2:2, 7:59-66; EX1024, 12; EX1026, 40, 42, 46; EX1044, 2-6; EX1052, 515-518; EX1059, 160.

3. Using Indexes To Expedite The Searching Process

45. Long before 2002, it was common practice 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. *E.g.*, EX1013, 3:49-57, 6:31-47, 17:43-47; EX1023, 6 (“An old and popular data structure for faster information retrieval is a collection of selected words or concepts with which are associated pointers to the related information (or documents) — the *index*. In one form or another, indexes 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."), 7, 191-194; EX1032, 6:48-10:6; EX1053, passim; EX1054, 1:18-43 ("For some time now, existing systems have provided the capability to index the contents of documents … An index contains indexing data derived from a document that allows queries about the content of the document to be performed efficiently."); EX1055, 241, 256; EX1057, 9; EX1059, 159 ("For a database to support full-text searches, it must have a current *full-text index*."), 160, 169-171; EX1061, 135-139.

46. Indexes expedite searches by cataloging which documents (e.g., email messages) contain a given search term, how many times the search term appears in each document, and/or where the search term appears in each document. EX1023, 7-9, 19, 191-194; EX1032, 6:48-10:6; EX1053 passim; EX1054, 1:18-43; EX1055, 241, 256; EX1057, 9 For example, for each term in an inverted index there is a "list … consisting of the identifiers of the documents containing the term and … the frequency," of the term in each document. EX1053, 12; *see also*, *e.g.*, EX1023, 192-194; EX1054, 1:18-32, 1:33-37 ("The indexing data of a multiple document index contains each major word of the documents, the document identifiers of the documents in which each major word is located, and the locations within each document of the major word."), 1:38-2:65.

Document Index 201

202 Keyword	203 Document Identifier, 204 (Occurrence)
Areas	(B, 12)
Cases	(A, 6)
Congress	(B, 1)
Court	(A, 10)
Courts	(A, 3, 16) (B, 6)
Created	(B, 3)
Disputes	(B, 9)
District	(A, 15)
Federal	(A, 9, 14) (B, 5)
Handle	(B, 8)
Law	(B, 14)
Most	(A, 5)
Particular	(B, 11)
Specialized	(B, 4)
System	(A, 11)
Trial	(A, 2)

205 Tuple

206

Two occurences of the word "Courts," in document A, one at position 3, another at position 16

Figure 2

EX1054, FIG. 2 (red annotations added); *see also id.*, 2:10-12.

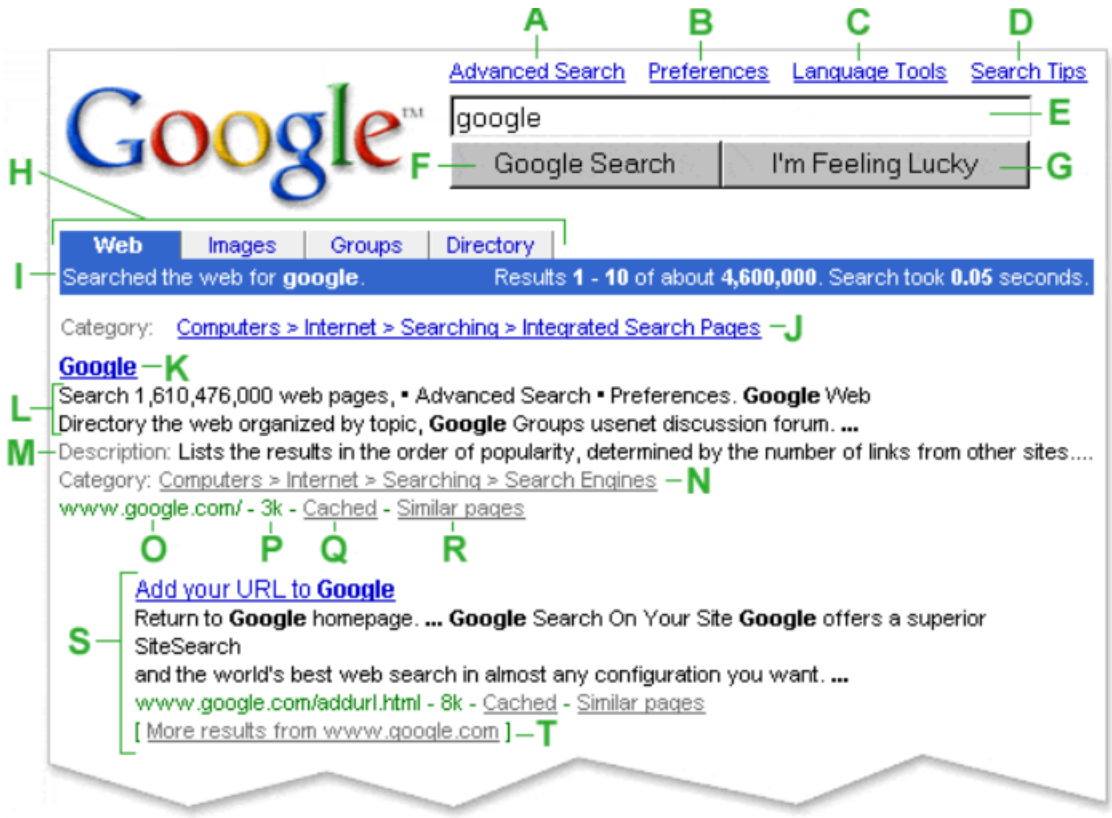
47. Thus, rather than scanning every word of every document sequentially, the search tool can more quickly and easily search the index to locate the documents that contain a given search term. *E.g.*, EX1023, 191.

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

48. 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. EX1023, 101, 103 ("Besides selecting the appropriate documents, the [information retrieval] system may also … highlight the occurrences within the documents of the words mentioned in the query."), 207 ("the final phase [of the search] retrieves the exact positions of the matches to highlight them in those documents that the user actually wants to see."), 289 ("[I]t is often useful to *highlight* the occurrences of the terms or descriptors that match those of the user's query. It can also be useful for the system to scroll the view of the document to the first passage that contains one or more of the query terms, and highlight the matched terms in a contrasting color or reverse video. This display is thought to help draw the user's attention to the parts of the document most likely to be relevant to the query. Highlighting of query terms has been found time and again to be a useful feature for information access interfaces."); EX1027, 3-10 ("In-document match highlighting and search capabilities can also support user confidence. When a document is selected for viewing from a search results list, the search terms matched in the document text should be highlighted. Facilities should also be provided to permit the user to move rapidly, forward or backward,

between successive highlighted search terms. This would permit staff to quickly find the portion of the displayed text that pertains to the query, and then to move through the document identifying any successive matches."); EX1017, 4:53-64, FIG. 2; EX1020, FIG. 1; EX1036, 6:8-20, FIG. 5; EX1030, Abstract, 2:29-34, 4:8-11, 7:15-28, 7:41-50, FIG. 9; EX1031, 17:28-33, FIG. 23; EX1033, 1:39-2:48, 5:41-7:43, FIG. 7; EX1034, FIGS. 55-59; EX1035, 16:57-17:5, 27:41-50, 28:7-10, 28:50-56, FIGS. 8-10; EX1038, 71-72 (APPX R); EX1057, 63, 67, 86-88, 163.

49. As just one example, Google's pre-2002 search results would display the portions of located documents containing the searched term(s) and would bold the searched terms in the search results to highlight the search terms to the user in the displayed portions of text. EX1028, 119 ("A typical Google summary contains an excerpt from the Web page where the search terms are shown highlighted *in the context of this page*."); EX1038, 56 (APPX L); EX1033, Title, Abstract, 6:49-59, FIG. 7; *see also* EX1033, 6:52-56 ("In this example, the user has searched for the term 'google' and has selected a link on the Google web site for a document relating to 'How to Google.' As shown, the browser assistant [] highlights each occurrence of the word 'google' in the document."); EX1057, 86-88.



EX1038, 56 (APPX L). *See also id*. (describing that the green "L" in the above figure corresponds to "an excerpt from the returned result page with your query terms bolded. These excerpts let you see the context in which your search terms appear on the page, before you click on the result.").

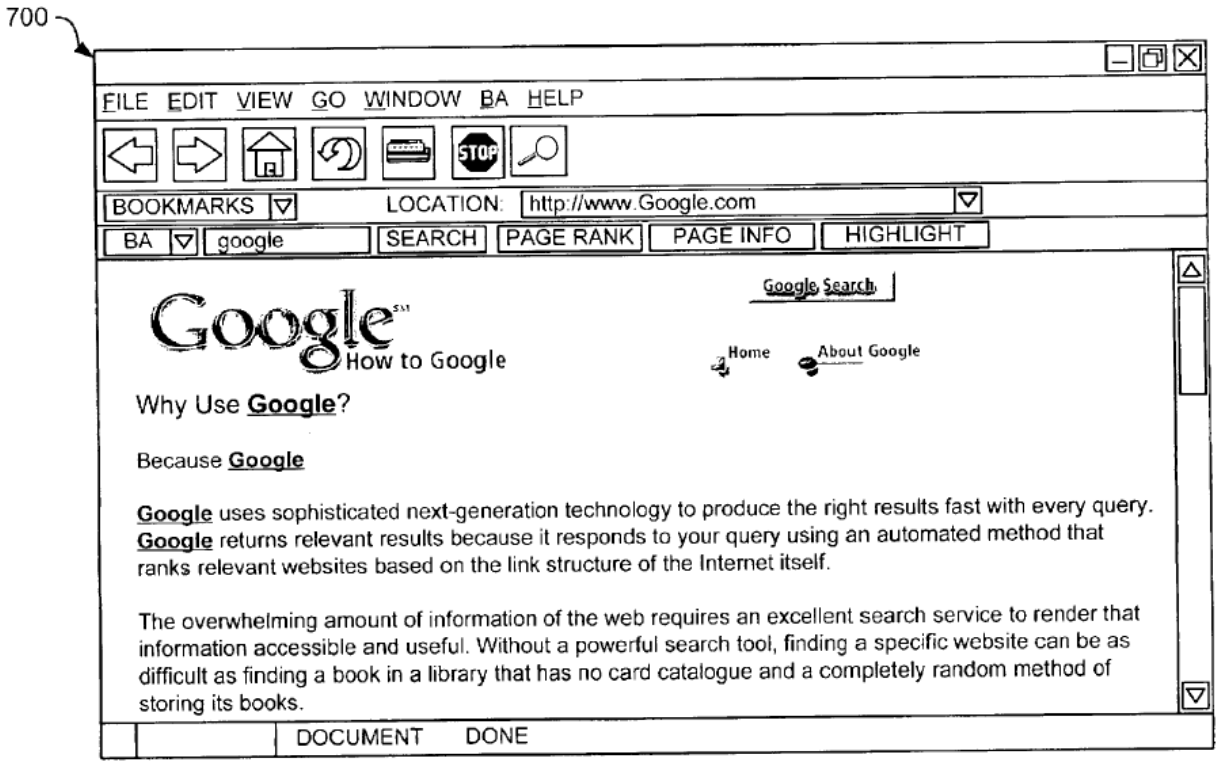


FIG. 7

EX1033, FIG. 7.

Human Genome Project History
SHORT **HISTORY**. Russian State **Human Genome Project** was started in 1988 when professor Alexander ...
www.eimb.relarn.ru/hugo/01.htm - 3k - Cached - Similar pages

History of the Human Genome Project
... The DOE **Human Genome** Program and the NIH National **Human Genome** Research Institute (NHGRI) together make up the US **Human Genome Project**. **History** of the **Project**. ...
www.ornl.gov/hgmis/project/hgp.html - 42k - Cached - Similar pages

1991 History of the US Human Genome Project
... A brief **history** of the US Department of Energy (DOE ... as those of the collaborative US **Human Genome Project**. The Office of Health and Environmental ...
www.ornl.gov/hgmis/project/91history.html - 40k - Cached - Similar pages

History of the UK Human Genome Project
... **History** of the UK **Human Genome Project** 1990 - Drs Sulston and Coulson in collaboration with Dr Waterston (Washington University, USA) began a pilot study of ...
www.wellcome.ac.uk/en/genome/history.html - 14k - Cached - Similar pages

Figure 6.24 To produce these results, we searched for `allintitle:human genome project history`. Note that all of our search terms appear in bold in the Web page titles.

EX1057, 88.

50. As another example, Shneiderman discloses that the Xinquery search engine similarly highlighted every occurrence of search strings in the located documents. EX1044, 5-6.

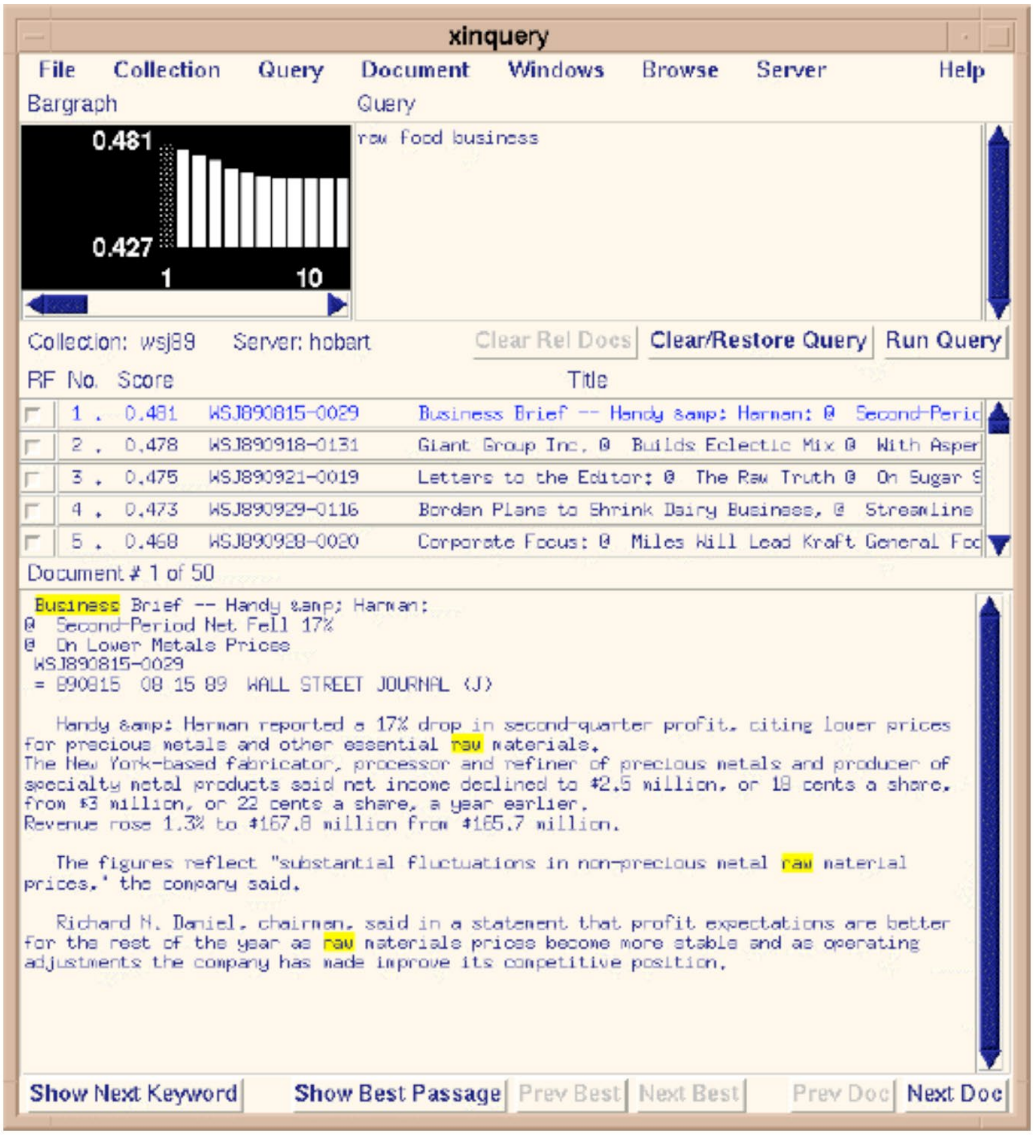


Figure 1. XINQUERY

EX1044, 5.

51. Two other search engines, Yahoo and AltaVista, also highlighted the search terms (by bolding the search terms) in the search results regardless of whether they appeared adjacent or non-adjacent to one another. EX1057, 63, 65, 74-75, 163.

1. The *Mozart* Project
The life, times and music of Wolfgang Amadeus **Mozart**. Includes a complete Köchel listing, bibliography, **biography**, links to related sites and...
URL: www.frontiernet.net/~sboerner/mozart/
Directory Match: Wolfgang A Mozart Bios & Works
Translate More pages from this site Related pages Facts about: Frontier Corp

2. *Mozart* Discussion
Discussion of Wolfgang Amadeus **Mozart** on the web....
URL: mozart.composers.net/wwwboard/index.html
Translate More pages from this site Related pages

3. The *Mozart* Project: Bibliography
HOME | **BIOGRAPHY** | COMPOSITIONS | SELECTED ESSAYS | BIBLIOGRAPHY
Mozart's Works and Writings. Anderson, Emily, ed. The...
URL: trpe.reflektor.cz/mozart/books/index.html
Translate More pages from this site Related pages

4. *Mozart, Wolfgang Amadeu: Biography*
Site ©1996 Timothy A. Smith. Wolfgang Amadeus **Mozart** 1756-1791.
Child prodigy **Mozart** made his first public appearance, at six years of age, in a...
URL: jan.ucc.nau.edu/~tas3/wamozart.html
Translate Related pages Facts about: Northern Arizona ...

Figure 5.23 Here's an example of how AltaVista presents search results for a query that includes the terms *Mozart* and *biography*.

EX1057, 63.

Yahoo! Category Matches (1 - 1 of 1)

Business and Economy > Shopping and Services > Antiques and **Collectibles** > Mechanical Musical Instruments

- Music Boxes

Yahoo! Site Matches (1 - 19 of 34)

Regional > U.S. States > Arizona > Cities > Phoenix > Business and Shopping > Shopping and Services > Antiques and **Collectibles**

- Sorrento **Music Boxes** - offers Italian inlaid **music boxes**, jewelry boxes, and musical Capodimonte carousels.

Business and Economy > Shopping and Services > Antiques and **Collectibles** > Mechanical Musical Instruments

- Mechantiques - dealer in antique mechanical musical instruments. Including disc or cylinder type **music boxes**, musical clocks, nickelodeons and more. Buy, sell, trade.
- Mechanical Musical Memories - **music boxes**, phonographs, nickelodeons, player pianos, and band organs.

Business and Economy > Shopping and Services > Books > Booksellers > Recreation and Sports > Antiques and **Collectibles**

- Arlington Book Co. - specializes in books on barometers, clocks, fountain pens, furniture, jewelry, **music boxes**, tools, trademarks, watches, wristwatches, and related subjects.

Figure 10.8 Here's an example of how Yahoo presents search results. Category Matches come first, followed by Site Matches.

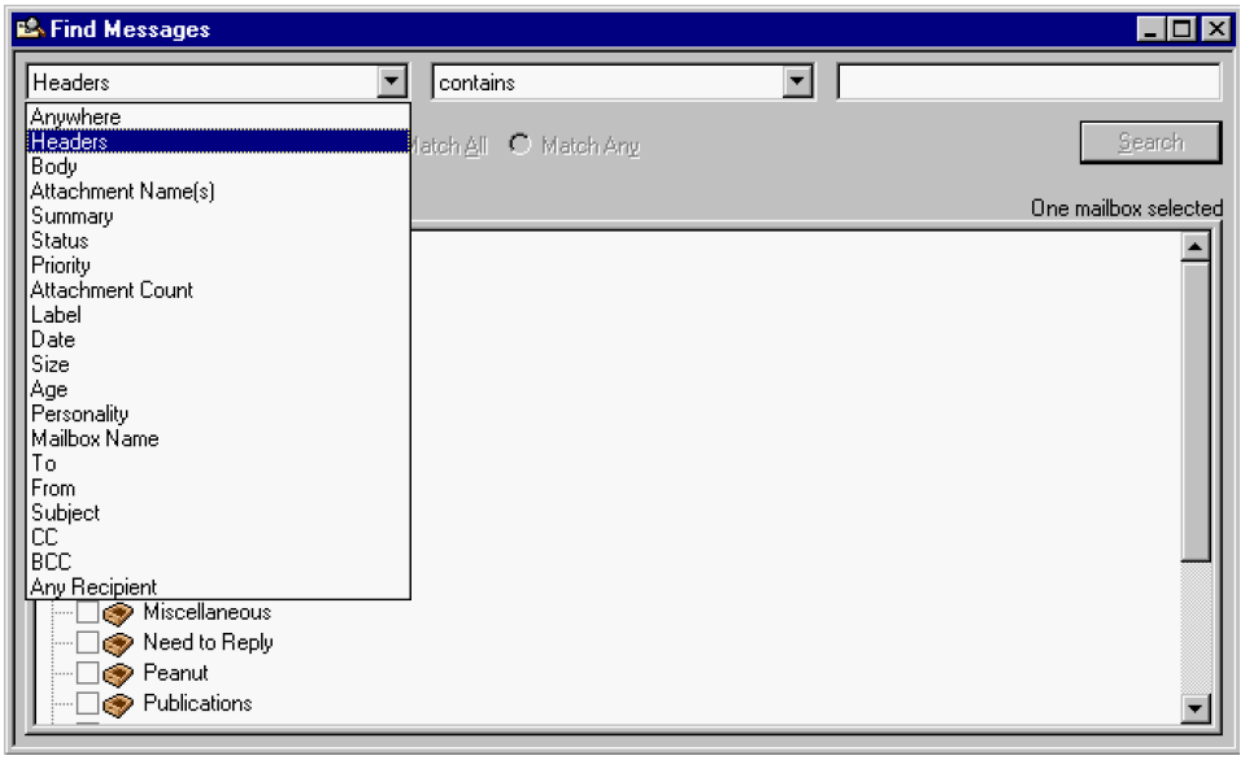
EX1057, 163 (showing results for a search of "music boxes" AND "collectible"); *see also id*., ("Your search terms will be displayed in bold to make them easier to spot.").

52. It also was known to highlight search terms in emails, specifically. EX1004, 11, 163.

5. Email-Specific Searching

53. 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. EX1004, 161-164; EX1006, 384-387; EX1007, 129-130; EX1008, 268-269, 471-478; EX1032, 5:51-56, 11:35-41, 11:64-12:11; EX1037, [0088]; EX1039, 100-105.

Find Messages drop-down and conjunction options lists



EX1007, 129.

VII. CLAIM CONSTRUCTION

54. I understand that, in IPR, claim terms should be given the ordinary meaning that the terms would have to a person of ordinary skill in the art on the earliest effective filing date, in view of the specification and file history.

55. For this declaration, I apply the plain and ordinary meaning of all claim terms. That said, I reserve the right to offer my opinions on any constructions or construction issues presented by Patent Owner.

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

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

57. 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 VI.B.1, VI.B.2), however, and Raskin expressly discloses (1) and Wu expressly discloses (2). Thus, claims 1-20 are obvious over Lotus Notes in view of Raskin and Wu.

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

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

59. 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, 1:29-30, 2:9-21, 3:34-62, 4:28-54, FIGS. 3A, 3G-3I; EX1059, 83-85, 159-171; EX1061, 40-41, 133-148.

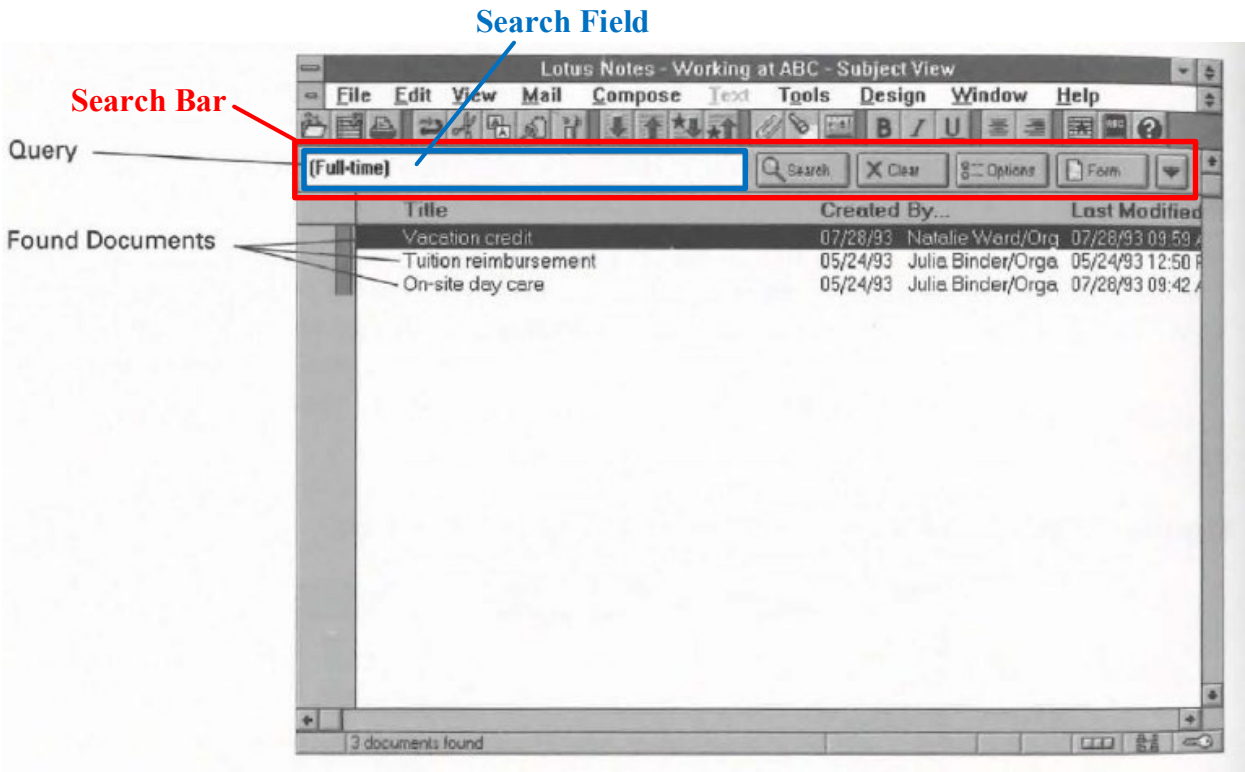
60. The Lotus Notes references explain how Lotus Notes works and how to use it. EX1059, passim. Londergan (EX1061) is directed to Lotus Notes R5 and Wilcox is directed to Lotus Notes 3.0 (an earlier version of Lotus Notes). EX1059, (a); EX1061, (i). Though Londergan explains some of the new features of Lotus Notes 5, including its calendaring and auto-complete features (EX1061, 45-74, 146-147), Wilcox provides additional details on the user interface and search functionality (EX1059, 84-85, 159-168) that a POSITA would have understood to also be present, or at least obvious to include, in Lotus Notes 5.

61. 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 files (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).

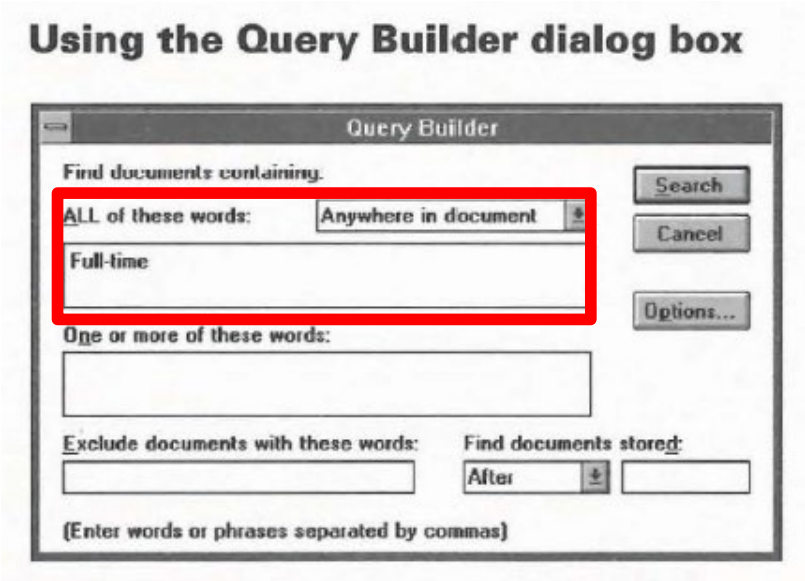
62. For example, the Lotus Notes references disclose two search tools—a simple Search bar and a more advanced Query Builder / Search Builder. EX1059, 160-161 ("Notes provides two tools for performing these searches, the Search bar and the Query Builder dialog box."), 162-168; EX1061, 133-148.

63. For the simple Search bar, the user enters multiple search strings in the search field of the Search bar and separates the search strings with a Boolean AND operator. EX1059, 161 ("In the text box on the Search bar, enter the text for which you wish to search."), 162 ("You can use the *logical operators* AND … to find documents that contain various combinations of words and text strings."); EX1061, 140 ("The Search bar contains several important parts. On the left, the blank box is where you type the text that you want to search for. … Words connected by *and* … [f]or example, [] **gold and silver** [] find[s] documents that contain both the word *gold* and the word *silver*.").



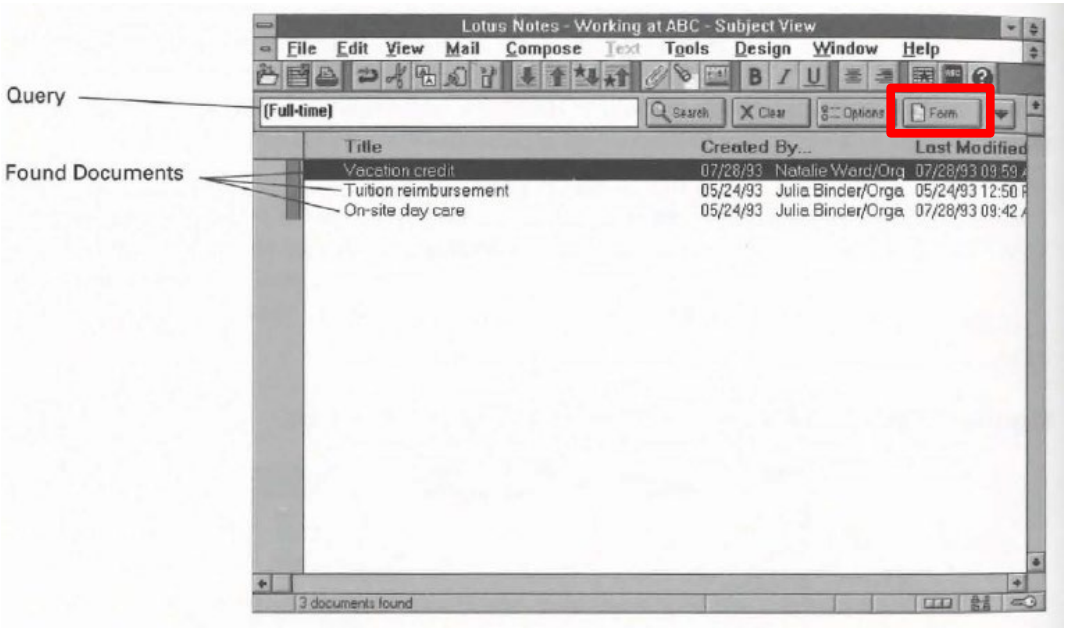
EX1059, 164 (blue and red annotations added).

64. For the Query Builder / Search Builder, the user similarly enters the multiple strings in a single search field but separates the search strings with a comma instead of a Boolean AND operator. EX1059, 162 ("To find documents containing all of several words, enter these words in the All Of These Words box. You should separate individual words by commas."); EX1061, 141-142.



EX1059, 163 (red annotations added).

65. Lotus Notes also discloses searching within specific fields, including within the title, category, and/or body fields of the files, by selecting the "Form" button. EX1059, 166-168; EX1061, 143-144.



EX1059, 164 (red annotations added).

Figure 5.20 **Searching using a form**

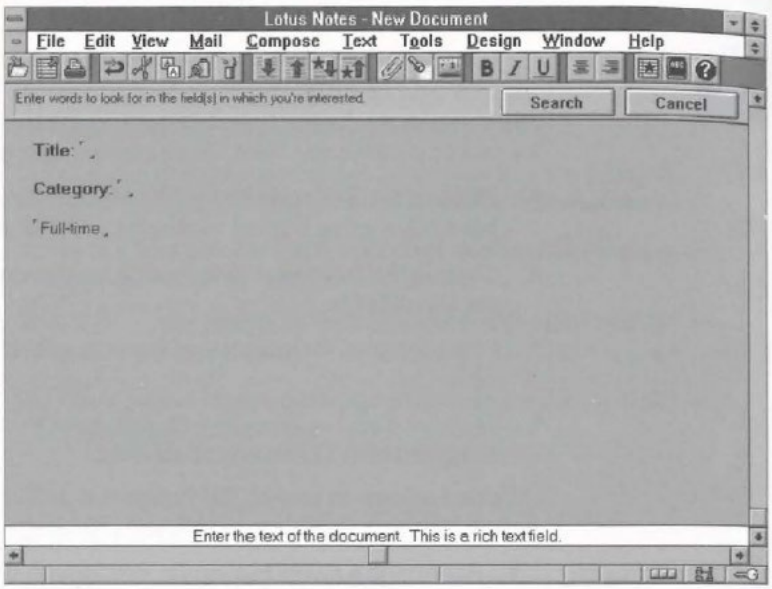


Figure 5.21 **A query created using a form**

EX1059, 168.

66. 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.

B. <u>Overview Of Raskin (EX1021)</u>

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

68. 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, 1:29-30, 2:9-21, 3:34-62, 4:28-54, FIGS. 3A, 3G-3I; EX1021, 118-132.

69. 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, 126 ("[T]he advantages of incremental searching are so numerous … that I can see almost no occasions when a delimited search would be preferred."), 127.

70. In a delimited search, a user "types the pattern, followed by [either] a delimiter … such as Return" or "click[ing] a button with a label such as OK, Search, Find, or Find Next." EX1021, 125. An incremental search, on the other hand, "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.

71. 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.

C. Overview Of Wu (EX1019)

72. 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.

73. 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, 1:29-30, 2:9-21, 3:34-62, 4:28-54, FIGS. 3A, 3G-3I; EX1019, Abstract, 3:24-43, 6:51-9:33, FIGS. 3-6.

74. Specifically, Wu describes a process for locating files containing at least two search strings 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, 8:66-9:44. For example, Wu describes how the search tool locates files containing the words “Game” and “Go” by searching each term separately, generates two match lists, one for files that contain the word “Game” and another for files that contain the word “Go,” and then identifies the files present in both sets of match lists by performing an AND operation (i.e., an intersection of the two match lists). EX1019, 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."), 8:66-11:32, FIGS. 5-6.

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

75. 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 using an intersection approach (i.e., by searching each string independently and then identifying the files that appear in both sets of search results).

76. 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. For example, Wilcox provides additional details on the core functionality of Lotus Notes' user interface and search functionality (EX1059, 84-85, 159-168) while Londergan provides details on new features of Lotus Notes R5, including its calendaring and auto-complete features (EX1061, 45-74, 146-147). A POSITA would have looked to Wilcox to gain a solid understanding of Lotus Notes' core functionality, and would have

looked to Londergan for details on the new features in Lotus Notes 5 that were added to this core functionality.

77. 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, the core search functionality of Lotus Notes' search tool (e.g., the Search bar) is largely, if not completely, the same between the two versions, and a POSITA would have understood that any uniquely described functionality for one version would be compatible with the other version given how similar the search tool is in these two versions of the software.

78. 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.").

79. 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 (i.e., 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/intersecting the results (i.e., identifying the files present in the first and second sets of files), 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 VI.B.2.

80. 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.

81. In sum, a POSITA would have been motivated with a reasonable expectation of success to modify Lotus Notes' multi-string, single-field search tool to locate files incrementally, as taught by Raskin, and to locate the files containing both search strings by searching each string separately and then intersecting the results, as taught by Wu.

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

1. Claim 1

a) Element [1.1]

[1.1] A computing system comprising:

82. 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.

b) Elements [1.2-1.4]

[1.2] one or more hardware computer processors;
[1.3] a display device viewable by a user of the computing system;
[1.4] a computer readable storage medium storing:

83. 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, 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.

c) Element [1.5]

[1.5] an index data structure comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included; and

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

d) Element [1.6]

[1.6] software instructions executable by the one or more hardware computer processors causing the computing system to:

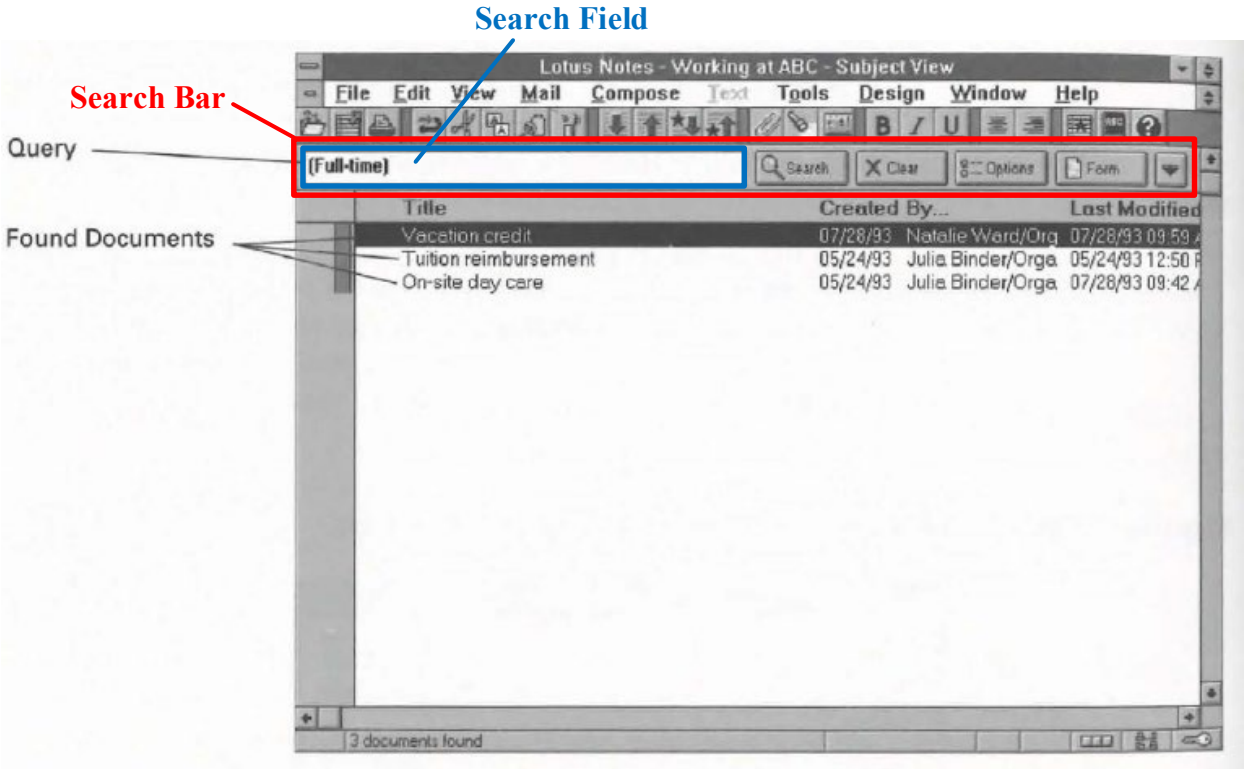
85. The Lotus Notes references teach this claim element at least because they disclose software instructions (the software that runs Lotus Notes) that are downloaded to the computing system's memory. EX1059, xx, xxv-xxix, 3-5, 6 ("[T]he Notes workstation software must be installed on your computer's 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 includes software instructions that are 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.

e) Element [1.7]

[1.7] cause display of an interactive user interface on the display device, the interactive user interface including at least a first search field;

86. 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.

87. 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).

f) <u>Elements [1.8-1.10]</u>

[1.8] receive a first partial search string in the first search field;
[1.9] receive a second partial search string in the first search field;
[1.10] wherein the first and second partial search strings are separated by a string separator character in the first search field;

88. The Lotus Notes references teach receiving a first search string (a word such as "gold") and a second search string (another word such as "silver") in

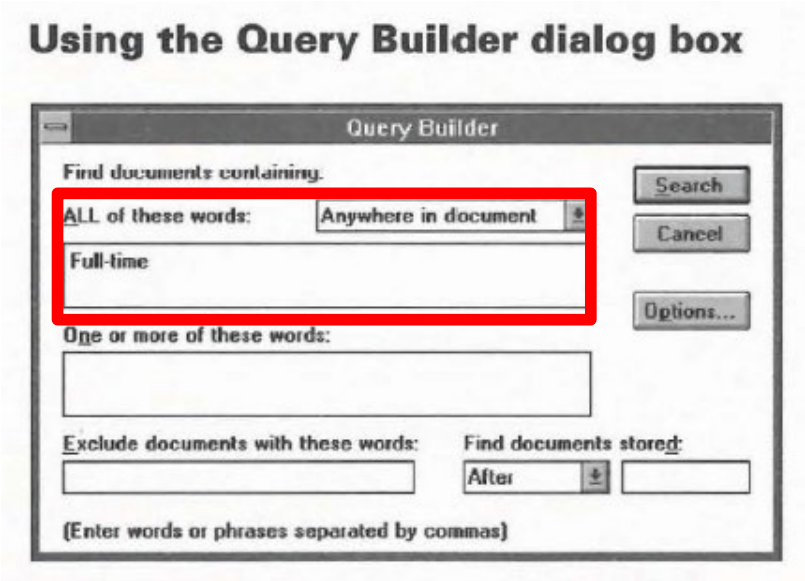
a single search field (e.g., a blank box of a Search bar and/or a blank box of a Query Builder / Search Builder dialog box) separated by a string separator character (e.g., a Boolean AND operator and/or a comma). EX1059, 159-168; EX1061, 17, 133-148, 195.

89. For example, the Lotus notes references describe receiving two search strings such as "gold" and "silver" in the search field of the Search bar separated by a Boolean AND operator. EX1061, 17, 140-141, 195; *see also* EX1059, 159-162.

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."). To the extent the Boolean AND operator is not considered a string separator character (which I do not agree with), 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 instead of the three-character Boolean AND operator, 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.

90. As another example, the Lotus Notes references describe receiving two search strings in the search field of the Query Builder / Search Builder dialog box separated by a comma. EX1059, 159-165; EX1061, 142-144.



EX1059, 163 (red annotations added). *See also id.*, 162 ("To find documents containing all of several words, enter these words in the All Of These Words box. You should separate individual words by commas."); EX1061, 141-142.

91. When implementing Lotus Notes' search tool to perform incremental searches, as taught by Raskin (*see infra* Section VIII.E.1.g), a POSITA would have understood or at least found it obvious that the received search strings are partial search strings because a purpose of incremental searches is to provide a user with

results as they type in their search query so that they do not have to type the full search strings before seeing results . *E.g.*, EX1021, 126-127.

g) <u>Elements [1.11-1.13]</u>

> [1.11] searching the index data structure for a first one or more of the plurality of files each including at least one string that corresponds to the first partial search string;
>
> [1.12] searching the index data structure for a second one or more of the plurality of files each including at least one string that corresponds to the second partial search string; and
>
> [1.13] updating a set of matching files of the plurality of files as each successive character of the first partial search string and the second partial search string is received respectively in the first search field by identifying any of the plurality of files included in both the first one or more of the plurality of files and the second one or more of the plurality of files, wherein the first partial search string and the second partial search string are non-adjacent in at least one of the matching files.

92. The Lotus Notes+Raskin+Wu combination teaches these elements. The Lotus Notes references disclose locating files in an index data structure containing both the two 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 VI.B.1) 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.

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

93. As explained above in Section VIII.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.

94. 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.").

95. 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. The user types a character and the incremental search tool locates all files containing words that start with that character, and then as the user types additional characters, the search tool narrows the results to only files containing those additional characters. EX1021, 125-127.

96. As also explained above in Section VIII.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.

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

97. As explained above in Section VIII.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 VI.B.2. In particular, a POSITA would have recognized that there were only 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.

98. 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. That is, a POSITA would have understood that the intersection approach's separate treatment of the search strings would provide quick results for such an incremental search because it would be able to efficiently update search results in response to modifications to either search string. For example, if a user were to modify only one string, the sequential search approach would have to completely rerun the search because it searches for both strings simultaneously, whereas the intersection approach could preserve previous results for the unmodified search string, and would only have to update results for the modified search string.

99. As also explained above in Section VIII.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.

Using An Index For The Search

100. 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.

Including Files In Which The Search Strings Are Non-Adjacent

101. 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.

102. 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 VI.B.2.

2. <u>Claims 2-3</u>

2. The computing system of claim 1, wherein the set of matching files includes a first file wherein the first partial search string precedes the second partial search string.
3. The computing system of claim 2, wherein the set of matching files includes a second file wherein the second partial search string precedes the first partial search string.

103. The Lotus Notes+Raskin+Wu combination renders these claims obvious at least because, as explained above, a POSITA would have understood that the Lotus Notes' disclosed Boolean search is order-agnostic because it cares only about whether a file contains the search strings, not the location of the search strings in the file. *Infra* Section VIII.E.1.g. 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 where the second partial search string precedes the first partial search string, as was well known in the art. *Supra* Sections VI.B.2, VIII.E.1.g.

3. <u>Claim 4</u>

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

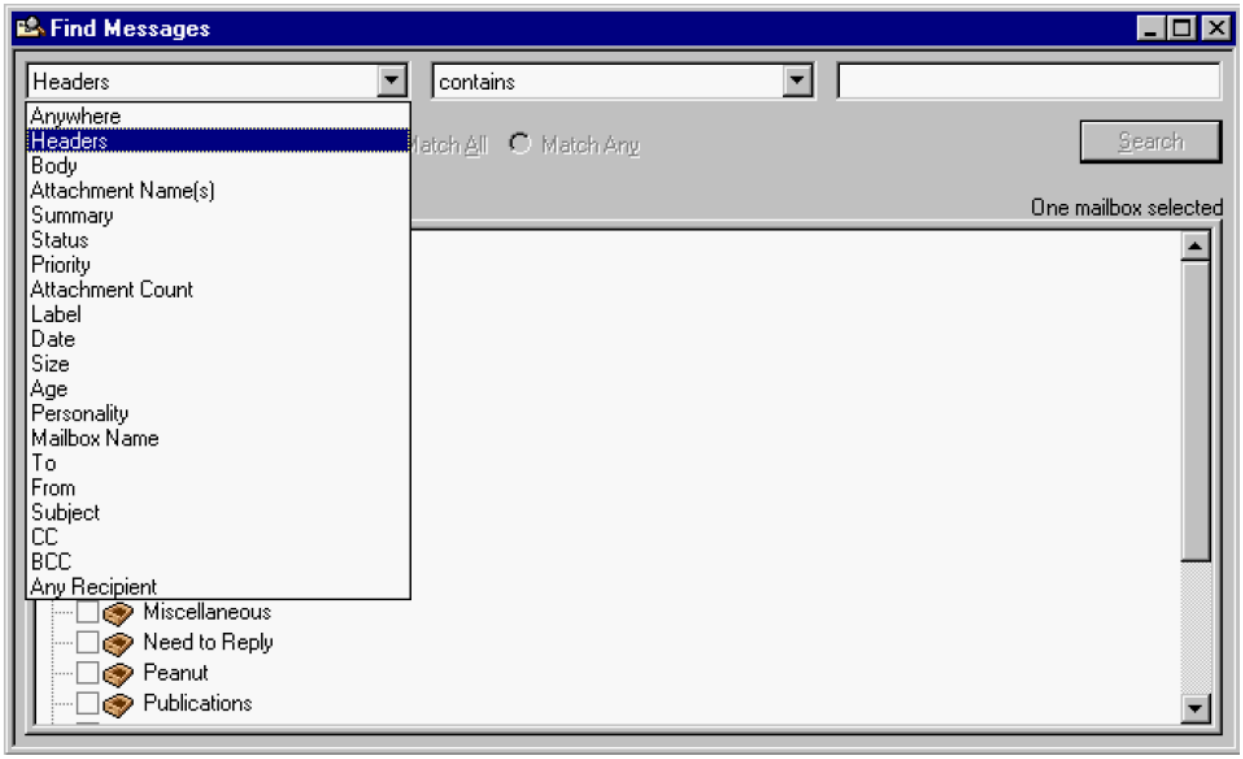
104. The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because a POSITA would have understood or at least found it obvious that the index data structure comprises indications of character strings associated with respective attribute fields of the files at least because the Lotus 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.

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

Find Messages drop-down and conjunction options lists



EX1007, 129.

106. 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.

4. Claims 5-7

> 5. The computing system of claim 1, wherein the interactive user interface includes at least:
>
> an email search mode interface having email-specific attribute search fields including at least a date field, a from field, and a sender field;
>
> a document search mode interface having document-specific attribute search fields including one or more of a file name field, a file type field, a date field, a file size field, and a path field; and
>
> an email attachment search mode interface including one or more of a name field, a date field, a size field, and an extension field.
>
> 6. The computing system of claim 1, wherein the index data structure is generated and updated by an index module that generates at least an email index corresponding to a first plurality of the files that are emails, a document index corresponding to a second plurality of the files that are documents, and an email attachment index corresponding to a third plurality of files that are email attachment files.
>
> 7. The computing system of claim 1, wherein the index data structure includes an email index, a document index, and an email attachment index.

107. The Lotus Notes+Raskin+Wu combination renders these claims 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).

108. 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 (i.e., the user is less likely to be overwhelmed with a high volume of 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* EX1061, 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.

5. Claim 8

8. The computing system of claim 1, wherein the first partial search string comprises characters corresponding to at least a portion of a first string being searched for and the second partial search string comprises characters corresponding to at least a portion of a second string being searched for.

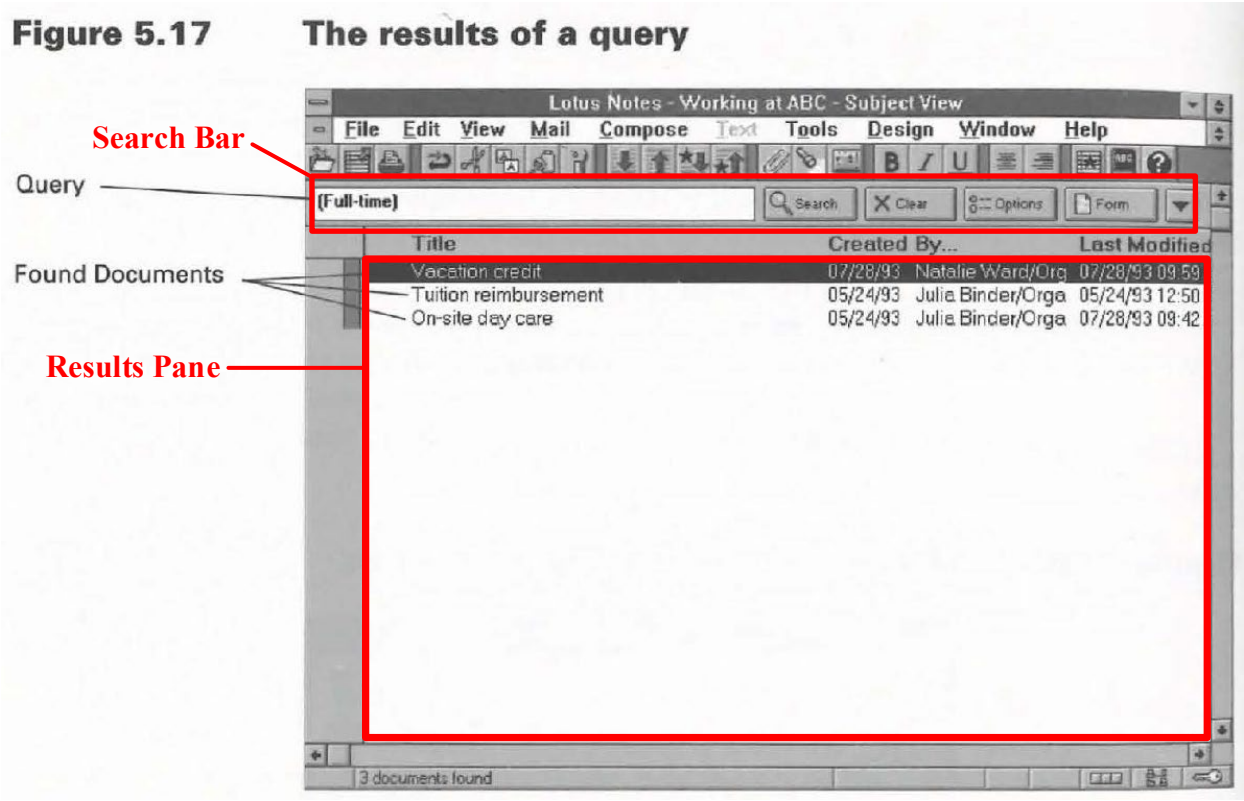
109. This claim is obvious at least because a POSITA understood that in an incremental search, a partial search string is a portion of string being searched for, such as the beginning of a word. *Supra* VII.E.1.f-VII.E.1.g; EX1021, 125-127.

6. Claim 9

9. The computing system of claim 1, wherein the software instructions further cause the computing system to display in the interactive user interface indications of the set of matching files.

110. 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.

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



EX1059, 164 (red annotations added).

7. <u>Claims 10-11</u>

> 10. The computing system of claim 9, wherein the software instructions further cause the computing system to: receive selection of a first file of the displayed set of matching files; and display in the interactive user interface a portion of the first file.
>
> 11. The computing system of claim 10, wherein the software instructions further cause the computing system to: highlight a plurality of occurrences of the first and second partial search strings in the first file while not highlighting portions of text between occurrences of the first and second partial search strings in the first file.

112. The Lotus Notes+Raskin+Wu combination renders these claims obvious at least because the Lotus Notes references disclose displaying a portion of a selected file and highlighting occurrences of only the searched strings and not

any of the text between the searched strings. EX1059, 164 ("The found text [] is indicated by a red box."), 165; EX1061, 145, 146 ("Notes [] shows you the document and highlights the text that you wanted to find."). 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 VI.B.4.

8. <u>Claim 12</u>

12. The computing system of claim 1, wherein the string separator character is selected from: a space, "<", "(", "[", "{", "‘", "!", "@", "#", "$", "%", "^", "&", "*", ")", ";", ":", "’", "?", ">", ",", "=", and "+".

113. 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, 4:23-5:10; 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,

9. Claim 13

> 13. The computing system of claim 1, wherein the files comprise emails, text documents, contact information, appointment information, task information, music files, system files, photographs, videos, and/or images.

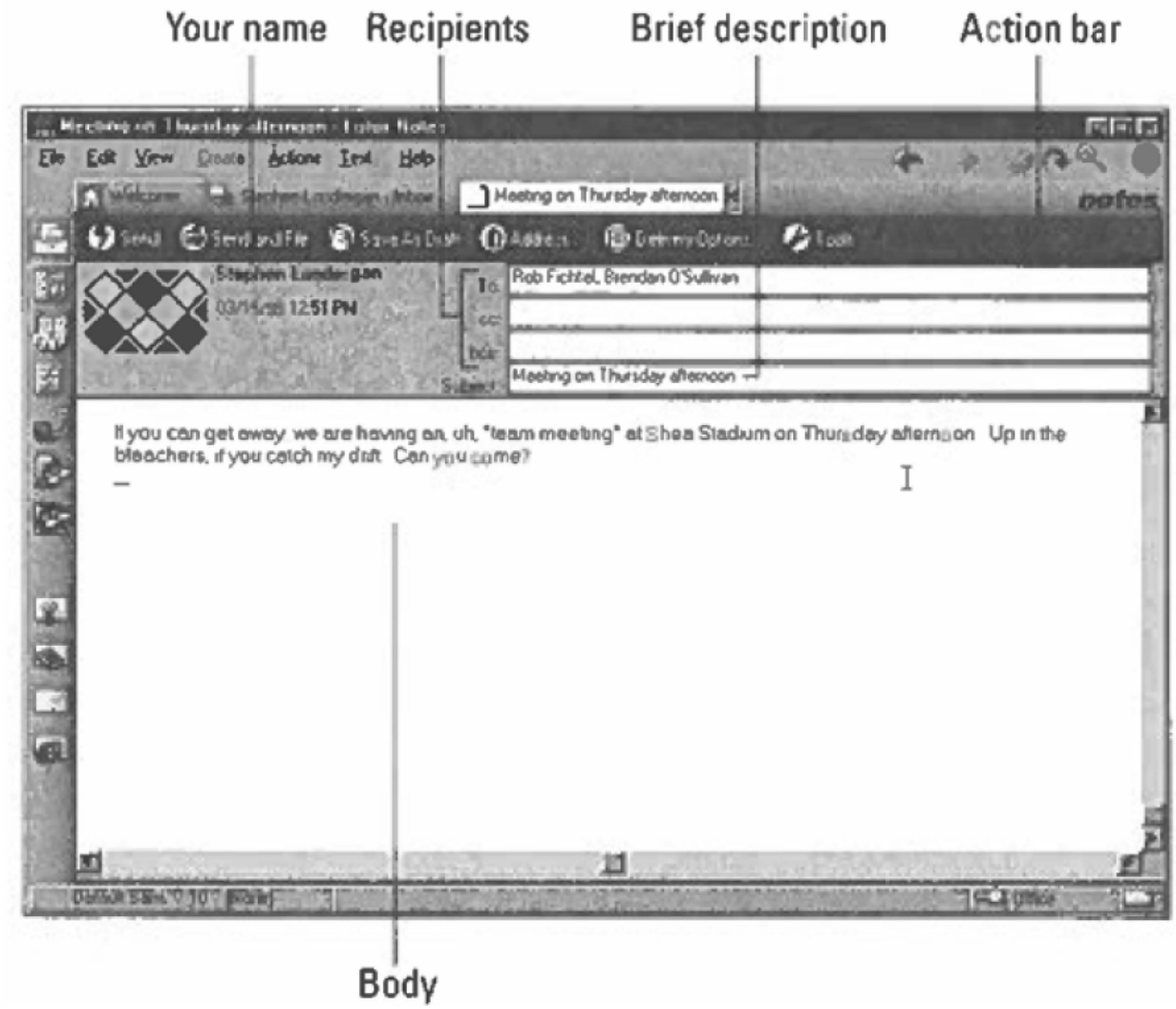
114. The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose that the files comprise, *inter alia*, emails (EX1059, 93-134; EX1061, 21-44), text documents (EX1059, 321, 324; EX1061, 5-6, 188), contact information (EX1059, 96, 121, 132; EX1061,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

10. Claims 14-15

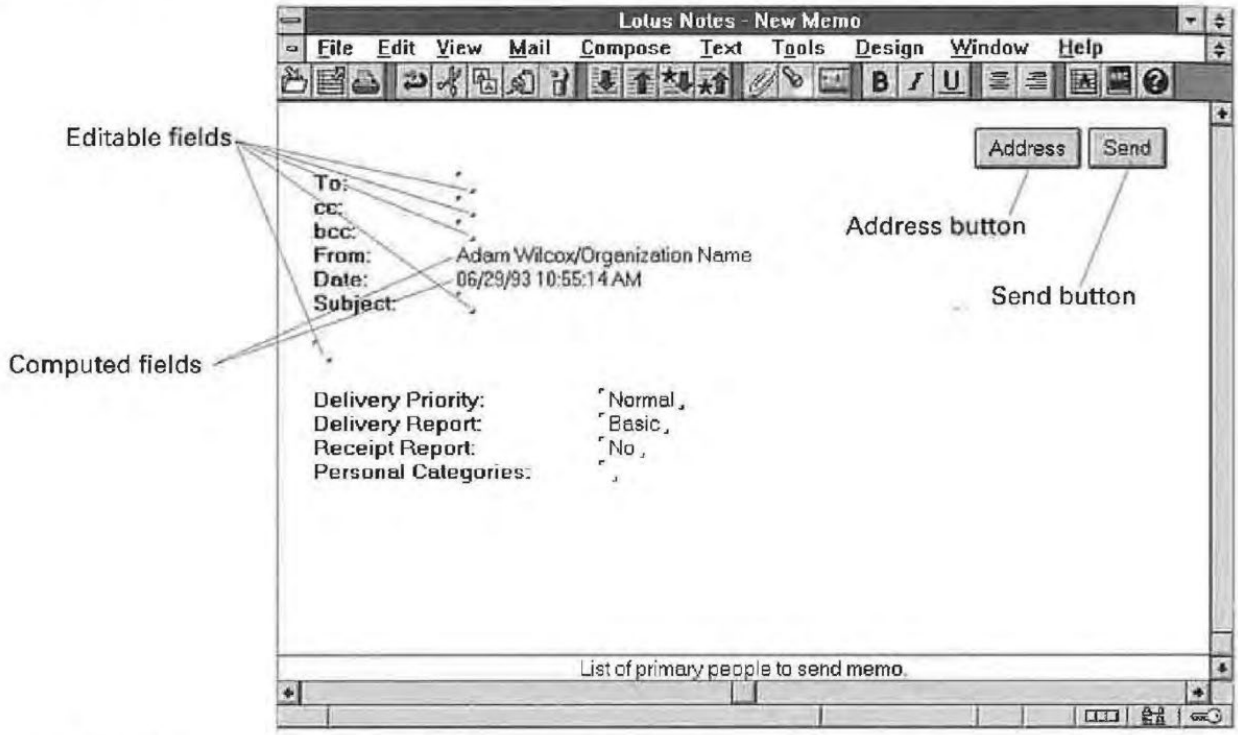
> 14. The computing system of claim 1, wherein the first search field corresponds to a single field of the files and matching files are identified based only on content of the single field of the files.
>
> 15. The computing system of claim 14, wherein the single field comprises one of: to, from, cc, bcc, subject, folder name, date, size, attachments, or body fields of email files.

115. The Lotus Notes+Raskin+Wu combination renders these claims obvious at least because as explained above for claim 4 (*supra* Section VIII.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.



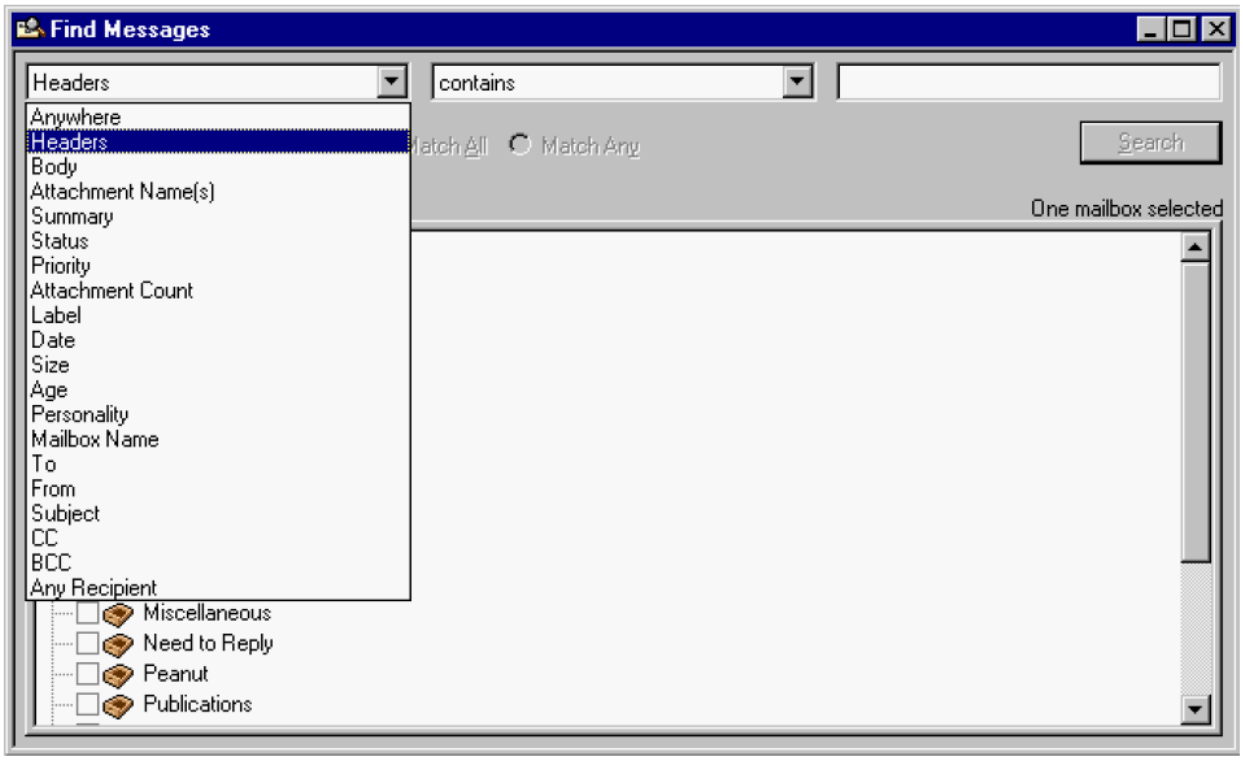
EX1059, 102.

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

EX1059, 168.

116. 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 VI.B.5.

Find Messages drop-down and conjunction options lists



EX1007, 129.

11. <u>Claim 16</u>

> 16. The computing system of claim 14, wherein the single field comprises one of: name, path, size, item type, date modified, date created, or content fields of document files.

117. 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**

EX1059, 168.

12. <u>Claim 17</u>

> 17. The computing system of claim 14, wherein the single field comprises one of: name, email address, phone number, company, job title, or content fields of contacts files.

118. 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**

EX1059, 168.

13. <u>Claim 18</u>

> 18. The computing system of claim 14, wherein the single field comprises one of: subject, start date/time, duration, recipients, location, recurrence, attendees, attachments, or content fields of calendar files.

119. The Lotus Notes+Raskin+Wu combination renders this claim obvious at least because the Lotus Notes references disclose that documents 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. The Lotus Notes references also 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**

EX1059, 168.

14. <u>Claim 19</u>

19. The computing system of claim 1, wherein the first partial search string applies to all textual content of the files and the second partial search string applies to only a single field of the files.

120. 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 (EX1059, 167-168, EX1061, 141-144).

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

EX1059, 168. Thus, a POSITA would have understood or at least found it obvious that if a user entered a first string by itself (without a preceding command) and then entered a second string with a command to search for that string within only a specific field, the Lotus Notes search tool would locate documents containing the first string anywhere in the document and the second string in only the field specified by the user. Continuing with the above example provided by Lotus Notes, a POSITA would have understood or at least found it obvious that a search query of "work AND [[Body] CONTAINS (Full-time)]" would return documents containing "work" anywhere in the document and "Full-time" in the body portion.

15. <u>Claim 20</u>

> 20. The computing system of claim 1, wherein modifiers can be applied to either or both of the first partial search string and second partial search string to include one or more operators selected from: AND, OR, NOT, NEAR, date ranges, and size ranges.

121. 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.

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

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

123. 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 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 VI.B.2-VI.B.3), however, and True expressly discloses (1) and Baeza-Yates expressly discloses (2). Thus, claims 1-20 are obvious over the Entourage reference in view of knowledge of a POSITA.

124. In sum, the patent's alleged improvement—incremental searching—was not an improvement, but rather a very well-known concept that dates back to at least the 1970s. *Supra* Section VI.B.1. The Entourage references expressly disclose this concept and disclose using it to specifically search emails, as recited in some of the dependent claims. The independent claims' recitation of the other well-known features disclosed by True and Baeza-Yates do not make these

otherwise obvious claims nonobvious. Just the opposite, it confirms that these claims are nothing more than the obvious inclusion of well-known features in a then-existing incremental search tool.

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

125. 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.

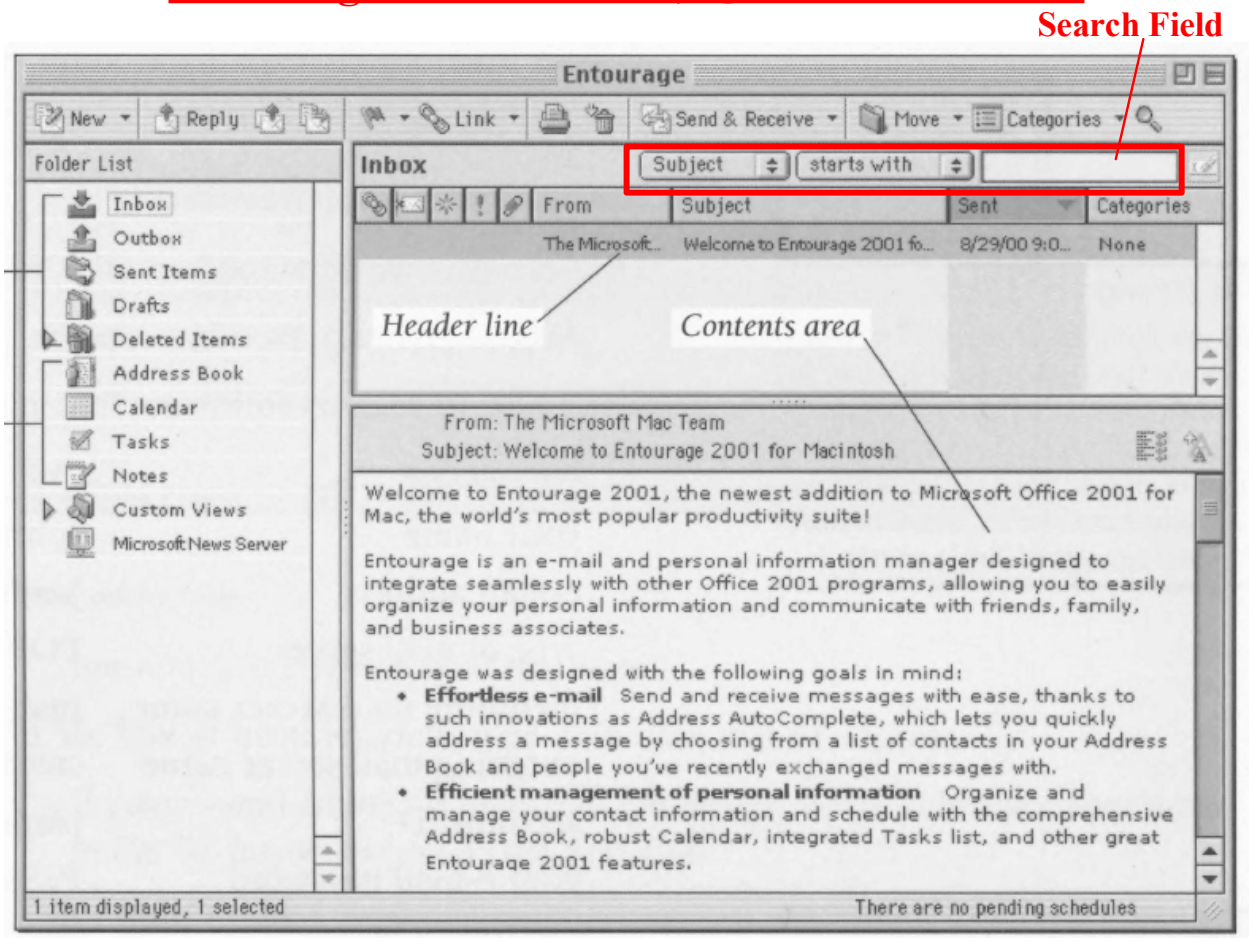
126. The Entourage references are analogous art to the '490 patent because they are from the same field of endeavor: systems and methods for providing a user interface through which a user can enter search strings, and locating files (e.g., emails) containing the search strings. *E.g.*, EX1001, 1:38-39,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

127. 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. Entourage provides several ways for users to search for files (e.g., emails, news articles, tasks, contacts) containing specific text: (1) an incremental, quick-find

search; (2) a "Find" search; and (3) and an "Advanced Find" search. EX1004, 8-14, 38-39, 89-90, 136, 161-164, 204-205, 209; EX1005, 173; EX1006, 384-387. A user can access the incremental, quick-find search from the main window of a folder (e.g., the main window of an email Inbox folder). EX1004, 8-9, 39, 89, 161, 204-205, 209; EX1005, 173; EX1006, 358, 384.

128. The incremental, quick-find search allows a user to search within specific email headers (To, From, Subject, Category) and displays results incrementally, "[a]s you type the filter text." EX1004, 9; *see also id*., 8, 161; EX1005, 173; EX1006, 358, 384. 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.

Entourage's Incremental, Quick-Find Search



EX1005, 173 (red annotations added).

Entourage's Incremental, Quick-Find Search

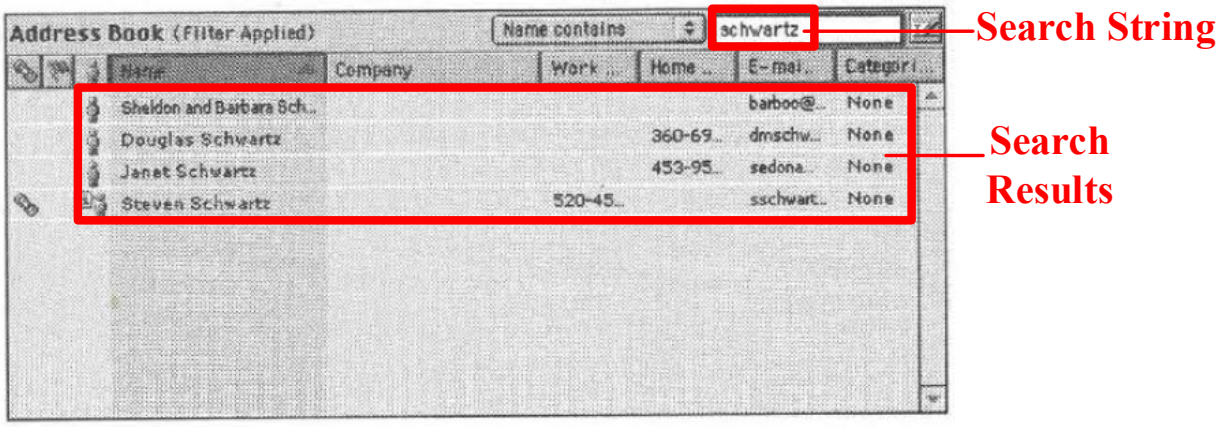


Figure 1.12 As you type the filter text, only matches are shown in the list.

EX1004, 9 (red annotations added); *see also id.*, 38.

129. The Find and Advanced Find searches, on the other hand, are delimited searches—they require the user to select a "Find" button before they begin displaying results. EX1004, 10-12, 162-163; EX1006, 384-387. The Find search can be initiated by, e.g., pressing command+F, which launches the search interface—a Find dialog box. EX1004, 161; EX1006, 384. The Advanced Find search can be initiated by, e.g., pressing the "More Options" button in the Find dialog box or pressing option+command+F. EX1004, 9, 11, 162; EX1006, 385.

Entourage's Find Search

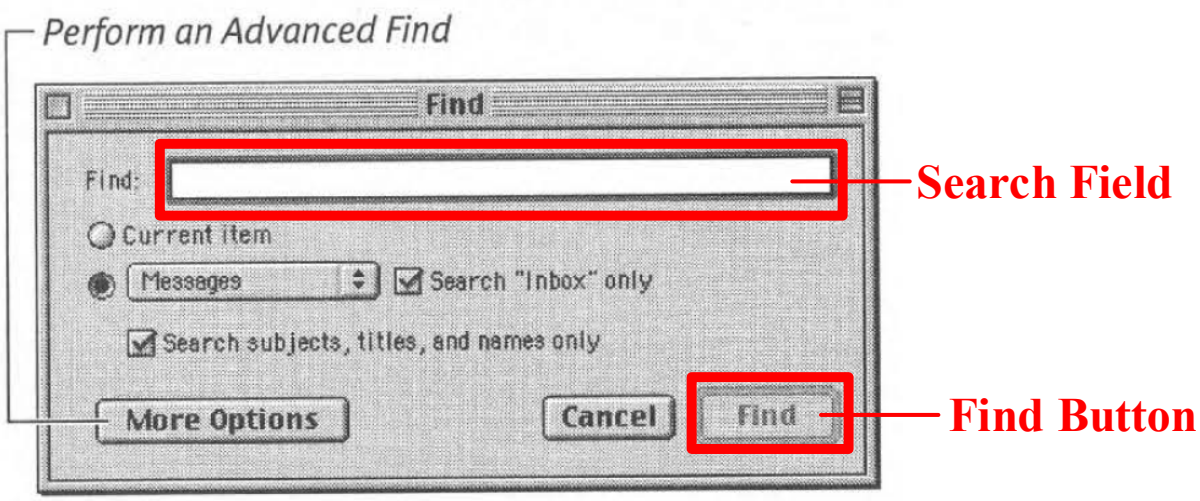


Figure 1.13 You can conduct simple searches by specifying criteria in the Find dialog box.

EX1004, 9 (red annotations added).

Entourage's Advanced Find Search

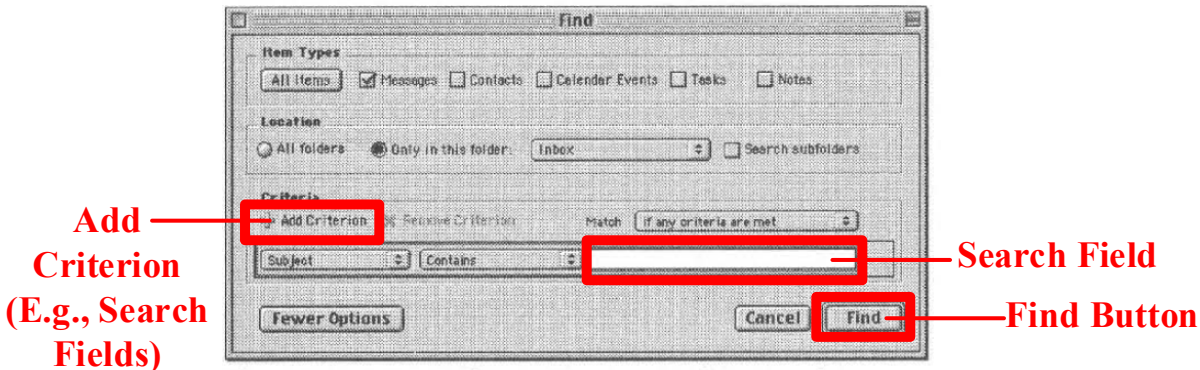


Figure 11.18 To perform a more complex search (such as one that uses multiple criteria), choose the Advanced Find command. You can find only unread messages, messages that contain attachments, or messages that have been identified as junk email, for example.

EX1004, 162 (red annotations added).

130. The Find search can search the body portion of emails (in addition to the headers as in the incremental, quick-find search) (EX1004, 9-10, 161-162; EX1006, 384-385) and the Advanced Find search can search across multiple headers, item types, and/or folders (EX1004, 11-12, 162-163; EX1006, 385-387) and allows a user to filter emails based on whether or not they have an attachment (EX1004, 162; EX1006, 386-387).

131. 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.

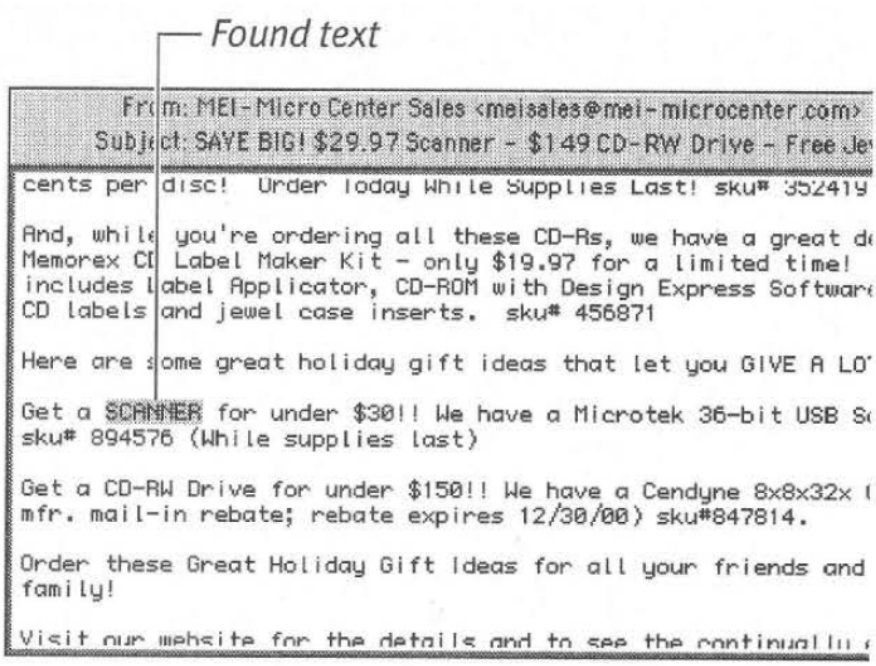


Figure 11.20 The first instance of the found text—if there is one—is highlighted for you.

EX1004, 163.

B. Combining The Entourage References

132. 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. 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). Thus, a POSITA would have been motivated to combine the teachings of the three Entourage references to gain a more complete understanding of how the Entourage system works.

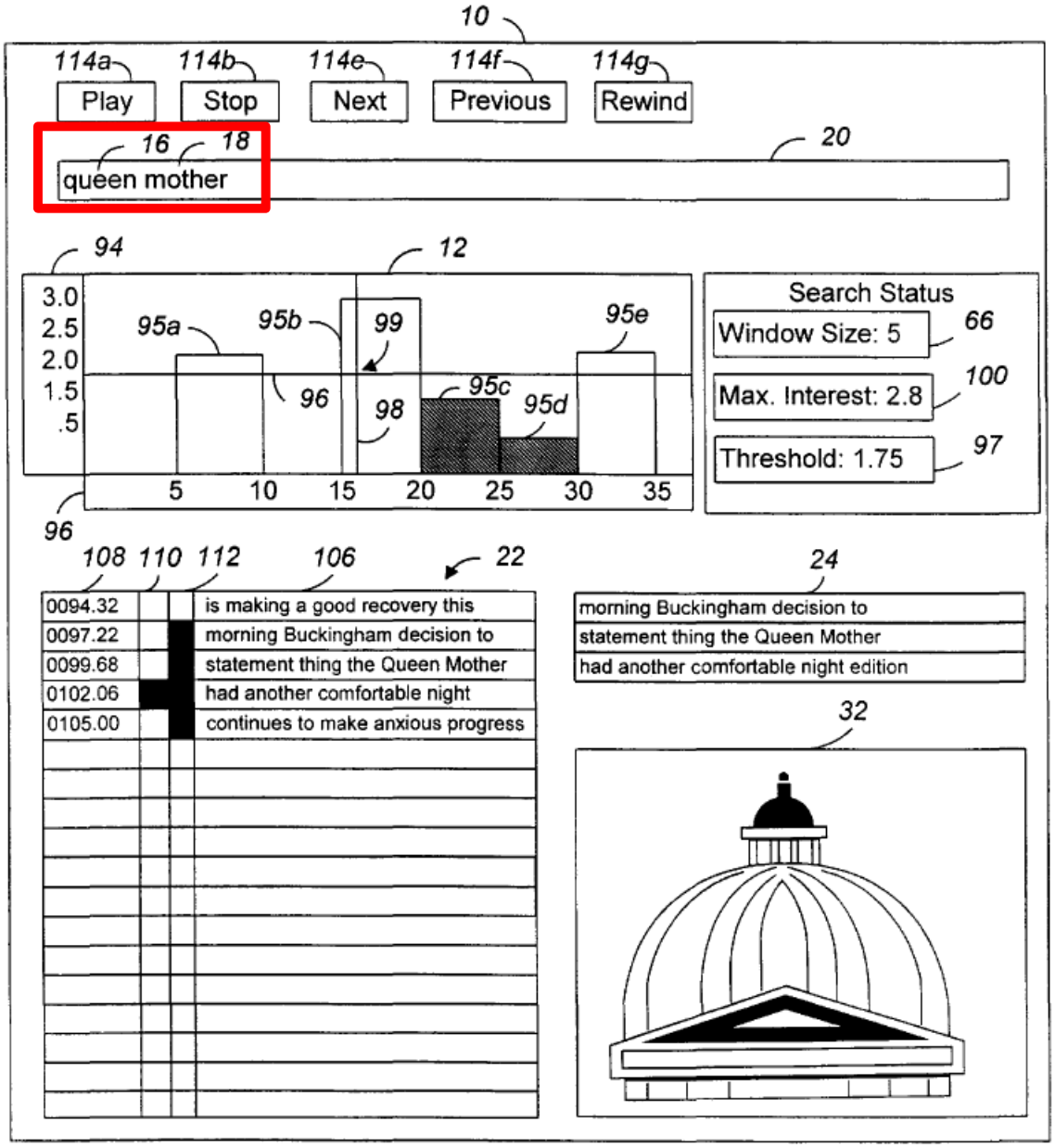
133. A POSITA also would have reasonably expected to succeed in combining the teachings of these three references at least because they describe the same version of Entourage (Entourage 2001) for the same computing system (Macintosh). *E.g.*, EX1004, (i); EX1005, (a)-(c), 171-188; EX1006, (i), 1-3, 5,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

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

134. 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, 11:38-39, 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.

135. True is an issued U.S. patent that discloses incrementally searching for, and locating, portions of an audio transcript containing multiple search strings entered in a single search field. EX1009, 2:26-31, 2:39-48, 4:38-43, 4:65-5:6, 5:41-50, FIG. 2. For example, True discloses that as a user enters the strings “Queen” and “mother” in a single search field, the search tool displays “query results … as the query is being input.” EX1009, 2:41-42; *see also id.*, 2:26-48, 4:66-5:6. A POSITA would have understood that True’s search tool separately searches each search string at least because the search tool ranks the search results based on the number of occurrences of each search string in the audio transcript, regardless of their position in the audio transcript. EX1009, 5:41-50.

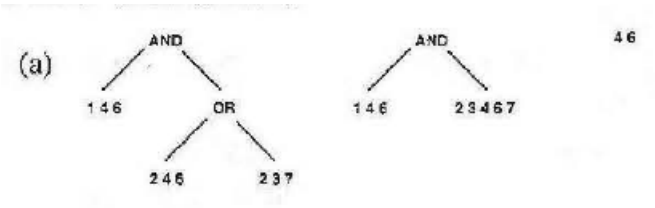
FIG. 2



EX1009, FIG. 2 (red annotation added).

136. 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 VI.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).

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

137. 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 (i.e., searching for a first set of files containing the first search string and searching for a second set of files containing the second search string) in an index data structure, and then intersecting the results (i.e., identifying the files that appear in both sets of results).

138. A POSITA would have been motivated to combine the Entourage references' teachings to incrementally search at least one search string entered in a single search field (*e.g.*, EX1004, 8-10, 38-39, 89, 161, 204-205, 209, 267; EX1005, 173; EX1006, 358, 384, 387; *infra* Section IX.E.1.c) with True's express teachings to do so for multiple search strings entered in the search field (EX1009, 2:26-31, 2:39-48, 4:38-43, 4:65-5:6, 5:41-50, FIG. 2; *infra* Sections IX.E.1.b-IX.E.1.c) at least because a POSITA knew that such a multi-string search would return more relevant results than a single-string query. *Infra* Sections IX.E.1.b-IX.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.”).

139. 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.

140. 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.

141. 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 VI.B.3).

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

1. Claim 1

a) Element [1.1] [1]

[1.1] A computing system comprising:

142. 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.

b) Elements [1.2-1.4]

[1.2] one or more hardware computer processors;
[1.3] a display device viewable by a user of the computing system;
[1.4] a computer readable storage medium storing:

143. The Entourage references teach these claim elements: a processor (e.g., a chip), display device (e.g., a monitor), and computer readable storage

[1] Reference numbers in the format of [claim#.element#] are added throughout for ease of reference.

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.

144. For example, "[t]o use Microsoft Entourage 2001, [the] computer must have at least: [a] 120-MHz PowerPC chip … 160 MB of disk space … [m]onitor that can display 256 grays or colors and 640 x 480 pixels or more … [and] 32 MB of memory" for Mac OS 8.1 through 8.6 "[or] 48 MB of memory" for Mac OS 9 and later." EX1006, 614.

145. 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.

c) <u>Element [1.5]</u>

[1.5] an index data structure comprising associations between each of a plurality of character strings to associated files of a plurality of files in which the character strings are included; and

146. 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 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, as taught by Baeza-Yates and was well known in the art. *Supra* Section VI.B.3.

147. "[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.

Document Index 201

202 Keyword	203 Document Identifier, 204 (Occurrence)
Areas	(B, 12)
Cases	(A, 6)
Congress	(B, 1)
Court	(A, 10)
Courts	(A, 3, 16) (B, 6)
Created	(B, 3)
Disputes	(B, 9)
District	(A, 15)
Federal	(A, 9, 14) (B, 5)
Handle	(B, 8)
Law	(B, 14)
Most	(A, 5)
Particular	(B, 11)
Specialized	(B, 4)
System	(A, 11)
Trial	(A, 2)

205 Tuple

206

Two occurences of the word "Courts," in document A, one at position 3, another at position 16

Figure 2

EX1054, FIG. 2 (red annotations added); *see also id.*, 2:10-12.

148. Thus, rather than scanning every word of every file sequentially, which a POSITA knew takes more time and consumes more computing resources, a POSITA would have understood or at least found it obvious that the computing system indexes the computing device's files so that the search process (e.g.,

Entourage's search process) is carried out more quickly and efficiently, as was well known in the art. *E.g.*, EX1023, 6, 191. Specifically, a POSITA would have been 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.

149. 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.

d) <u>Element [1.6]</u>

[1.6] software instructions executable by the one or more hardware computer processors causing the computing system to:

150. 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 ("Drag the Office folder from the Office CD-ROM onto your hard drive."), 614-618; *see also* EX1043, 573-576. A POSITA would have understood or at least found it obvious

that the software includes software instructions that are 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.

e) <u>Element [1.7]</u>

[1.7] cause display of an interactive user interface on the display device, the interactive user interface including at least a first search field;

151. 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.

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

Searching Emails

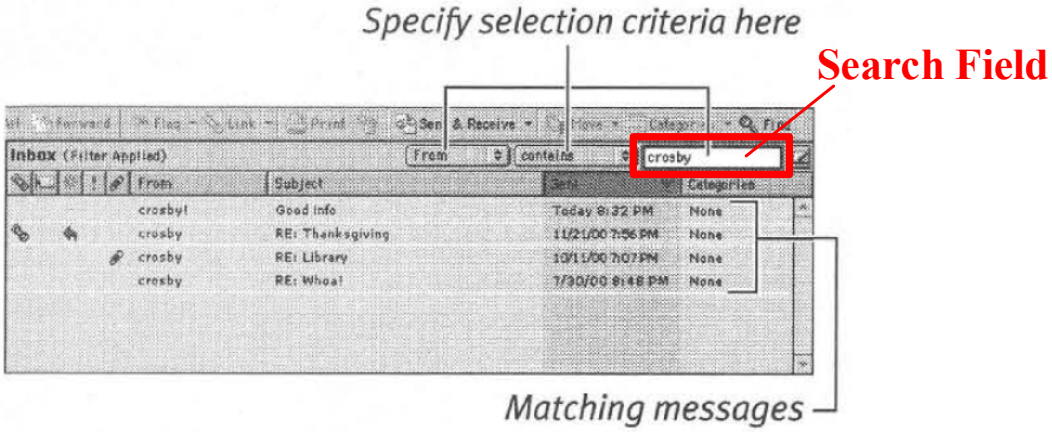


Figure 11.15 To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161 (red annotations added).

Searching Contacts

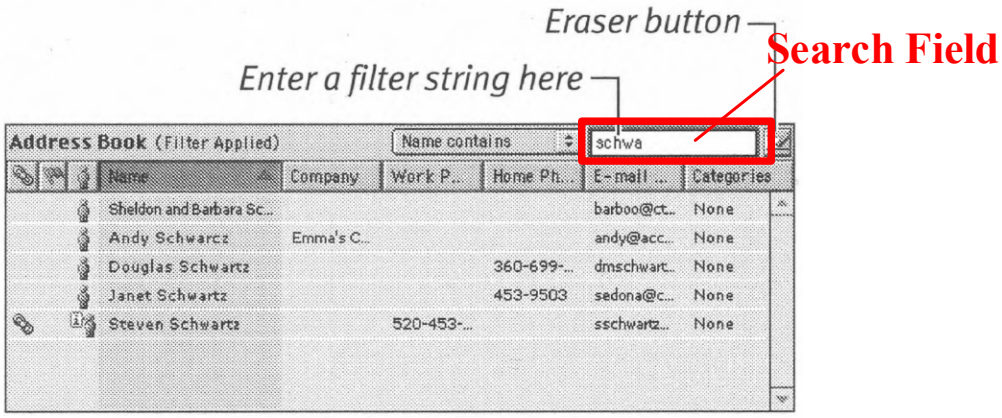


Figure 3.4 You can limit the display to records that contain a particular text string.

EX1004, 38 (red annotations added).

f) Elements [1.8-1.10]

[1.8] receive a first partial search string in the first search field;

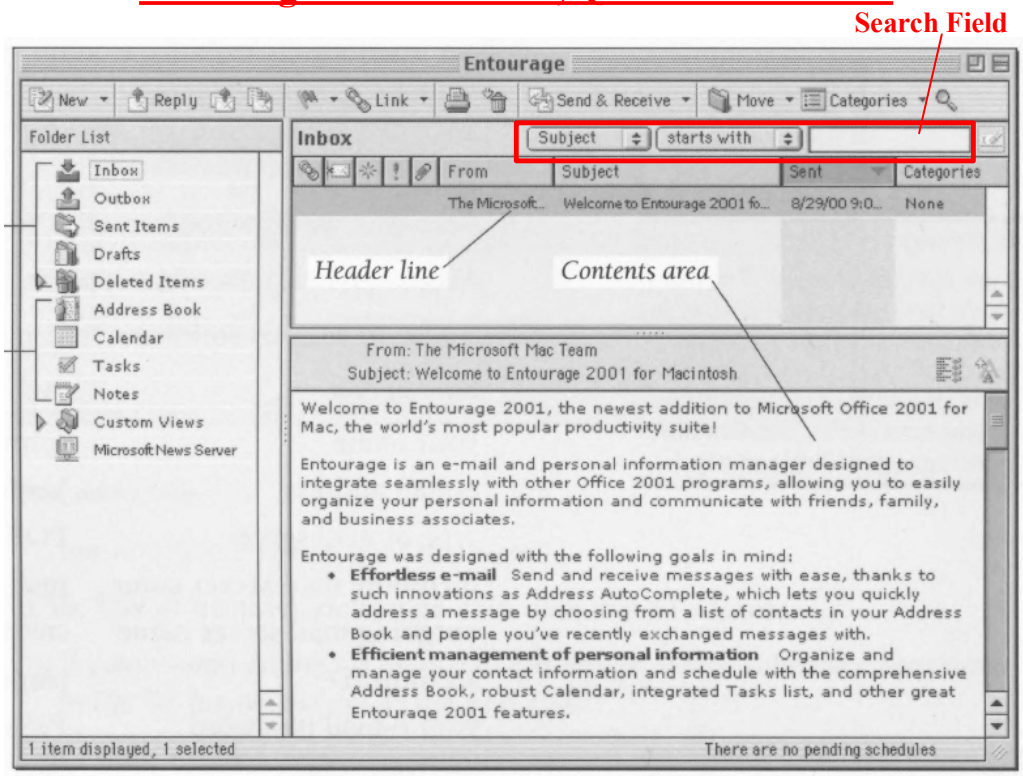
[1.9] receive a second partial search string in the first search field;

[1.10] wherein the first and second partial search strings are separated by a string separator character in the first search field;

153. 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 VI.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 VI.B.2.

154. 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 suggest receiving at least two strings (e.g., a person's first and last names, such as "Orson Welles," etc.) (*e.g.*, 387) in the same search field separated by a string separator character (e.g., a space) (EX1006, 387). *See also* EX1004, 267 ("searching for text string**s**," plural) (emphasis added); EX1043, 327, 338, 343-344, 379-380, 385-389, 402-409. The Entourage references also expressly disclose "searching for text string**s**" (plural), at least suggesting to a POSITA to receive multiple search 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).

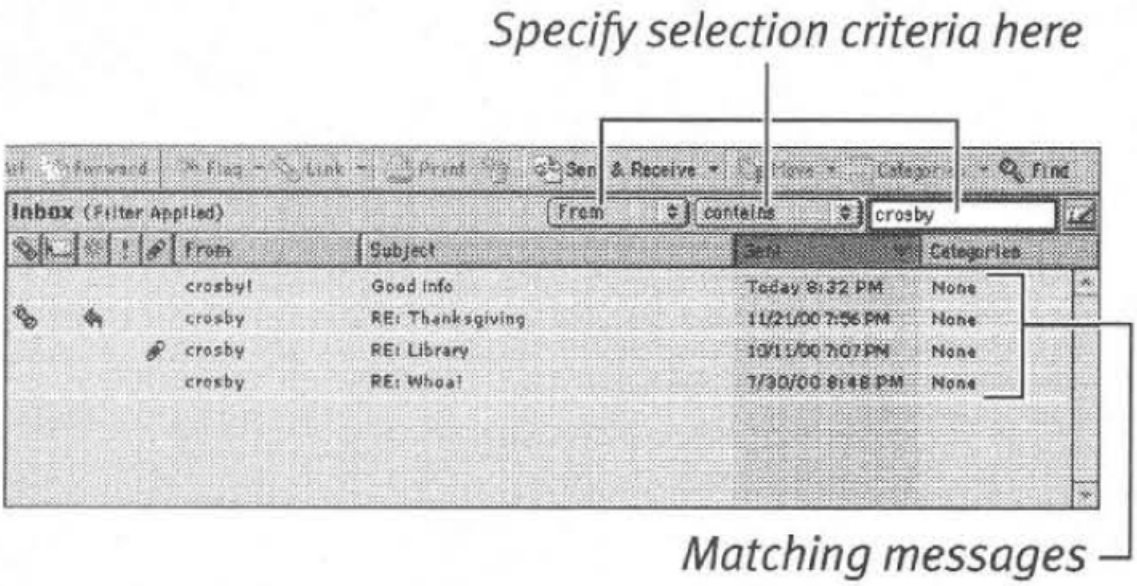


Figure 11.15 To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161.

155. 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 VI.B.2. For example, a POSITA knew, based on their own knowledge and common sense, that a multi-string search such as "Orson Welles" returns more specific and thus more relevant results than searching for just "Orson" or just "Welles" alone. 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.

156. 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.' Using Boolean operators in search queries is conventional and will not be discussed in great detail here for brevity's sake."); 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. A POSITA would have understood that a user can perform an exact search for a phrase (e.g., Orson Welles) by placing quotation marks around the phrase, as was well known in the art, to return only files containing the exact phrase. EX1023, 287; EX1035, 26:64-27:13; EX1057, 19 ("[Y]ou[] need to enclose the words in double quotes to let them know you're doing a *phrase search*."), 89, 109, 129, 151, 166; EX1059, 162 ("To search for multiple-word phrases, you should enclose them in quotation marks."); EX1061, 140 ("[T]ype **'gold and silver'** to find documents that contain the phrase *gold and silver*.").

157. Moreover, it was well known to use various operators, 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; 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 ("In the text box on the Search bar, enter the text for which you wish to search."), 162 ("You can use the *logical operators* AND … to find documents that contain various combinations of words and text strings."), 163 ("[T]o search for documents that contain all of several words, type them all in this box, separated by commas.").

158. Thus, even if the Entourage references do not expressly disclose receiving two search strings in the same search field (which I do 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.

g) <u>Elements [1.11-1.13]</u>

[1.11] searching the index data structure for a first one or more of the plurality of files each including at least one string that corresponds to the first partial search string; [1.12] searching the index data structure for a second one or more of the plurality of files each including at least one string that corresponds to the second partial search string; and [1.13] updating a set of matching files of the plurality of files as each successive character of the first partial search string and the second partial search string is received respectively in the first search field by identifying any of the plurality

of files included in both the first one or more of the plurality of files and the second one or more of the plurality of files, wherein the first partial search string and the second partial search string are non-adjacent in at least one of the matching files.

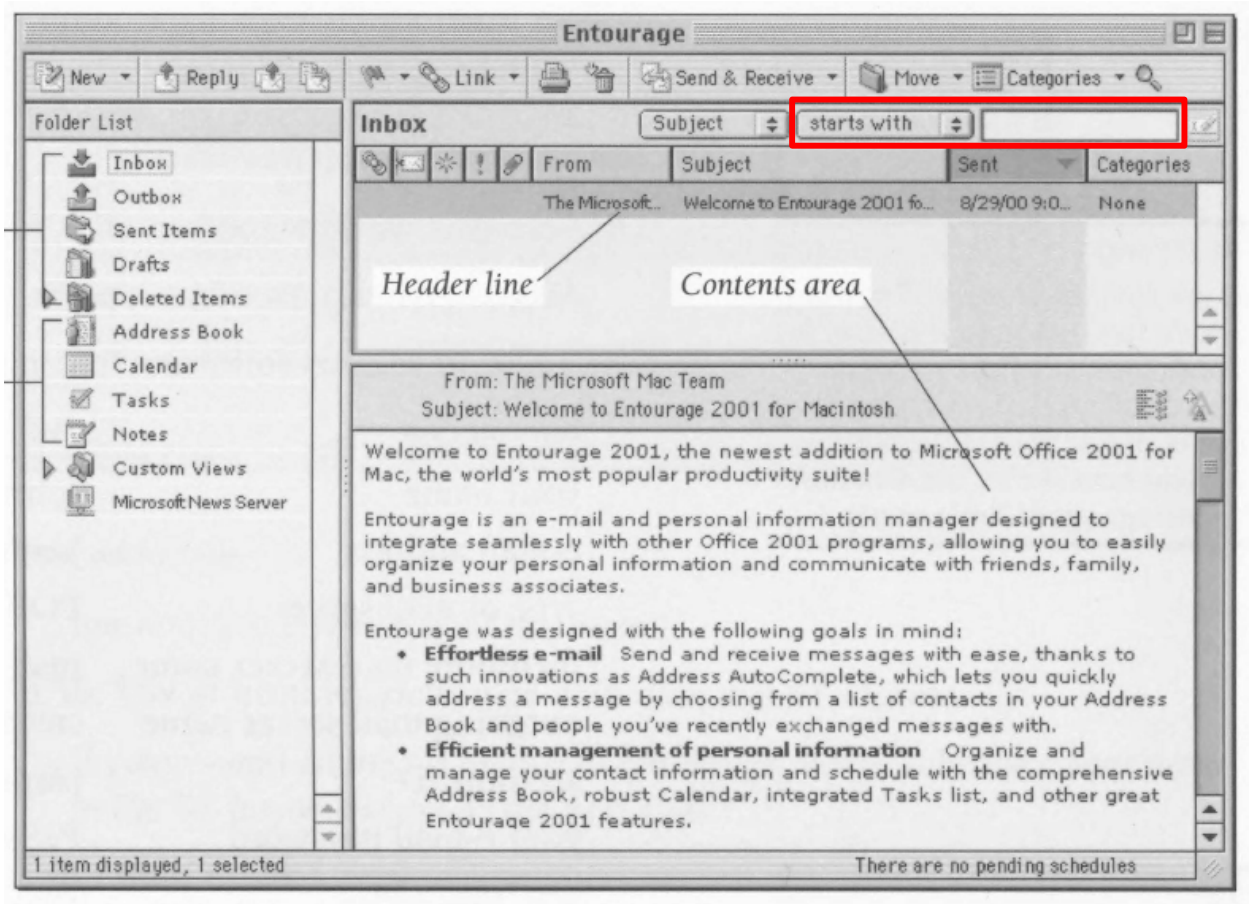
159. The Entourage references in combination with True and Baeza-Yates teach these claim elements, especially in view of the knowledge of a POSITA reflected in the numerous background references described above in Sections VI.B.1-VI.B.3. EX1004, 2-3, 7-14, 38-39, 89-90, 161-164, 204-205, 209; EX1005, 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 VI.B.1-VI.B.3; *see also* EX1043, 83-84, 337-338, 343-344, 355, 385-386, 404.

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

160. The Entourage references teach what was generally well known in the art: incrementally locating and displaying files[2] (e.g., contacts, emails, news articles, tasks, etc.) containing partial search string(s), i.e., locating and displaying matching files 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,

[2] 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.

204-205, 209; EX1005, 173; EX1006, 358, 384; *supra* Section VI.B.1; *see also* EX1043, 83-84, 337-338, 343-344, 355, 385-386, 404.



EX1005, 173 (red annotation added).

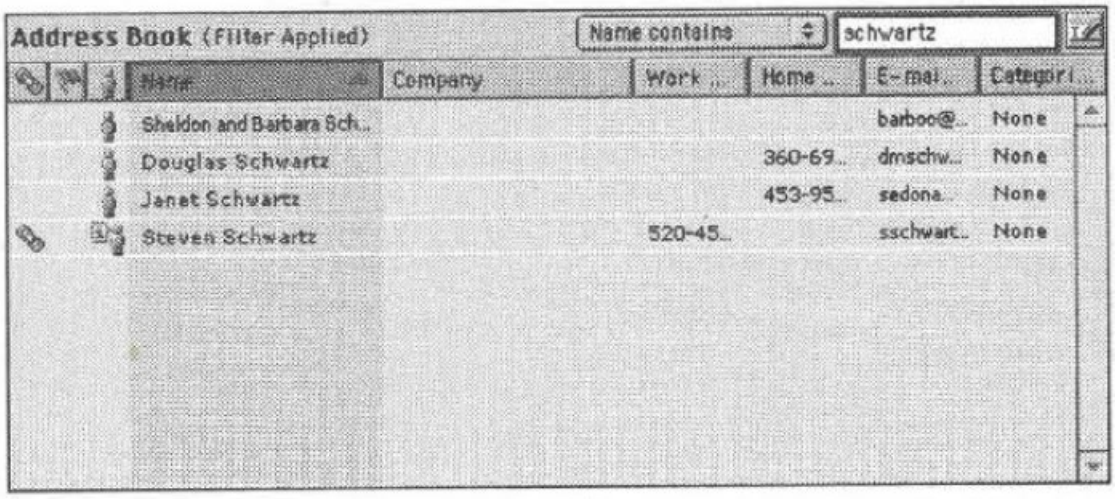


Figure 1.12 As you type the filter text, only matches are shown in the list.

EX1004, 9.

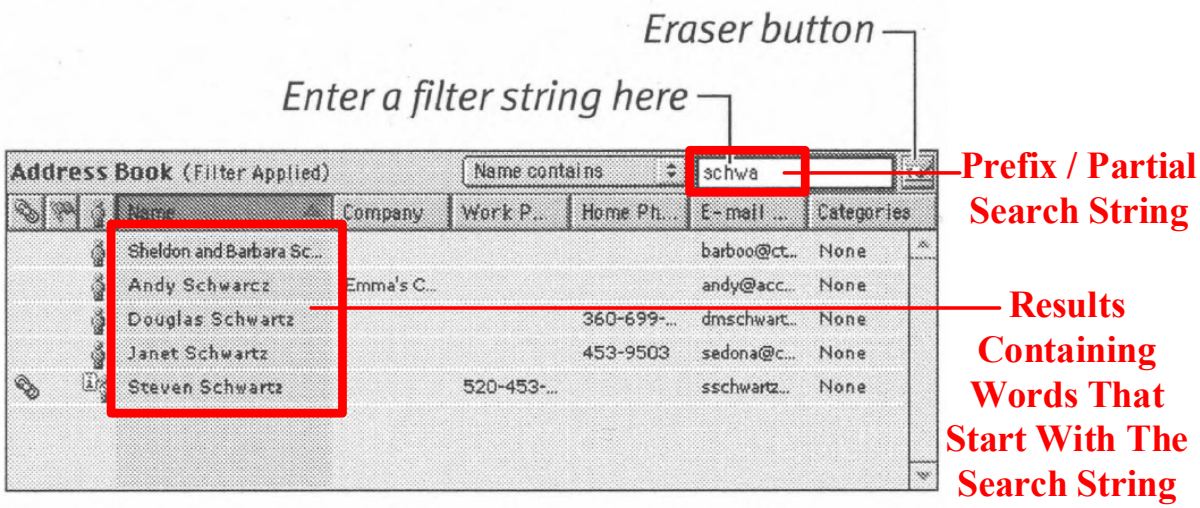


Figure 3.4 You can limit the display to records that contain a particular text string.

EX1004, 38 (red annotations added).

Searching For Multiple Strings

161. 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 VI.B.2 (state of the art) and IX.E.1.f (elements [1.8-1.10]).

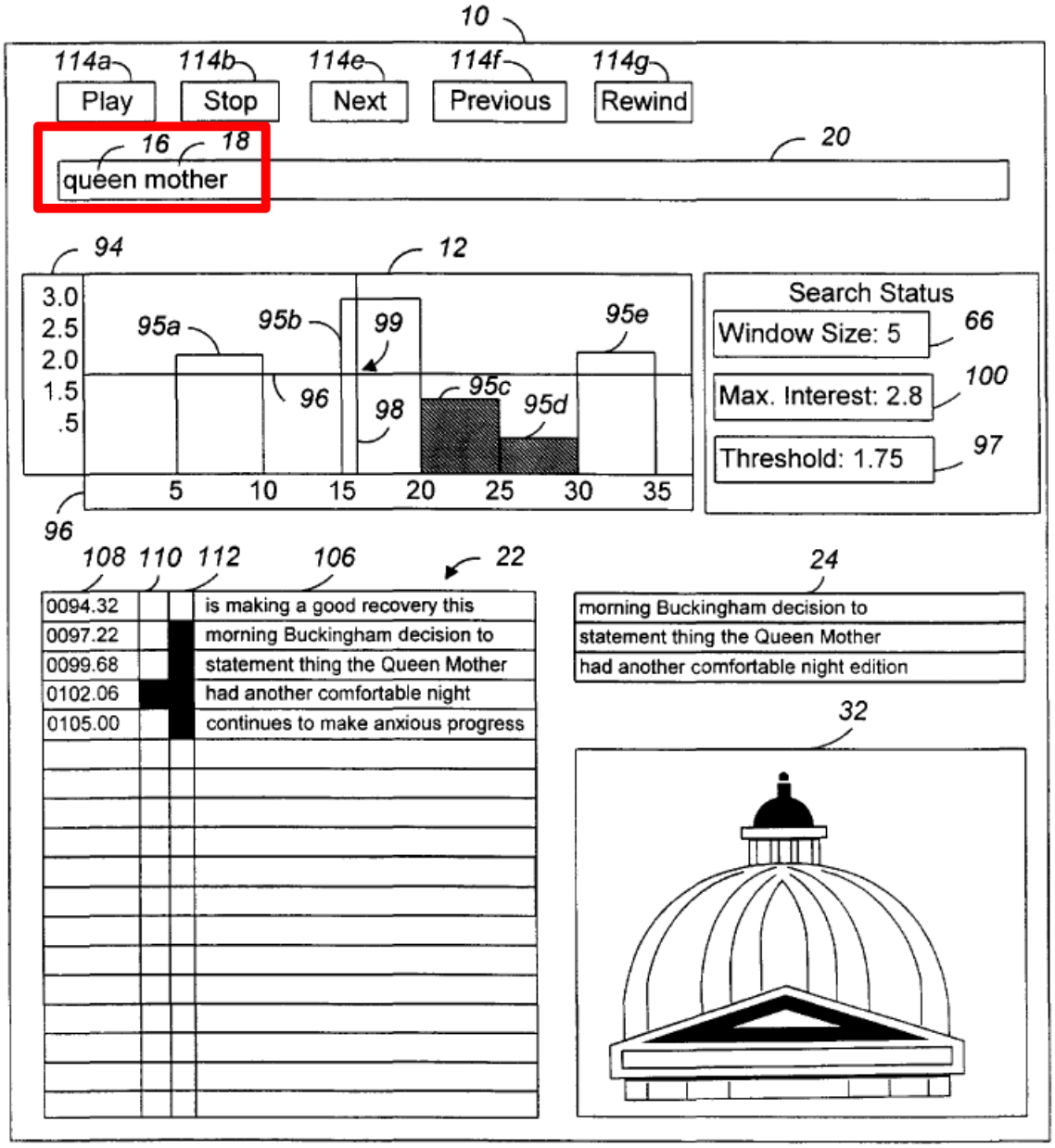
162. Moreover a POSITA would have been motivated to incorporate True's teachings to incrementally locate (i.e., locate as the user types or modifies their search query) files containing multiple search strings entered in the same search field (e.g., files containing the strings "queen" and "mother") 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 VI.B.2, IX.E.1.b.

163. 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.

164. 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).

165. 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 partial 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. A POSITA would have understood that True's search tool separately searches each search string at least because the search tool ranks the search results based on the number of occurrences of each search string in the audio transcript, regardless of their position in the audio transcript. EX1009, 5:41-50.

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

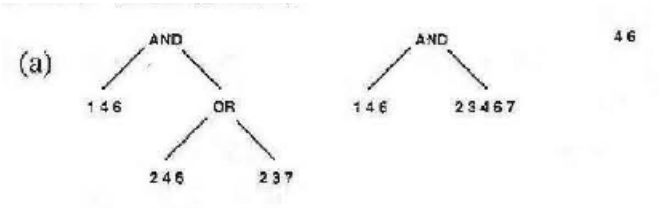
greatly reducing the number of search results you have to consider."); *see also* EX1021, 127 ("Even building a Boolean search pattern is made more effective when the early results of the pattern appear as the user adds to the pattern's specificity").

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

167. 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* Sections VI.B.2-VI.B.3.

168. 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.

169. 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 (“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.”); 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.

170. Thus, it would have been obvious to a POSITA to identify the files containing both 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 a POSITA naturally would have utilized. In particular, a POSITA would have recognized that there were only 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.

171. 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. That is, a POSITA would have understood that the intersection approach's separate treatment of the search strings would provide quick results for such an incremental search because it would be able to efficiently update search results in response to modifications to either search string. For example, if a user were to modify only one string, the sequential search approach would have to completely rerun the search because it searches for both strings simultaneously, whereas the intersection approach could preserve previous results for the unmodified search string, and would only have to update results for the modified search string.

172. Thus, it would have been obvious to a POSITA to identify the files containing both 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 a classical technique that a POSITA naturally would have utilized.

Using An Index For The Search

173. 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 VI.B.3, IX.E.1.c. Moreover, it would be nonsensical to create the index data structure but not use it to expedite the search.

Including Files In Which The Search Strings Are Non-Adjacent

174. 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. The alternative—treating the multi-word query as a single string (e.g., "Queen Mother")— performs a single search for the exact phrase and yields only files containing that exact phrase (with the two words of the phrase directly adjacent to one another).

175. Moreover, in the search results, 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. *Supra* Section VI.B.2. For example, by at least 1990, then-existing search techniques "add[ed] variables indicating word order and proximity" such that "given [a search query of] 'patent AND chemical WITHIN 3 WORDS'

the technique locates 'patent on a chemical'." EX1017, 1:66-2:2; *see also*, *e.g.*, EX1023, 101-102; EX1059, 160. Further, by their very nature, classical Boolean AND queries locate files in which the search strings are non-adjacent at least because these queries care only about whether the files contain the search strings, not whether the search strings are adjacent or non-adjacent in the files. *E.g.*, EX1023, 24-27, 100-103, 192-196.

2. Claims 2-3

> 2. The computing system of claim 1, wherein the set of matching files includes a first file wherein the first partial search string precedes the second partial search string.
>
> 3. The computing system of claim 2, wherein the set of matching files includes a second file wherein the second partial search string precedes the first partial search string.

176. 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 VI.B.2, IX.E.1.g. For example, the well-known Boolean AND search cares only about whether the search strings

appear at all in the files, not what order they appear in. *Supra* Sections VI.B.2, IX.E.1.g.

177. 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.

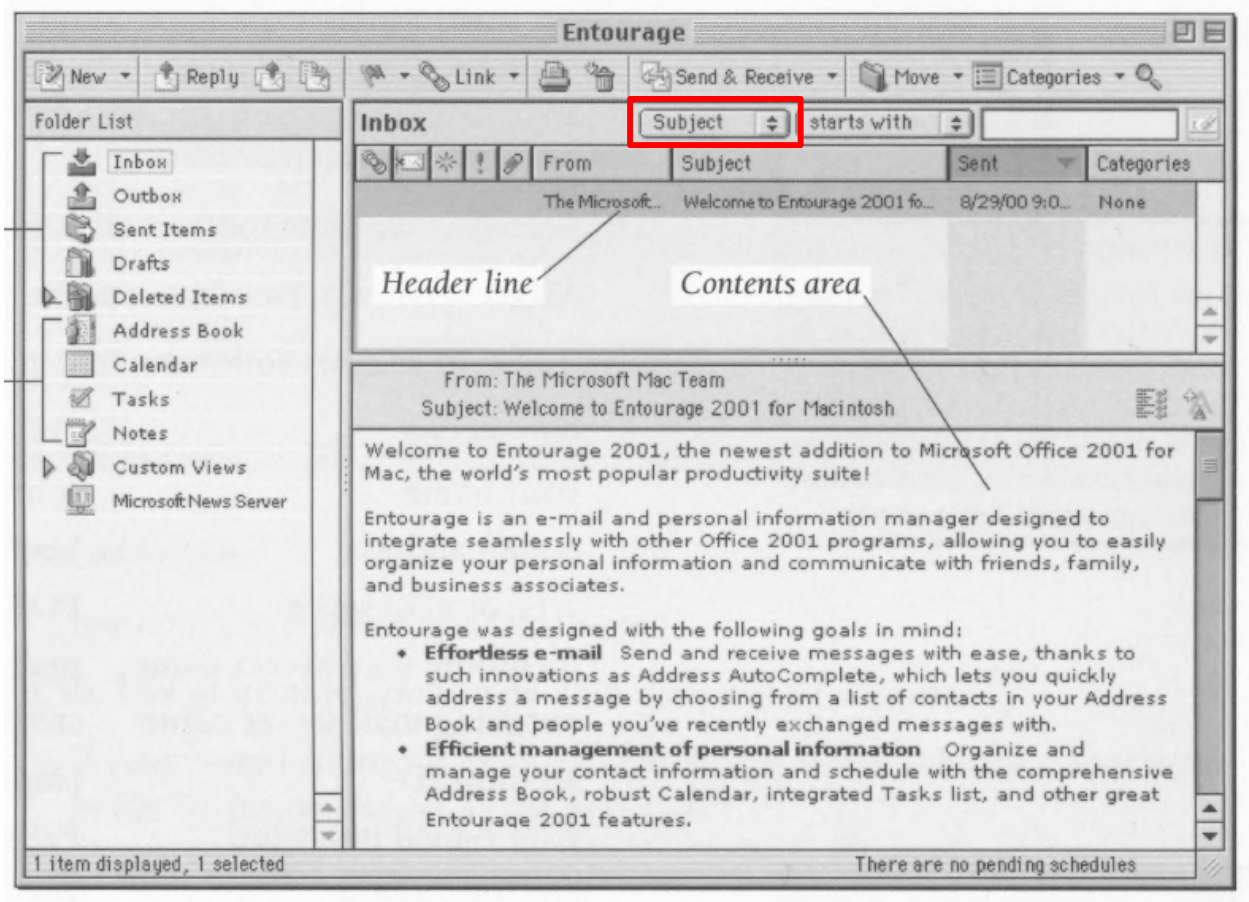
178. Thus, a POSITA would have understood or at least found it obvious that the computing system would locate some matching files in which the first search string preceded the second search string and other matching files in which the second search string preceded the first search string at least because the POSITA understood that the multi-string search does not require the search strings appear in any particular order in the searched files.

3. <u>Claim 4</u>

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

179. 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 IX.C.10 (claim 14).



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

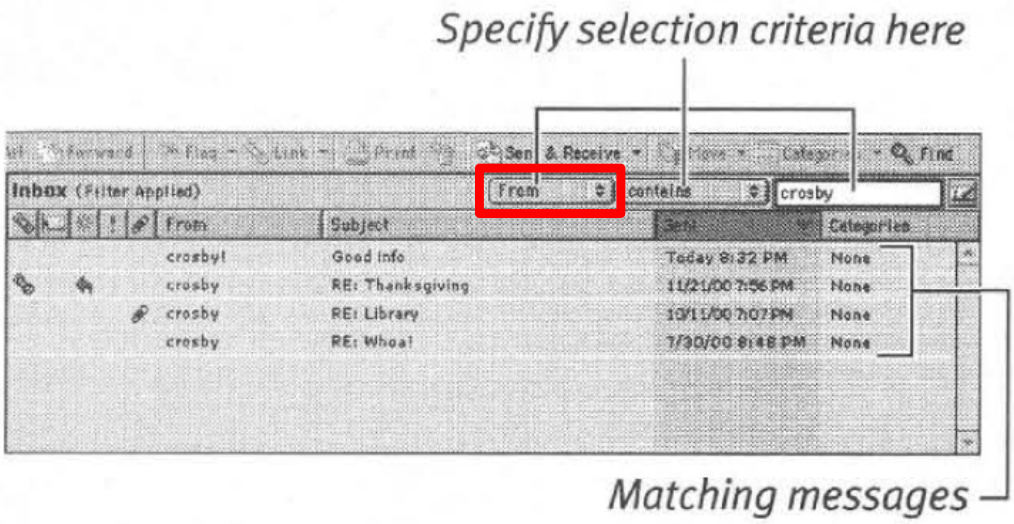
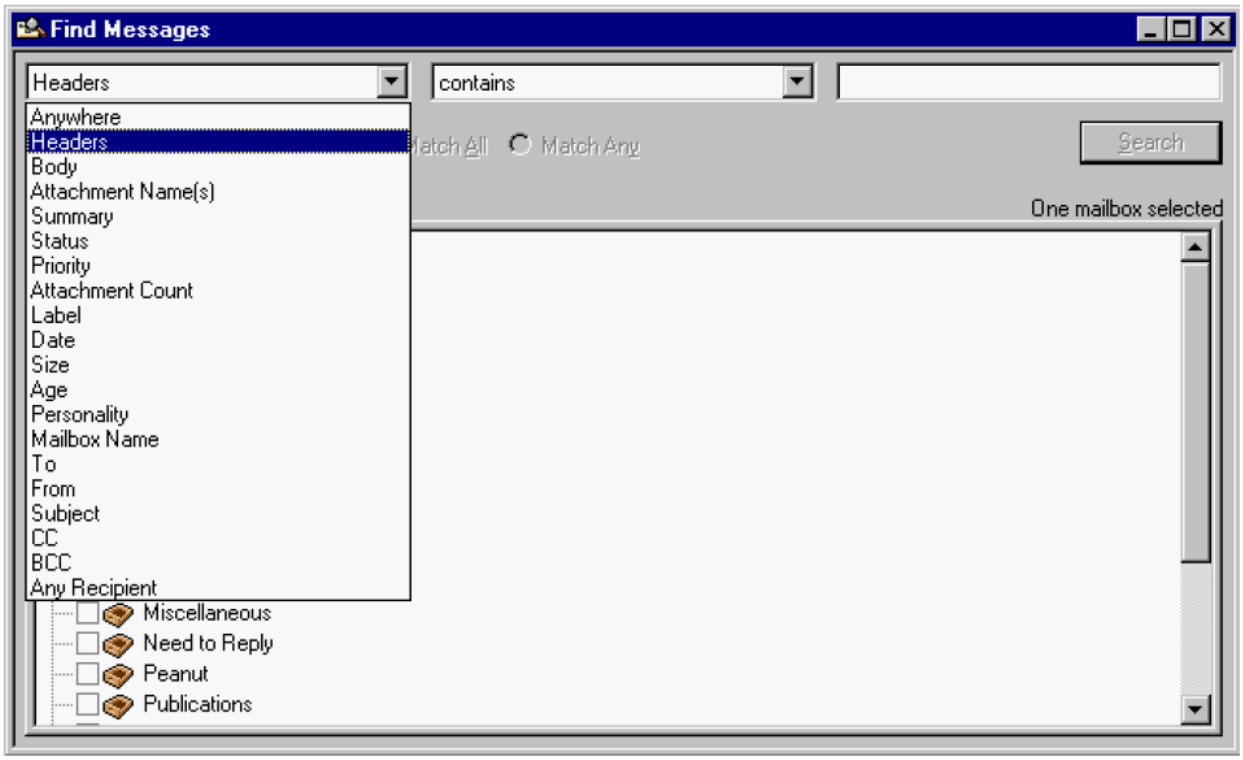


Figure 11.15 To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161 (red annotations added showing searching only within the "From" header).

180. Moreover, it was well known to perform this kind of attribute field-specific search. *Supra* Section VI.B.5. EX1007, 129-130; EX1008, 268-269, 471-478; EX1032, 5:51-56, 11:35-41, 11:64-12:11; EX1037, [0088]; EX1039, 100-105.

Find Messages drop-down and conjunction options lists



EX1007, 129.

181. 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.

4. <u>Claims 5-7</u>

> 5. The computing system of claim 1, wherein the interactive user interface includes at least:
>
> an email search mode interface having email-specific attribute search fields including at least a date field, a from field, and a sender field;
>
> a document search mode interface having document-specific attribute search fields including one or more of a file name field, a file type field, a date field, a file size field, and a path field; and

an email attachment search mode interface including one or more of a name field, a date field, a size field, and an extension field.

6. The computing system of claim 1, wherein the index data structure is generated and updated by an index module that generates at least an email index corresponding to a first plurality of the files that are emails, a document index corresponding to a second plurality of the files that are documents, and an email attachment index corresponding to a third plurality of files that are email attachment files.

7. The computing system of claim 1, wherein the index data structure includes an email index, a document index, and an email attachment index.

182. 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 IX.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).

183. 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.

184. 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.

185. For searching documents, for example, a POSITA would have reasonably expected success in copying and modifying the Entourage email search interface to allow for document-specific attribute fields to be specified (instead of email-specific attributes).

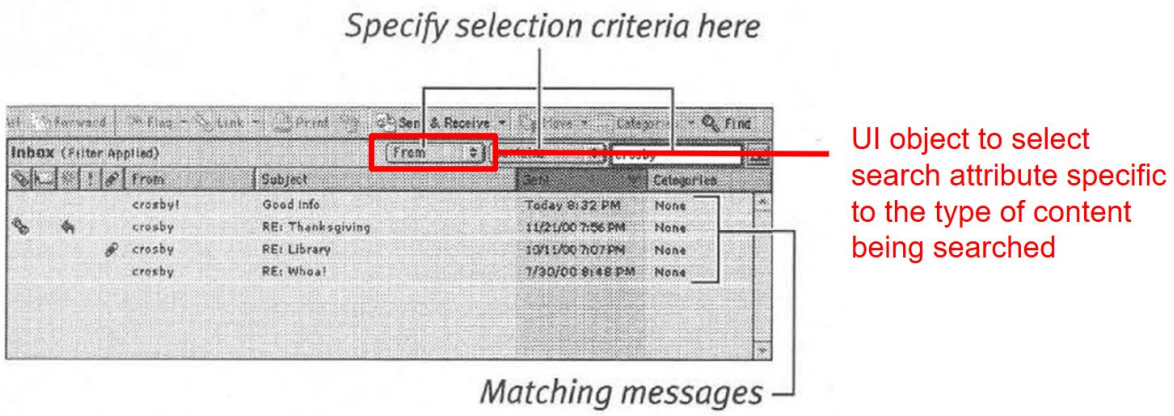


Figure 11.15 To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161 (red annotations added). To enable the document search, a POSITA would have reasonably expected success in using and/or modifying the existing

code for creating an email index to create an analogous index structure for document files. Further, a POSITA would have reasonably expected success in modifying the existing code for performing the email search to perform a search on document files (and their attributes). The same applies for email attachments. The code modifications needed to perform searches on these other types of files would have been trivial and well within the abilities of a POSITA. 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 (email searching), 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 (web searching); EX1024, 1-5.

5. <u>Claim 8</u>

8. The computing system of claim 1, wherein the first partial search string comprises characters corresponding to at least a portion of a first string being searched for and the second partial search string comprises characters corresponding to at least a portion of a second string being searched for.

186. 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 IX.E.1.f.

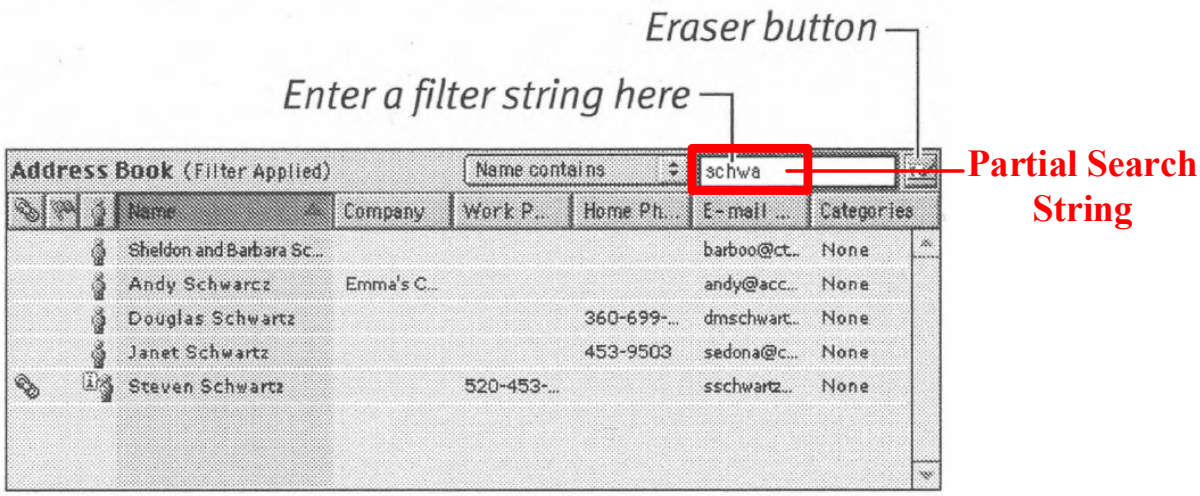


Figure 3.4 You can limit the display to records that contain a particular text string.

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 VI.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.

6. <u>Claim 9</u>

> 9. The computing system of claim 1, wherein the software instructions further cause the computing system to display in the interactive user interface indications of the set of matching files.

187. 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 IX.E.1.

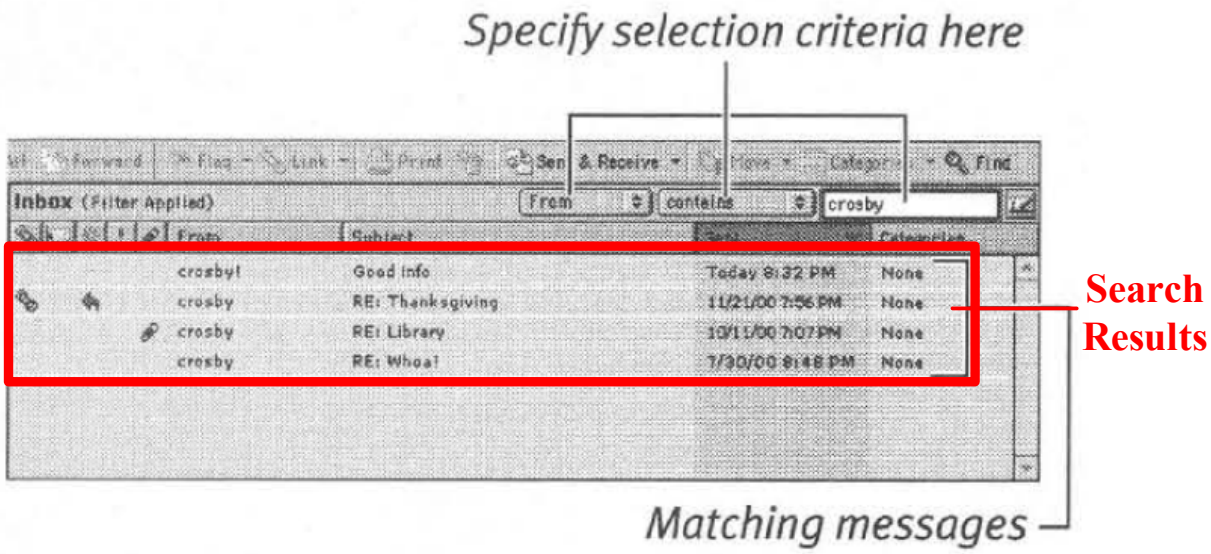


Figure 11.15 To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

EX1004, 161 (red annotations added).

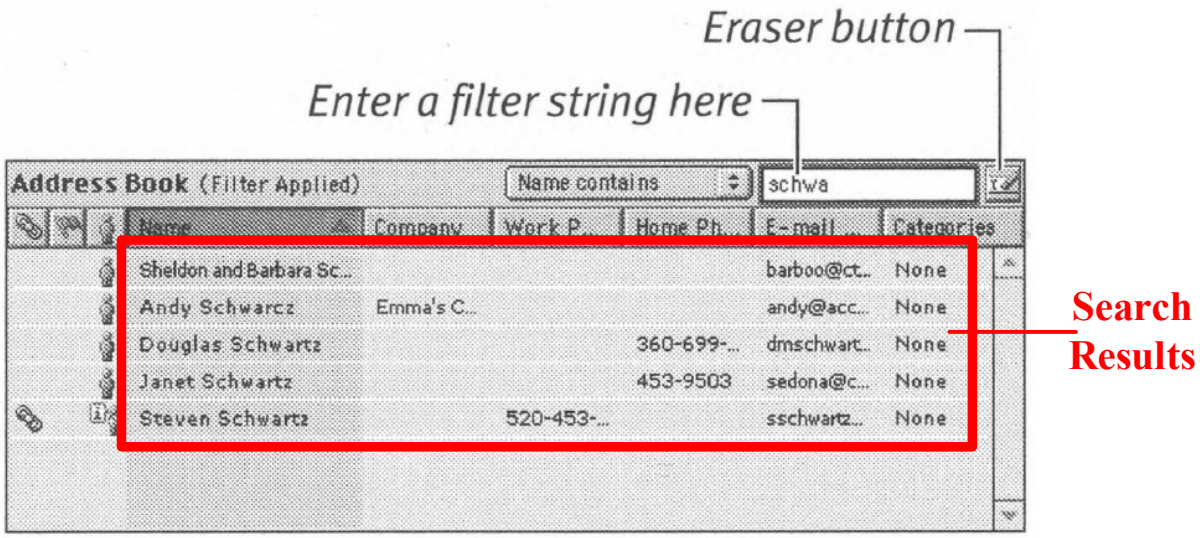


Figure 3.4 You can limit the display to records that contain a particular text string.

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

7. Claims 10-11

> 10. The computing system of claim 9, wherein the software instructions further cause the computing system to: receive selection of a first file of the displayed set of matching files; and display in the interactive user interface a portion of the first file.
>
> 11. The computing system of claim 10, wherein the software instructions further cause the computing system to: highlight a plurality of occurrences of the first and second partial search strings in the first file while not highlighting portions of text between occurrences of the first and second partial search strings in the first file.

188. 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.

189. For example, a user can select a particular file (e.g., email, contact) and the body 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.

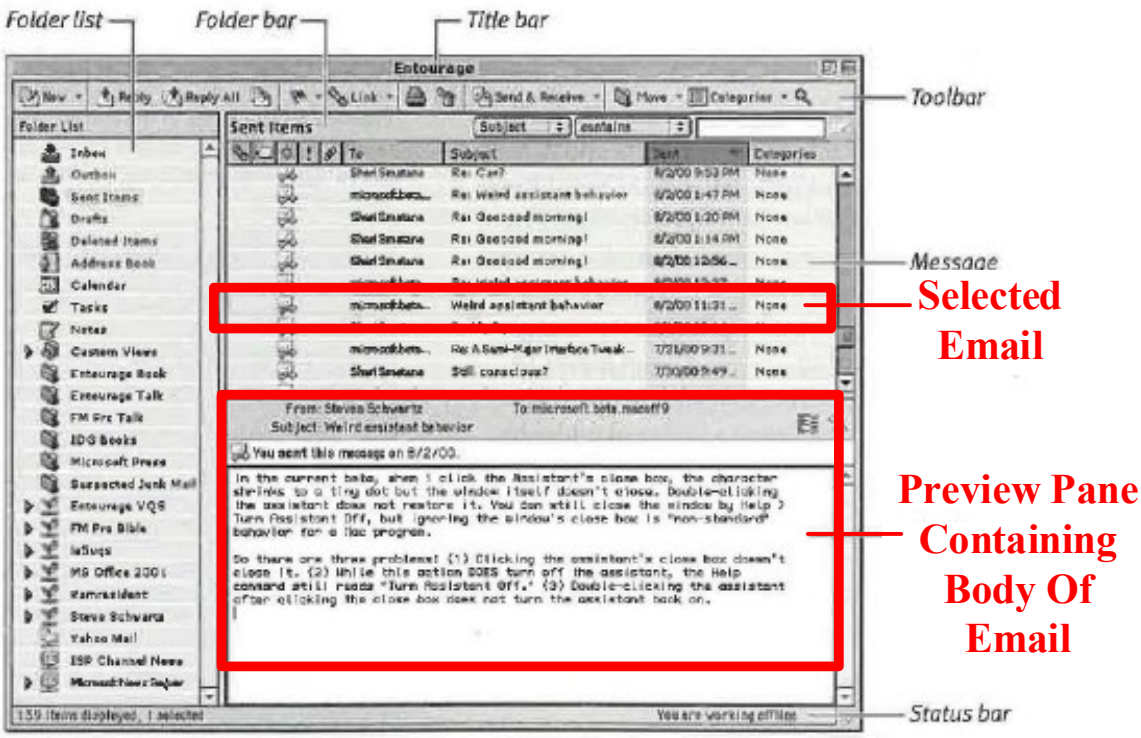
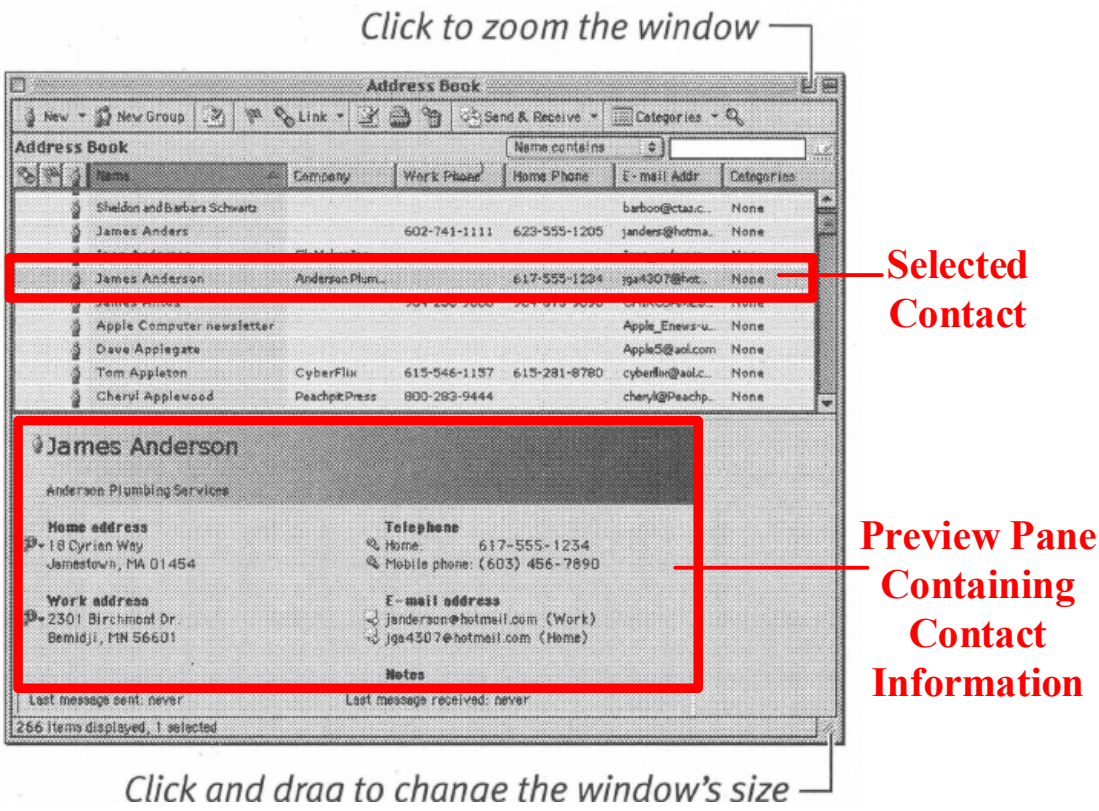


Figure 1.4 The main Entourage window.

EX1004, 3 (red annotations added).



EX1004, 38 (red annotations added).

190. Moreover, even assuming *arguendo* the Entourage references do not expressly disclose displaying a portion of a selected document that appears in a set of search results (which I do not agree with), it would have been obvious to a POSITA to do so to provide the user with more information about each of the search results and allow them to make a more informed decision about which document to open, as was well known in the art and within a POSITA's knowledge. *Supra* Section VI.B.4.

191. The Entourage references also disclose highlighting the search term in the search results. EX1004, 11, 163; *see also* EX1043, 386.

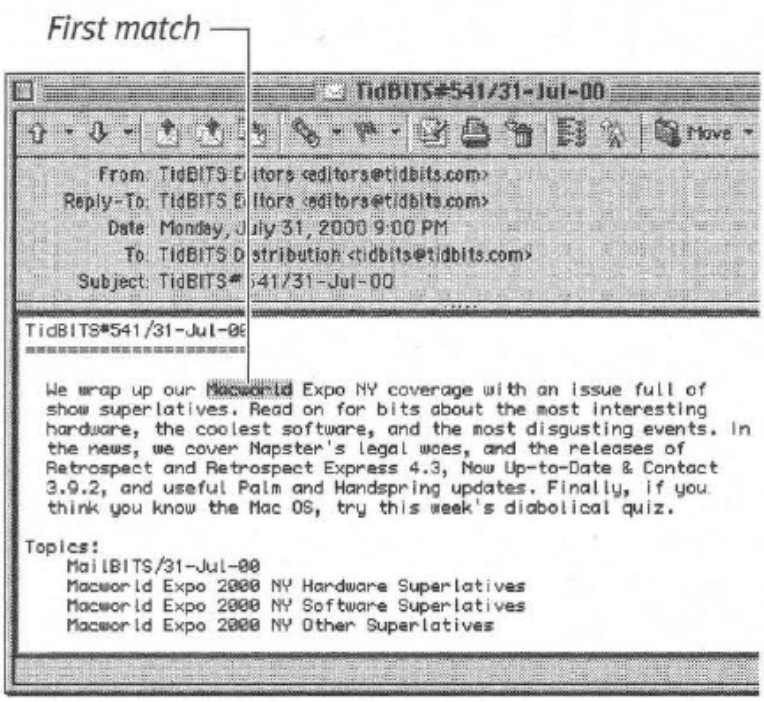


Figure 1.16 When you search for matching text within an item, Entourage highlights the first match, scrolling as necessary to display it.

EX1004, 11.

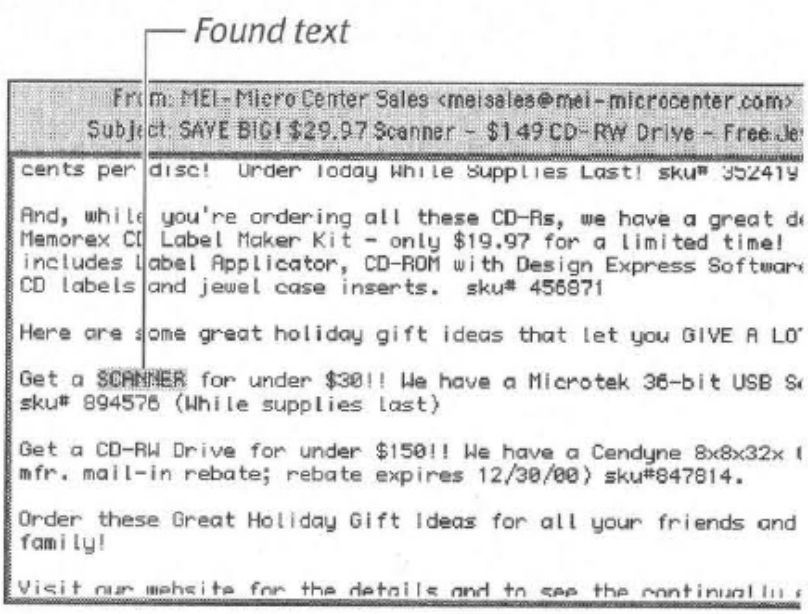


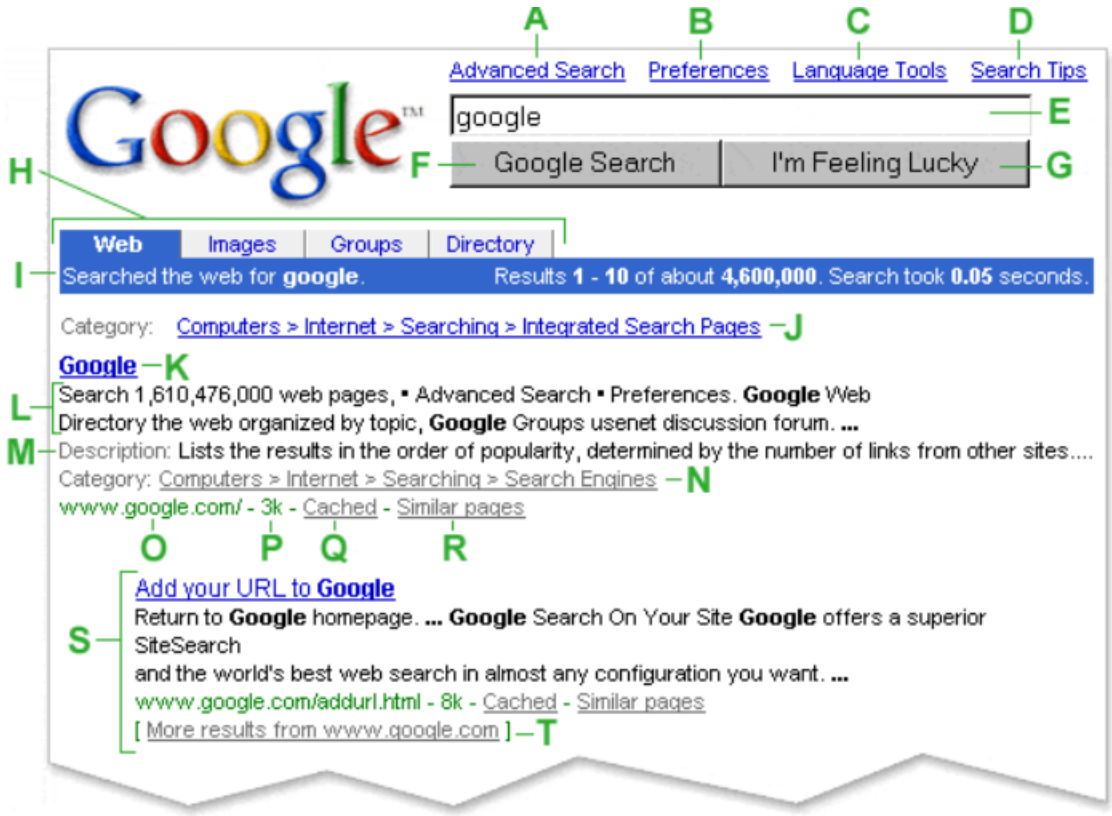
Figure 11.20 The first instance of the found text—if there is one—is highlighted for you.

EX1004, 163.

192. 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 VI.B.4.

193. For example, as explained by Baeza-Yates, Johnson, and others, highlighting a plurality of occurrences of the search string(s) provides several benefits, including pointing the user to the relevant portions of the document and increasing the user's confidence in the search results. EX1023, 101, 103, 207, 289; EX1027, 3-10; *supra* Section VI.B.4. As just a few examples, Google (EX1038),

Shneiderman (EX1044), Jensen (EX1035), and Glossbrenner (EX1057) each disclose highlighting every occurrence of the search string(s) in the search results.



EX1038, 56 (APPX L) (showing every occurrence of the searched word "google").

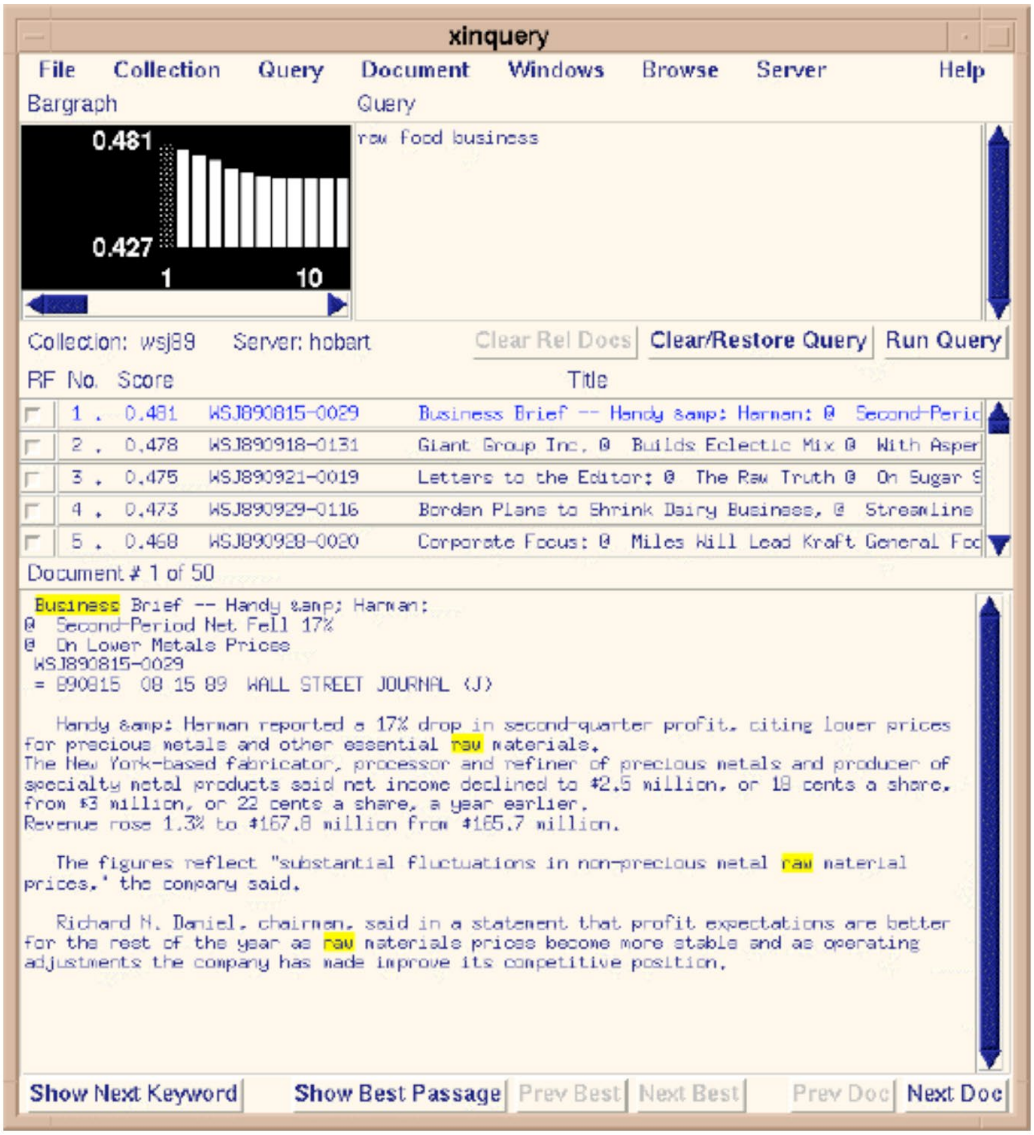


Figure 1. XINQUERY

EX1044, 5 (showing every occurrence of the searched strings “business” and “raw”).

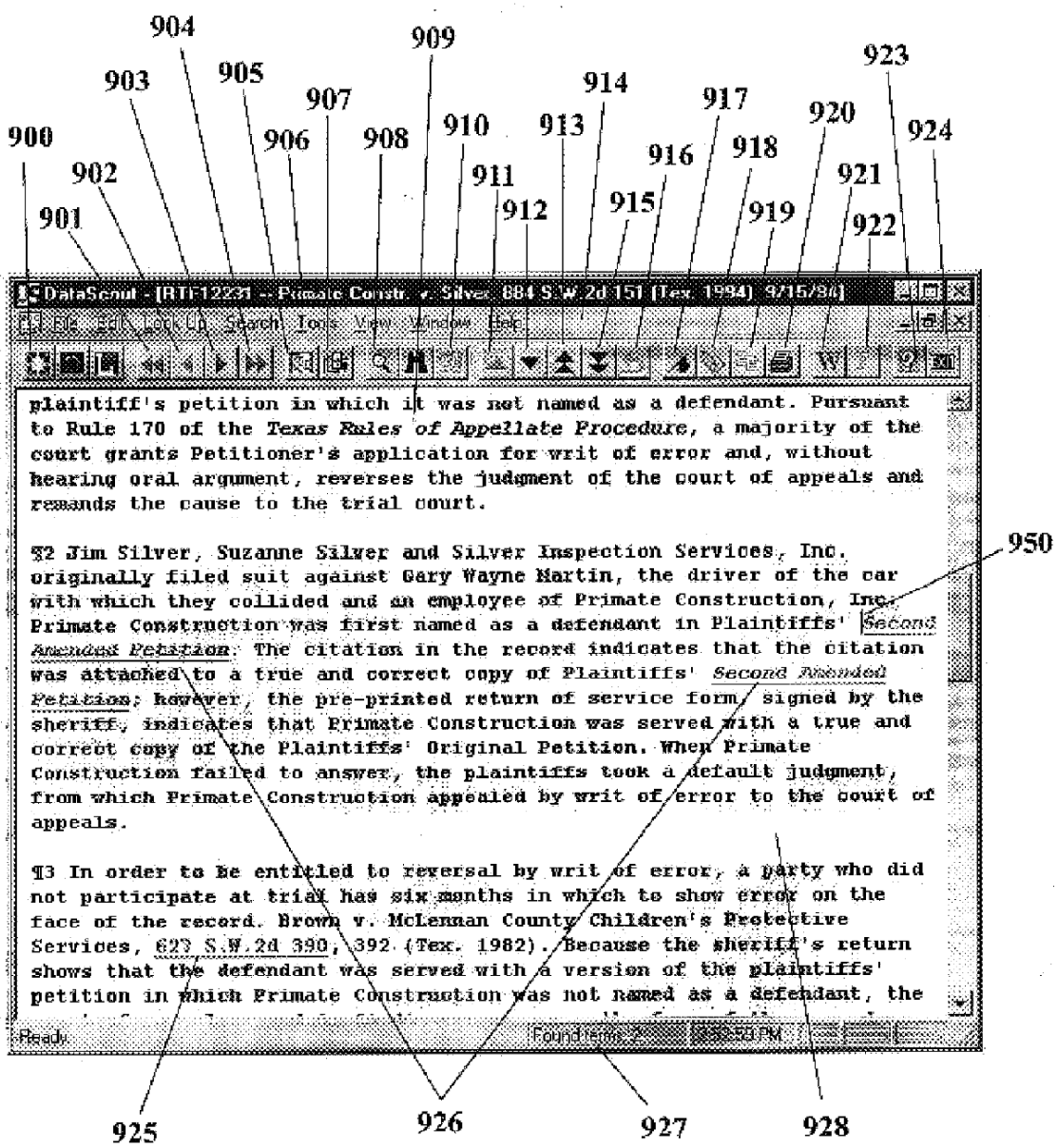


Fig. 9

EX1035, FIG. 9 (showing highlighting, via italicizing and underlining, every occurrence of the searched term “second amended petition”).

Human Genome Project History
SHORT **HISTORY**. Russian State **Human Genome Project** was started in 1988 when professor Alexander ...
www.eimb.relarn.ru/hugo/01.htm - 3k - Cached - Similar pages

History of the Human Genome Project
... The DOE **Human Genome** Program and the NIH National **Human Genome** Research Institute (NHGRI) together make up the US **Human Genome Project**. **History** of the **Project**. ...
www.ornl.gov/hgmis/project/hgp.html - 42k - Cached - Similar pages

1991 History of the US Human Genome Project
... A brief **history** of the US Department of Energy (DOE ... as those of the collaborative US **Human Genome Project**. The Office of Health and Environmental ...
www.ornl.gov/hgmis/project/91history.html - 40k - Cached - Similar pages

History of the UK Human Genome Project
... **History** of the UK **Human Genome Project** 1990 - Drs Sulston and Coulson in collaboration with Dr Waterston (Washington University, USA) began a pilot study of ...
www.wellcome.ac.uk/en/genome/history.html - 14k - Cached - Similar pages

Figure 6.24 To produce these results, we searched for `allintitle:human genome project history`. Note that all of our search terms appear in bold in the Web page titles.

EX1057, 88.

1. The *Mozart* Project
The life, times and music of Wolfgang Amadeus **Mozart**. Includes a complete Köchel listing, bibliography, **biography**, links to related sites and...
URL: www.frontiernet.net/~sboerner/mozart/
Directory Match: Wolfgang A Mozart Bios & Works
Translate More pages from this site Related pages Facts about: Frontier Corp

2. *Mozart* Discussion
Discussion of Wolfgang Amadeus **Mozart** on the web....
URL: mozart.composers.net/wwwboard/index.html
Translate More pages from this site Related pages

3. The *Mozart* Project: Bibliography
HOME | **BIOGRAPHY** | COMPOSITIONS | SELECTED ESSAYS | BIBLIOGRAPHY
Mozart's Works and Writings. Anderson, Emily, ed. The...
URL: trpe.reflektor.cz/mozart/books/index.html
Translate More pages from this site Related pages

4. *Mozart, Wolfgang Amadeu: Biography*
Site ©1996 Timothy A. Smith. Wolfgang Amadeus **Mozart** 1756-1791.
Child prodigy **Mozart** made his first public appearance, at six years of age, in a...
URL: jan.ucc.nau.edu/~tas3/wamozart.html
Translate Related pages Facts about: Northern Arizona ...

Figure 5.23 Here's an example of how AltaVista presents search results for a query that includes the terms *Mozart* and *biography*.

EX1057, 63.

Yahoo! Category Matches (1 - 1 of 1)

Business and Economy > Shopping and Services > Antiques and **Collectibles** > Mechanical Musical Instruments

- Music Boxes

Yahoo! Site Matches (1 - 19 of 34)

Regional > U.S. States > Arizona > Cities > Phoenix > Business and Shopping > Shopping and Services > Antiques and **Collectibles**

- Sorrento **Music Boxes** - offers Italian inlaid **music boxes**, jewelry boxes, and musical Capodimonte carousels.

Business and Economy > Shopping and Services > Antiques and **Collectibles** > Mechanical Musical Instruments

- Mechantiques - dealer in antique mechanical musical instruments. Including disc or cylinder type **music boxes**, musical clocks, nickelodeons and more. Buy, sell, trade.
- Mechanical Musical Memories - **music boxes**, phonographs, nickelodeons, player pianos, and band organs.

Business and Economy > Shopping and Services > Books > Booksellers > Recreation and Sports > Antiques and **Collectibles**

- Arlington Book Co. - specializes in books on barometers, clocks, fountain pens, furniture, jewelry, **music boxes**, tools, trademarks, watches, wristwatches, and related subjects.

Figure 10.8 Here's an example of how Yahoo presents search results. Category Matches come first, followed by Site Matches.

EX1057, 163.

194. Finally, a POSITA would have understood or at least found it obvious to highlight the occurrences of both partial search strings (not just one of them) to "help draw the user's attention to the parts of the document most likely to be relevant to the query" (EX1023, 289), as was well known in the art. *Supra* Section VI.B.4. It would have been nonsensical to highlight the occurrences of only one of the strings or highlight only a single instance of one of the strings. A POSITA

would have understood that highlighting a plurality of occurrences of both search strings provides the user with more context, helps the user quickly determine how relevant the file is (a file with many occurrences of the search strings is likely more relevant than a file with fewer occurrences), and helps draw the user to all potentially relevant portions of the document, not just some of them.

8. Claim 12

> 12. The computing system of claim 1, wherein the string separator character is selected from: a space, "<", "(", "[", "{", "'", "!", "@", "#", "$", "%", "^", "&", "*", ")", ";", ":", "'", "?", ">", ",", "=", and "+".

195. The Entourage+True+Baeza-Yates combination renders this claim obvious at least because, as explained above for claim element [1.10], the Entourage references teach that the string separator character is a space. *Supra* Section IX.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. *Supra* Section VI.B.2; EX1057, 60-61, 84, 104, 109, 125, 129, 143, 146, 151, 163, 166, 183. Thus, in view of the Entourage+True+Baeza-Yates combination and their own knowledge, a POSITA would have understood or at least found it obvious that the string separator character is selected from at least a space and plus sign (+).

9. Claim 13

> 13. The computing system of claim 1, wherein the files comprise emails, text documents, contact information, appointment information, task information, music files, system files, photographs, videos, and/or images.

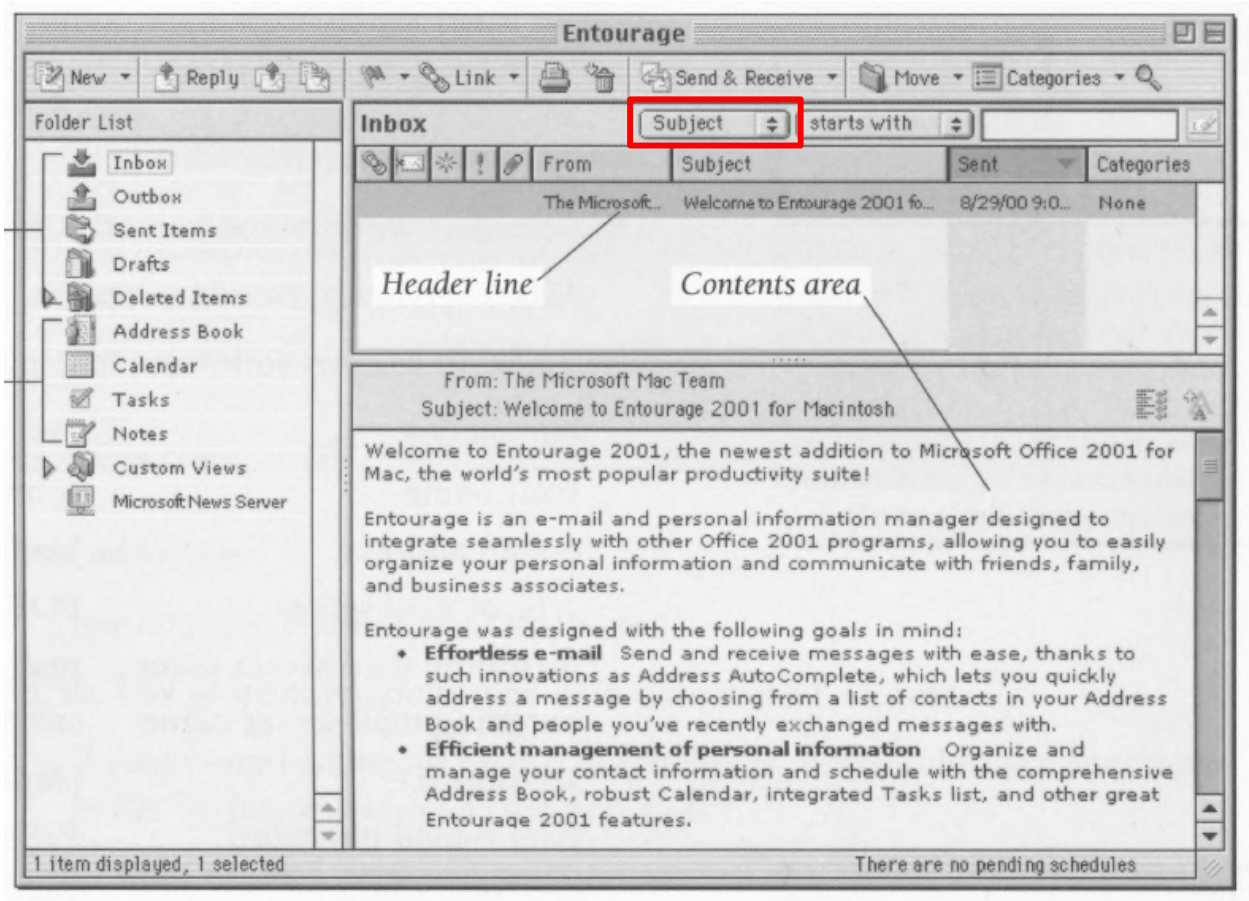
196. The Entourage+True+Baeza-Yates combination renders this claim obvious at least because, as explained above for elements [1.11-1.13], the Entourage references disclose that the files comprise emails, contact information, task information, and text documents (news articles). *Supra* Section IX.E.1.g; EX1004, 2-3, 8-14, 38-39, 89-90, 161-164, 204-205, 209; EX1005, 173; EX1006, 358-384.

10. <u>Claims 14-15</u>

14. The computing system of claim 1, wherein the first search field corresponds to a single field of the files and matching files are identified based only on content of the single field of the files. 15. The computing system of claim 14, wherein the single field comprises one of: to, from, cc, bcc, subject, folder name, date, size, attachments, or body fields of email files.

197. The Entourage+True+Baeza-Yates combination renders this claim 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 documents and only that field of the documents is incrementally searched. *Supra* Section IX.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 IX.A; 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.



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

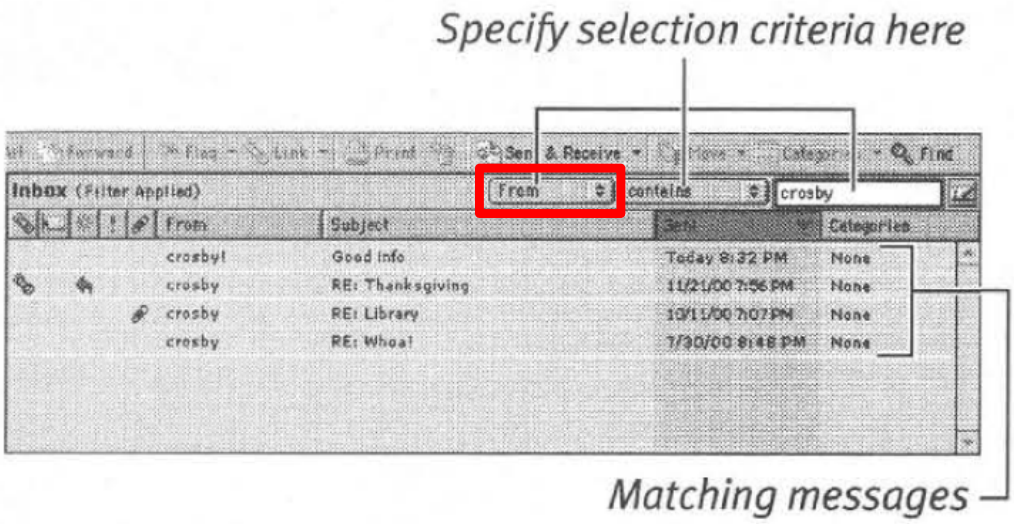
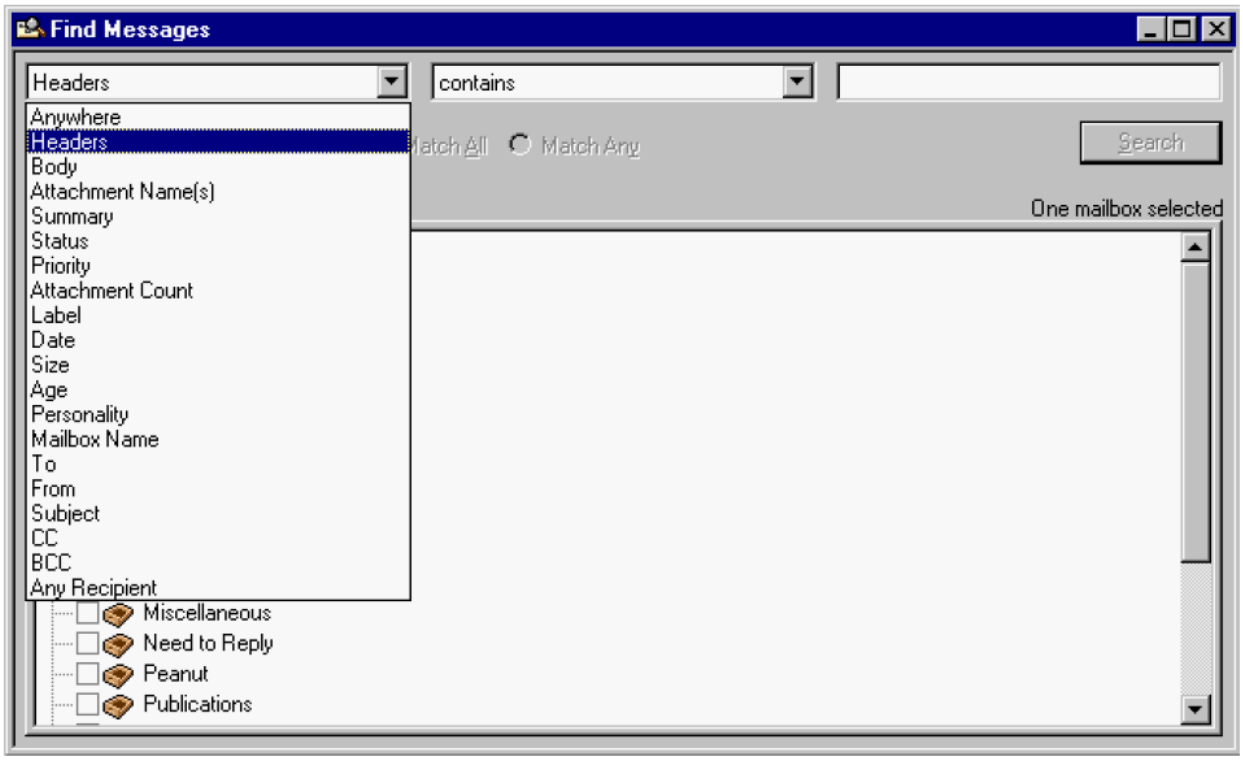


Figure 11.15 To locate messages quickly based on Subject, From, To, or Category information, you can enter selection criteria above the message list.

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 VI.B.5.

Find Messages drop-down and conjunction options lists



EX1007, 129.

11. Claim 16

> 16. The computing system of claim 14, wherein the single field comprises one of: name, path, size, item type, date modified, date created, or content fields of document files.

198. The 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 VI.B.1, IX.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 prior to the priority date of the ’490 patent. *Supra* Section IX.E.4.

12. Claim 17

> 17. The computing system of claim 14, wherein the single field comprises one of: name, email address, phone number, company, job title, or content fields of contacts files.

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

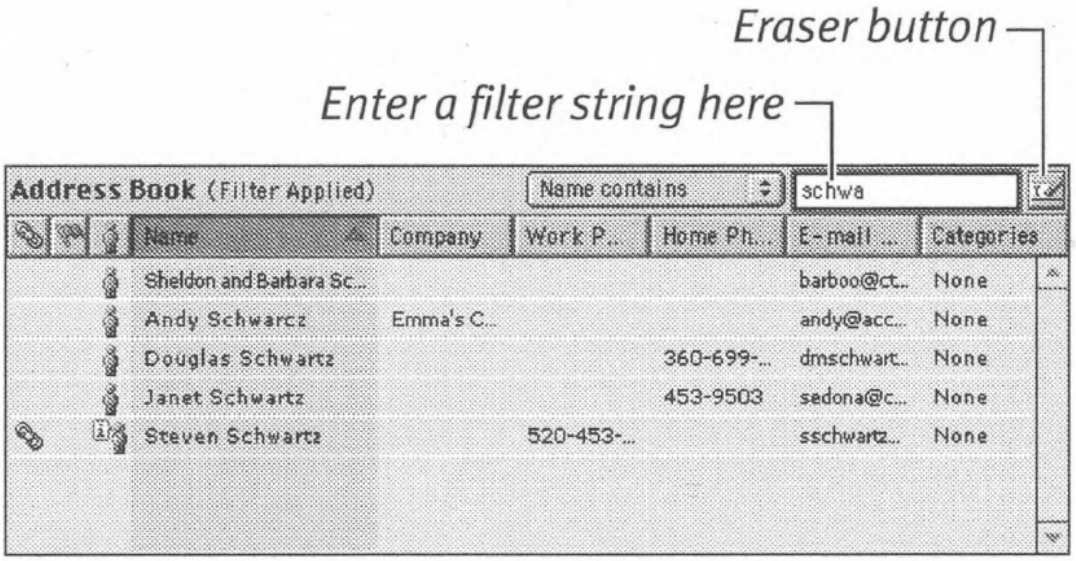


Figure 3.4 You can limit the display to records that contain a particular text string.

EX1004, 38.

13. <u>Claim 18</u>

18. The computing system of claim 14, wherein the single field comprises one of: subject, start date/time, duration, recipients, location, recurrence, attendees, attachments, or content fields of calendar files.

200. 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 VI.B.1, IX.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 thus 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 IX.E.4.

14. Claim 19

> 19. The computing system of claim 1, wherein the first partial search string applies to all textual content of the files and the second partial search string applies to only a single field of the files.

201. 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 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, a POSITA would have understood that a search query of "lyrics AND title: John Lennon" would search all textual content of the files for "lyrics" and only the title field of the files for "John Lennon" and would return files containing "lyrics" somewhere in the file and "John Lennon" in the title of the file. Thus, the Entourage+True+Baeza-Yates combination renders this claim obvious.

15. Claim 20

20. The computing system of claim 1, wherein modifiers can be applied to either or both of the first partial search string and second partial search string to include one or more operators selected from: AND, OR, NOT, NEAR, date ranges, and size ranges.

202. The Entourage+True+Baeza-Yates combination renders this claim obvious at least because Baeza-Yates discloses including Boolean operators in a single search field 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 VI.B.2, IX.E.1.g; EX1020, 5:5-10; 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."); EX1053, 11 ("The adopted query type is the 'boolean and'."); EX1057, 19-26, 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.").

X. NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS

203. I am aware that Patent Owner raised secondary considerations arguments in another proceeding (an ITC proceeding) and reviewed the exhibits that Patent Owner submitted in that proceeding to support its secondary considerations allegations (EX1063-EX1074). Nothing in those exhibits, however, changes my opinion about the obviousness of the challenged claims or suggests that the objective indica would overcome the strong showing of obviousness presented in Grounds 1-2. At best, the exhibits cited by Patent Owner in the ITC proceeding point to claim features that were well known in the art, such as incremental searching. For example, EX1067 confirms that Firefox had such incremental searching, stating that: "As you type [X1's search tool] eliminates

documents with no matches. Users of the search feature in Firefox are familiar with how this approach works." EX1067, 1. EX1073 also confirms that Outlook and Lookeen had document previewing and term highlighting functionality, stating that "Much like Outlook—or Lookeen—for that matter, X1 Search shows the list of the items it finds along with a preview pane so you can browse through the list and see a preview of each item as you highlight it." EX1073, 4.

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

Dated: December 19, 2024

Signature

APPENDIX A

Loren G. Terveen

terveen@umn.edu	University of Minnesota	(612) 624-8310

Address Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455

Education	Ph.D. in Computer Sciences	University of Texas	1991
	M.S. in Computer Sciences	University of Texas	1988
	B.A. in Computer Science, Mathematics, History	University of South Dakota	1984
Affiliations	Member of ACM, ACM/SIGCHI		
Research Interests	Human-Computer Interaction, Social Computing, Artificial Intelligence		
Professional Experience	Professor	University of Minnesota	2002 --
	Distinguished McKnight University Professor		2017 --
	Principal Member of Technical Staff	AT&T Labs - Research	1996-2002
	Member of Technical Staff	AT&T Bell Labs	1991-1996
	Graduate Research Intern	Microelectronics and Computer Corporation	1986-19991
	Teaching Assistant	University of Texas at Austin	1985

AWARDS AND MAJOR ROLES

ACM Distinguished Scientist (awarded 2009)

ACM SIGCHI Academy (2019)

ACM SGICHI Lifetime Service Award (2023)

ACM SIGCHI President (2015-2018)

ACM Council, SIG Governing Board Representative (2016 – 2018)

RESEARCH

External Funding

NSF:CISE: Community-Engaged Design and Implementation of a Framework for Ethical Online Communities Research (**Co-PI**, with Svetlana Yarosh PI, Stevie Chancellor Co-PI, Joseph Konstan Co-PI, Kangjie Lu Co-PI), $400,000 for 01/01/2023 to 12/31/2025.

Cisco: Detecting and Reducing Bias in Collaborative Virtual Meetings (**Co-PI**, with Stevie Chancellor PI). **$177,915** for 9/1/2021 to 12/31/2022.

NSF:CHS:Small: Incorporating and Balancing Stakeholder Values in Algorithm Design (UMN **PI**), Award Total $500,000, UMN Share $243,941 for 8/1/2019 to 7/31/2022.

NSF: EAGER: AI-DCL: Capture, Explain and Negotiate the Inherent Trade-offs in Machine Learning Algorithms (UMN **co-PI**), Award Total $295,713, UMN Share $103,267 for 10/1/2019 to 9/30/2022.

NSF:CHS: Small: Collaborative Research: Structured Data Peer Production: Addressing Challenges and Leveraging Opportunities (**PI**), $249,738 for 9/1/2018 - 8/31/2021.

National Cancer Institute (NCI): Restore: Improving sexual outcomes of gay and bisexual prostate cancer survivors (R. Simon Rosser **PI**; my role: **Co-PI);** $2,039,220 (directs) $3,039,541 (total), 2017-2022.

NSF: "PFI:BIC: Smart Human-Centered Collision Warning System: sensors, intelligent algorithms and human-computer interfaces for safe and minimally intrusive car-bicycle interactions" (Co-PI, with Rajesh Rajamani (PI), Max Donath, and Nichole Morris), $999,773 for September 1, 2016 to August 31, 2019.

NSF: "Computer-Supported Cooperative Work Doctoral Colloquium" (**PI**), $25,000 for 03/01/2016 to 02/28/2017.

National Institute on Drug Abuse: "A Technology-Delivered Peer-to-Peer Support ARB Adherence Intervention for HIV+ Adults", (Co-PI, with Keith Horvath (PI) and Darin Erickson), $3,302,62 for 07/01-2015 to 05/31/2020.

NSF: "HCC: Tools and Mechanisms to Support Social Participation Efforts", (**PI)**, $499,399 for 10/01/2012 to 09/30/2015.

NSF: "SoCS: Collaborative Research: Novel Algorithms and Interaction Mechanisms to Enhance Social Production", (**PI**), $527,140, for 7/01/2012 to 06/30/2015.

NSF: "Collaborative Research: Supporting Newcomer Socialization in Online Production Communities", (**PI**), $301,135.00 for 08/2011 to 08/31/2015.

Minnesota Department of Transportation: "Statewide Cycloplan: A Bicycle Planning Tool with Participatory GIS", (**PI**), $130,000, for 10/01/2011 to 06/30/2013.

Metropolitan Council: "Cycloplan II", (**PI**), $71,350, for 08/15/2011 to 05/31/2012.

NSF: "Wikisym Doctoral Consortium", (**PI**), $13,163, for 05/01/2011 to 04/30/2012.

IBM: "Mobile Crowdsensing", (**PI**), $100,000 awarded 04/01/2011

NSF: "Social-Computational Systems (SoCS) Community Meeting" (**PI**), $48,801 for 09/01/2010 to 08/31/2011

NSF: "SoCS: Collaborative Research: Information Framing: Intelligent Interfaces for an Online Production Community", (**PI)**, $375,000 for 09/15/2010 to 08/31/2015.

NSF: "Collaborative Research: Guiding Folksonomy Development to Enable Novel Tagging Applications" (**PI,** with J. Riedl and S. Sen (Macalester College)). $949, 788 for 04/1/2010 to 03/312014 (UMN Share).

Minnesota Department of Transportation: Bike, Bus, and Beyond: Extending Cyclopath to Enable MultiModal Routing (**PI**), $60,627, for 07/08/2010 to 01/31/2012

NIH: An Interactive Website to Promote Communication about Sexual Health and Dating Relationships between Parents and Teens (**Co-PI**, with Sonya Brady (PI), Simon Rosser, and Renee Sieving), $679,500 for 09/30/2009 to 08/31/2011.

Metropolitan Council: "Cycloplan" (**PI**), $185,000, October 2009 – March 2011.

NSF "Collaborative Research: Understanding Online Volunteer Communities: Toward Theory-Based Design" (**co-PI,** with J. Riedl, J. Konstan, M. Snyder, & Y. Ren; R. Kraut (CMU) $2,400,000 for 08/01/2008 to 07/31/2013.

NSF: "Recommender Systems Doctoral Consortium" (**PI**) $15,415, 2007-2008.

NSF: "Mining Spatiotemporal Data: From Personal Use to Community Knowledge" (**PI**) $449,570 for 12/1/2005 to 11/30/2009.

NSF: "Collaborative Research: Mark This! - Operationalizing the notion of "place" for interactive community systems" (**PI**, with Q. Jones (NJIT) and S. Whittaker (Univ. of Sheffield)). $173,411 for 6/1/2003 to 5/31/2007 (UMN share).

NSF: "Being There: Mobile Devices for Community and Commerce" (**PI**, with J. Konstan, J. Riedl, and S. Shekhar). $120,000 for 9/1/2002 to 8/31/2005.

NSF: “ITR: Collaborative Research: Designing On-Line Communities to Enhance Participation” (**co-PI**, with J. Konstan & J. Riedl, R. Kraut & S. Kiesler (CMU), P. Resnick and Y. Chen (Univ. of Michigan)). $1,246,017 for 9/1/2003 to 8/31/2009 (UMN share).

AT&T: “VURI: Collaborative Filtering and Intelligent Interface Design for Enhanced TV Applications” (**PI**). $35,000 for November 1, 2008 to October 31, 2009.

AT&T: “VURI: Collaborative Filtering and Intelligent Interface Design for Enhanced TV Applications” (**PI**). $35,000 for June 1, 2007 to May 31, 2008.

Internal Funding

Minnesota/China Collaborative Research Grant: "Expertise Oriented Mining for Web Community". $10,000 for July 1, 2007 to June 30, 2008 (PI, with Jie Tang, Tsinghua University, Beijing China).

University of Minnesota TEL grant: “The Next Generation Online Learning Environment: Designing for Community and Collaboration”. $10,00 for September 2006 to May 2007. (co-PI, with Joan Hughes, David Ernst, and Ann Ooms, College of Education and Human Development).

University of Minnesota Digital Technology Center: “Indoor Navigation Aids for Visually Impaired People”. $25,67 for June 2005 to December 2006. (co-PI, with S. Shekhar and G. Legge).

University of Minnesota Digital Technology Center: “Eye-Tracking Research on Community Websites: Photo Directories and Building Social Networks”. $19,300 for June 2004 to June 2005. (co-PI, with J. Konstan).

University of Minnesota Grant-In-Aid: “Facilitating Participation in Online Communities”. $20,397 for 1/1/2003 to 6/30/2004.

Books

1. Bickhard, M.H. and Terveen, L.G. Foundational Issues in Artificial Intelligence and Cognitive Science: Impasse and Solution, (1995), Elsevier Science.

Refereed Journal Papers

2. Mo Houtti, Moyan Zhou, Loren Terveen, and Stevie Chancellor. “All of the White People Went First": How Video Conferencing Consolidates Control and Exacerbates Workplace Bias. *PACM Human-Computer Interaction (2023).*
3. Mo Houtti, Isaac Johnson, Joel Cepeda, Soumya Khandelwal, Aviral Bhatnagar, and Loren Terveen. 2022. “We Need a Woman in Music”: Exploring Wikipedia’s Values on Article Priority. *PACM Human-Computer Interaction* 7, (2022).
4. Charles Chunkai Zhang*, Mo Houtti*, C, Estelle Smith, Ruoyan Kong, and Loren Terveen. Working for Invisible Machines or Pumping Information into an Empty Void? An Exploration of Wikidata Contributors’ Motivations. In *Proceedings of the ACM on Huma-Computer Interaction*, Vol. 6 (2022). (* co-first authors)
5. C. Estelle Smith, William Lane, Hannah Miller Hillberg, Daniel Kluver, Loren Terveen, and Svetlana Yarosh. Effective Strategies for Crowd-Powered Cognitive Reappraisal Systems: A

Field Deployment of the Flip*Doubt Web Application for Mental Health. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 5 (2021). **Best Paper Honorable Mention.**

6. C. Estelle Smith, Avleen Kaur, Katie Z. Gach, Loren Terveen, Mary Jo Kreitzer, and Susan O'Conner-Von. What is Spiritual Support and How Might It Impact the Design of Online Health Communities? In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 5 (2021).
7. Levonian, Z., Dow, M., Erickson, D.R., Ghosh, S., Miller Hillberg, H., Narayanan, S., Terveen, L. and Yarosh, L. Patterns of Patient and Caregiver Mutual Support Connections in an Online Health Community. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 4 (2020).
8. Zhu, H., Yu, B., Halfaker, A., and Terveen, L. Value-Sensitive Algorithm Design: Method, Case Study, and Lessons. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 2, No. 2 (2018).
9. Miller Hillberg, H., Levonian, Z., Terveen, L., and Hecht, B. What I See is What You Don't Get: Effects of Seeing Emoji Rendering Differences Across Platforms. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 2, No. 2 (2018).
10. Hall, A., Terveen, L., and Halfaker, A. Bot Detection in Wikidata Using Behavioral Cues. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 2, No. 2 (2018).
11. Sheppard, S.A., Turner, J., Thebault-Spieker, J., Zhu, H., and Terveen, L. Never Too Old, Cold or Dry to Watch the Sky: A Survival Analysis of Citizen Science Volunteerism. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 1, No. 2 (2017).
12. Thebault-Spieker, J., Kluver, D., Klein, M., Halfaker, A., Hecht, B., Terveen, L., and Konstan, J. Simulation Experiments On (The Absence of) Ratings Bias in Reputation Systems. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 1, No. 2 (2017).
13. Yu, B., Wang, X., Lin, A.Y., Ren, Y., Terveen, L., and Zhu, H. Out With The Old, In With The New? Unpacking Member Turnover in Online Production Groups. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 1, No. 2 (2017).
14. Nguyen, T.T., Harper, F.M., Terveen, L., and Konstan, J. User Personality and User Satisfaction with Recommender Systems. In *Information Systems Frontiers* (2017).
15. Filson Moses, J., Dwyer, P.C., Fugelstad, P.T., Kim, J.S., Maki, A., Synder, M., and Terveen, L. Encouraging Online Engagement: The Role of Interdependent Self-Construal and Social Motives in Fostering Online Participation. In *Personality and Individual Differences* (2017).
16. Thebault-Spieker, J., Terveen, L., and Hecht, B. Towards a Geographic Understanding of the Sharing economy: Systemic Biases in UberX and TaskRabbit. In *ACM Transactions on Computer-Human Interaction* (2017).
17. Brady, S.S., Sieving, R.E., Terveen, L.G., Rosser, R. S., Kodet, A.J., and Rothberg, V.D. An Interactive Website to Reduce Sexual Risk Behavior: Process Evaluation of TeensTalkHealth, JMIR Research Protocols (2015).
18. Ren, Y., Harper, F.M., Drenner, S., Terveen, L., Kiesler, S., Riedl, J., and Kraut, R.E. (2012). Building Member Attachment in Online Communities: Applying Theories of Group Identity and Interpersonal Bonds. *Management Information Systems Quarterly*.
19. Jones, Q., Grandhi, S., Karam, S., Whittaker, S., Zhou, C., and Terveen, L. Geographic 'Place' and Community Information Preferences, in *Computer-Supported Cooperative Work*.
20. Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., Terveen, L., Discovering Personally Meaningful Places from Location Data: An Interactive Clustering Approach. *ACM Transactions on Information System,* 25, 3 (July 2007).

21. Ling, K., Beenen, G., Ludford, P.J., Wang, X., Chang, K., Li, X., Cosley, D., Frankowski, D., Terveen, L., Rashid, A.M., Resnick, P., and Kraut, R.E. Using Psychology to Motivate Contributions to Online Communities. *Journal of Computer-Mediated Communication*, 10, 4 (June 2005).
22. Terveen, L. and McDonald, D. Social Matching: A Framework and Research Agenda. *ACM Transactions on Computer-Human Interaction*, 12, 3 (2005), 401-434.
23. Whittaker, S., Jones, Q., Nardi, B., Creech, M., Terveen, L., Isaacs, E., and Hainsworth, J. ContactMap: organizing communication in a social desktop, in *ACM Transactions on Computer-Human Interaction*, 11, 4 (December 2004), 445-471.
24. Jones, Q., Grandhi, S., Terveen, L., and Whittaker, S. People-To-People-to-Geographical-Places: The P3 Framework for Location-Based Community Systems. in *Computer-Supported Cooperative Work*, 13, 3-4 (August 2004), 249-282.
25. Herlocker, J.L., Konstan, J.A., Terveen, L.G., and Riedl, J.T. Evaluating Collaborative Filtering Recommender Systems, *ACM Transactions on Information Systems* (2004).
26. Amento, B., Terveen, L., Hill, W., Hix, D., and Schulman, R. Experiments in Social Data Mining: The TopicShop System, in *ACM Transactions on Computer-Human Interaction*, 10, 1 (March 2003), 54-85.
27. Whittaker, S., Terveen, L.G, and Nardi, B.A. Let's stop pushing the envelope and start addressing it, in *Human-Computer Interaction*, 15, 2-3 (Sep 2000), 75-106.
28. Terveen, L.G., Hill, W.C., and Amento, B. Constructing, Organizing, and Visualizing Collections of Topically Related Web Resources, in *ACM Transactions on Computer-Human Interaction*, 6, 1 (Mar. 1999), 67-94.
29. Selfridge, P.G. and Terveen, L.G. Knowledge Management Tools for Business Process Support and Reengineering, in *Journal of Intelligent Systems in Accounting, Finance, and Management* (Jan. 1996).
30. Terveen, L.G. An Overview of Human-Computer Collaboration, in *Knowledge-Based Systems*, 8, 2-3 (1995), 67-81.
31. Terveen, L.G., Selfridge, P.G., and Long, M.D. Living Design Memory: Framework, System, and Lessons Learned, in *Human-Computer Interaction*, 10, 1 (1995), 1-37.
32. Terveen, L.G. Intelligent Systems as Cooperative Systems, in *International Journal of Intelligent Systems*, 3, 2-4 (1993), 217-250.
33. Brachman, R.J., Selfridge, P.G., Terveen, L.G., Altman, B., Borgida, A., Halper, F., Kirk, T., Lazar, A., McGuinness, D.L., and Resnick, L.A. Integrated Support for Data Archaeology, in *International Journal of Intelligent and Cooperative Information Systems*, 2, 2 (1993), 159-185.

Refereed Conference Papers

34. Zhang, C and Terveen, L. Quantifying the Gap: A Case Study of Wikidata Gender Disparities. In *Proceedings of OpenSym 2021*, the 17th International Symposium on Open Collaboration
35. Yu, B., Ye, Y., Terveen, L., Wu, Z.S., Forlizzi, J., and Zhu, H. Keeping Designers in the Loop: Communicating Inherent Algorithmic Trade-offs Across Multiple Objectives. In *Proceedings of DIS 2020,* the ACM Conference on Designing Interactive Systems.
36. Smith, C.E., Yu, B., Srivastava, A., Halfaker, A., Terveen, L., and Zhu, H. Keeping Community in the Loop: Understanding Wikipedia Stakeholder Values for Machine

Learning-Based Systems. In *Proceedings of CHI 2020, the ACM Conference on Human Factors in Computing Systems.* **Best Paper Honorable Mention.**

37. Levonian, Z., Erikson, D., Luo, W., Narayanan, S., Rubya, S., Vachher, P., Terveen, L. and Yarosh. S. Bridging Qualitative and Quantitative Methods for User Modeling: Tracing Cancer Patient Behavior in an Online Health Community. In *Proceedings of ICWSM 2020, AAAI Conference on the Web and Social Media.*
38. Cheng, H.F., Yu, B., Fu, S., Zhao, J., Hecht, B., Konstan, J., Terveen, L., Yarosh, S., and Zhu, H. Teaching UI Design at Global Scales: A Case Study of Designing Collaborative Capstone Projects for MOOC Students, in *Proceedings of ACM Learning at Scale 2019.*
39. Hall, A., Thebault-Spieker, J., Sen, S., Hecht, B., and Terveen, L. Exploring the Relationship Between "Informal Standards" and Contributor Practice in OpenStreetMap. In *Proceedings of OpenSym 2018.*
40. Thebault-Spieker, J., Halfaker, A., Terveen, L., and Hecht, B. Distance and Attraction: Gravity Models for Geographic Content Production. In *Proceedings of CHI 2018, the ACM Conference on Human Factors in Computing Systems*.
41. Thebault-Spieker, J., Hecht, B., and Terveen, L. Geographic Biases are 'Born, not Made': Exploring Contributors' Spatiotemporal Behavior in OpenSteetMap. In *Proceedings of ACM Group* 2018.
42. Kang, J., Condiff, K., Chang, S., Terveen, L., Konstan, J., and Harper, F.M. Understanding How People Use Natural Language to Ask for Recommendations, in *RecSys 2017.*
43. Miller, H., Kluver, D., Thebault-Spieker, J., Terveen, L., and Hecht, B. Understanding Emoji Ambiguity in Context: The Role of Text in Emoji-Related Miscommunication. In *Proceedings of ICWSM 2017, AAAI Conference on the Web and Social Media.*
44. Hall, A., McRoberts, S., Thebault-Spieker, J., Lin, A.Y., Sen, S., Hecht, B., and Terveen, L. Freedom versus Standardization: Structured Data Generation in a Peer Production Community. In *Proceedings of CHI 2017, the ACM Conference on Human Factors in Computing Systems (CHI 2017)*.
45. Yu, B., Ren, Y., Terveen, L., and Zhu, H. Predicting Member Productivity and Withdrawal from Pre-Joining Attachments in Online Production Groups. In *Proceedings of CSCW 2017, The ACM Conference on Computer-Supported Cooperative Work and Social Computing.*
46. Chang, S., Harper, M., and Terveen, L Crowd-Based Personalized Natural Language Explanations for Recommendations In *Proceedings of RecSys 2016, The ACM Conference on Recommender Systems.*
47. Chang, S., Harper, M., He, L., and Terveen, L CrowdLens: Experimenting with Crowd-Powered Recommendation and Explanation. In *Proceedings of ICWSM 2016, AAAI Conference on the Web and Social Media.*
48. Miller, H., Thebault-Spieker, J., Chang, S., Johnson, I., Terveen, L., and Hecht, B. "Blissfully happy" or "ready to fight": Varying Interpretations of Emoji. In *Proceedings of ICWSM 2016, AAAI Conference Web and Social Media.*
49. Zhao, Q., Huang, Z., Harper, F.M., Terveen, L., and Konstan, J. "Precision Crowdsourcing: Closing the Loop to Turn Information Consumers into Information Producers". In *Proceedings of CSCW 2016, The ACM Conference on Computer Supported Cooperative Work and Social Computing.*

50. Harper, F.M., Xu, F., Kaur, H., Condiff, K., Chang, S., and Terveen, L. Putting Users in Control of their Recommendations. In *Proceedings of RecSys2015, The ACM Conference on Recommender Systems.*
51. Kapoor, K., Kumar, V., Terveen, L., Konstan, J.A., and Schrater, P. "I like to explore sometimes" – Adapting to Dynamic User Novelty Preferences. In *Proceedings of RecSys2015, The ACM Conference on Recommender Systems.*
52. Warncke-Wang, M., Ranjan, V., Terveen, L., and Hecht, B. Misalignment Between Supply and Demand of Quality Content in Peer Production Communities. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM'15).*
53. Chang, S., Harper, F.M., and Terveen, L. Using Groups of Items to Bootstrap New Users in Recommender Systems. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing* (CSCW '15). ACM, New York, NY, USA.
54. Miller, H., Chang, S., and Terveen, L. "I LOVE THIS SITE!" vs. "It's a little girly": Perceptions of and Initial User Experience with Pinterest. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing* (CSCW '15). ACM, New York, NY, USA.
55. Thebault-Spieker, J., Terveen, L., and Hecht, B. 2015. Avoiding the South Side and the Suburbs: The Geography of Mobile Crowdsourcing Markets. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing* (CSCW '15). ACM, New York, NY, USA.
56. Warncke-Wang, M., Hecht, B., and Terveen, L. The Success and Failure of Quality Improvement Projects in Peer Production Communities. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing* (CSCW '15). ACM, New York, NY, USA.
57. Panciera, K., Masli, M., and Terveen, L. Crème de la Crème: Elite contributors in an online community, in *Proceedings of the 10th International Symposium on Open Collaboration* (OpenSym 2014).
58. Kumar, V., Kluver, D., Terveen, L., and Riedl, J. More Efficient Tagging Systems with Tag Seeding, in Proceedings of IEEE SocialCom 2014.
59. Nguyen, T., Hui, P., Harper, F.M., Terveen, L., and Konstan, J. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity, in Proceedings of WWW 2014.
60. Halfaker, A., Geiger, S., and Terveen, L. Snuggle: Designing for Efficient Socialization and Ideological Critique, in Proceedings of CHI 2014.
61. Chang, S., Kumar, V., Gilbert, E., and Terveen, L. Specialization, Homophily, and Gender in a Social Curation Site Findings from Pinterest, in Proceedings of CSCW 2014.
62. Grevet, C., Terveen, L., and Gilbert, E. Managing Political Differences in Social Media, in Proceedings of CSCW 2014.
63. Masli, M., and Terveen, L. Leveraging the Contributory Potential of User Feedback, in Proceedings of CSCW 2014.
64. Sheppard, S.A., Wiggins, A., and Terveen, L. Capturing Quality: Retaining Provenance for Curated Volunteer Monitoring Data, in Proceedings of CSCW 2014.
65. Delong, C., Terveen, L., and Srivastava, J. (2013). TeamSkill and the NBA: Applying Lessons from the Virtual World to the Real World, in Proceedings of ASONAM 2013.

66. Torre, F., Liu, Y., Liu, Z., and Terveen, L. (2013). Local Knowledge Matters for Crowdsourcing Systems: Experience from Transferring an American Site to China, in Proceedings of ICWSM 2013.
67. Gilbert, E., Bakshi, S, Chang, S., and Terveen, L. (2013). "I Need to Try This!": A Statistical Overview of Pinterest, in Proceedings of CHI 2013.
68. Torre, F., Pitchford, D., Brown, P., and Terveen, L. (2012). Matching GPS Traces to (Possibly) Incomplete Map Data: Bridging Map Building and Map Matching, in ACM SIGSPATIAL GIS 2012.
69. Dunne, L, Zhang, J., and Terveen, L. (2012). An Investigation of Contents and Use of the Home Wardrobe, in UbiComp 2012.
70. Dong, Z., Shi, C., Sen, S., Terveen, L., and Riedl, J. (2012). War Versus Inspirational in Forrest Gump: Cultural Effects in Tagging Communities, in ICWSM 2012.
71. Masli, M. and Terveen, L. (2012) Evaluating Compliance-Without-Pressure Techniques for Increasing Participation in Online Communities, in CHI 2012.
72. Fuglestad P.T., Dwyer, P.C., Filson Moses, J., Kim, J.S., Mannino, C.A., Terveen, L., and Snyder, M. (2012) What Makes Users Rate (Share, Tag, Edit…)? Predicting Patterns of Participation in Online Communities, in CSCW 2012.
73. Nathan, M., Topkara, M., Lai, J., Pan, S., Wood, S., Boston, J., and Terveen, L. (2012) In Case You Missed It: Benefits of Attendee-Shared Annotations for Non-attendees of Remote Meetings, in CSCW 2012.
74. Priedhorsky, R., Pitchford, D., Sen, S., and Terveen, L. (2012), Recommending Routes in the Context of Bicycling: Algorithms, Evaluation, and the Value of Personalization, in CSCW 2012.
75. Lam, S.K., Uduwage, A., Dong, Z., Sen, S., Musicant, D.R., Terveen, L., and Riedl, J. (2011). WP:Clubhouse? An Exploration of Wikipedia's Gender Imbalance, in Wikisym 2011. **Best Paper Winner.**
76. Panciera, K., Masli, M., and Terveen, L.G. (2011). "How Should I Go from __ to __ without Getting Killed? Motivations and Benefits in Open Collaboration", in Wikisym 2011.
77. Priedhorsky, R., and Terveen, L.G. (2011). Wiki Grows Up: Arbitrary Data Models, Access Control, and Beyond, in Wikisym 2011.
78. Sheppard, S.A. and Terveen, L.G. (2011). Quality is a Verb: The Operationalization of Data Quality in a Citizen Science Community, in Wikisym 2011.
79. Masli, M., Priedhorsky, R., and Terveen, L. (2011). Task Specialization in Social Production Communities: The Case of Geographic Volunteer Work, in the Proceedings the 4th International AAAI Conference on Weblogs and Social Media (ICWSM 2011).
80. Torre, F., Sheppard, S.A., Priedhorsky, R., and Terveen, L. (2010) bumpy, caution with merging: An Exploration of Tagging in a Geowiki, in GROUP 2010.
81. Panciera, K., Priedhorsky, R., Erickson, T., and Terveen, L. (2010). Lurking? Cyclopaths? A Quantitative Analysis of User Behavior in a Geowiki, in CHI 2010. **Best of CHI Nominee**.
82. Priedhorsky, R., Masli, M., and Terveen, L. (2010). Eliciting and Focusing Geographic Volunteer Work, in CSCW 2010.
83. Panciera, K., Halfaker, A., and Terveen, L. (2009). Wikipedians are Born, Not Made: A Study of Power Editors on Wikipedia, in GROUP 2009. *36% Acceptance Rate.*
84. Reily, K., Ludford Finnerty, P., and Terveen, L. Two Peers are Better than One: Aggregating Peer Reviews for Computing Assignments is Surprisingly Accurate, in GROUP 2009. *36% Acceptance Rate.*

85. Ludwig, M., Priedhorsky, R., and Terveen, L. (2009). Path Selection: A Novel Interaction Technique for Mapping Applications, in CHI 2009. *24% acceptance rate*.
86. Priedhorsky, R., and Terveen, L. (2008). The Computational Geowiki: What, Why, and How, in CSCW 2008. *23% acceptance rate*. **Best of CSCW Nominee**.
87. Drenner, S., Sen, S., and Terveen, L (2008). Crafting the Initial User Experience to Achieve Community Goals, in RecSys 2008. *30% acceptance rate*.
88. Reily, K., Ludford, P., and Terveen, L. (2008). Sharescape: An Interface for Place Annotation, in NordiCHI 2008. *30% acceptance rate*.
89. Nathan, M., Harrison, C., Yarosh, S., Terveen, L., Stead, L., and Amento, B. (2008), CollaboraTV: Making Television Viewing Social Again, in uxTV 2008.
90. Priedhorsky, R., Jordan, B., and Terveen, L. (2007), How a Personalized Geowiki Can Help Bicyclists Share Information More Effectively, in WikiSym 2007. 50% acceptance rate.
91. Priedhorsky, R., Chen, J., Lam, A., Panciera, K., Terveen, L., and Riedl, J. (2007), Creating, Destroying, and Restoring Value in Wikipedia, in ACM Group 2007.
92. Rouben, A. and Terveen, L. (2007), Speech and Non-Speech Audio: Navigational Information and Cognitive Load, in International Conference on Auditory Displays (ICAD).
93. Ludford, P., Priedhorsky, R., Reily, K., and Terveen, L. (2007), Capturing, Sharing, and Using Local Place Information, in CHI 2007. *25% acceptance rate*.
94. Cosley, D., Frankowski, D., Terveen, L., and Riedl, J. (2007), SuggestBot: Using Intelligent Task Routing to Help People Find Work in Wikipedia, in IUI 2007. *22% acceptance rate*.
95. Harper, F.M., Frankowski, D., Drenner, S., Ren, Y., Kiesler, S., Terveen, L., Kraut, R., and Riedl, J. (2007), Talk Amongst Yourselves: Inviting Users To Participate In Online Conversations, in IUI 2007. *22% acceptance rate*.
96. Frankowski, D., Cosley, D., Sen, S., Terveen, L., and Riedl, J. You Are What You Say: Privacy Risks of Public Mentions, in *SIGIR 2006*. *19% acceptance rate.*
97. Ludford, P.J., Frankowski, D., Reily, K., Wilms, K., and Terveen, L., Because I Carry My Cell Phone Anyway: Functional Location-Based Reminder Applications, in *Proceedings of CHI 2006*. *23% acceptance rate*.
98. Cosley, D., Frankowski, D., Terveen, L, and Riedl, J., Using Intelligent Task Routing and Contribution Review to Help Communities Build Artifacts of Lasting Value, in *Proceedings of CHI 2006*. *23% acceptance rate*.
99. Drenner, S., Harper, M., Frankowski, D., Riedl, J., and Terveen, L. Insert Movie Reference Here: A System to Bridge Conversation and Item Oriented Web Sites, in *Proceedings of CHI 2006* (Tech Note).
100. Zhou, C., Ludford, P., Frankowski, D., and Terveen, L. How Do People's Concepts of Place Relate to Physical Locations? In *Proceedings of INTERACT 2005*. *27% acceptance rate.*
101. D. Cosley, D. Frankowski, S. Kiesler, L. Terveen, J. Riedl. How Oversight Improves Member-Maintained Communities. In *Proceedings of CHI 2005*, Portland, OR, 2005. *25% acceptance rate*.
102. Jones, Q., Grandhi, S., Whittaker, S., Chivakula, K, and Terveen, L. Putting Systems into Place: A Qualitative Study of Design Requirements for Location Aware Community Systems, in *Proceedings of CSCW 2004*. *30% acceptance rate*.
103. Zhou, C., Ludford, P., Shekhar, S., and Terveen, L. Discovering Personal Gazetteers: An Interactive Clustering Approach, in *ACM GIS 2004* (12th International Symposium on Geographic Information Systems). *33% acceptance rate.*

104. Ludford, P., Cosley, D., Frankowski, D., and Terveen, L.G. Think Different: Increasing Online Community Participation Using Uniqueness and Group Dissimilarity, in *Proceedings of CHI 2004*. *16% acceptance rate*.
105. Cosley, D., Ludford, P. and Terveen, L.G. Studying the Effect of Similarity in Online Task-Focused Interactions, *Proceedings of GROUP 2003*. *35% acceptance rate*.
106. Ludford, P. and Terveen, L.G. Does an Individual's Myers-Briggs Type Indicator Preference Influence Task-Oriented Technology Use?, *Proceedings of Interact 2003*. *34% acceptance rate*.
107. Whittaker, S., Jones, Q., and Terveen, L.G. Contact Management: Identifying Contacts to Support Long-Term Communication, *Proceedings of CSCW 2002*, 216-225. *20% acceptance rate*.
108. Terveen, L.G., McMackin, J., Amento, B., and Hill, W. Specifying Preferences Based On User History, *Proceedings of CHI 2002*, 315-322. *15% acceptance rate.*
109. Whittaker, S., Jones, Q., and Terveen, L.G. Managing Long Term Conversations: Conversation and Contact Management, *Proceedings of HICSS 2002*. *50% acceptance rate.*
110. Amento, B., Terveen, L., Hill, W., and Hix, D. TopicShop: Enhanced Support for Evaluating and Organizing Collections of Web Sites, *Proceedings of UIST 2000*. *26% acceptance rate.*
111. #Amento, B., Terveen, L., and Hill, W. Does 'Authority' Mean Quality? Predicting Expert Quality Ratings of Web Documents, *Proceedings of SIGIR 2000*. *27% acceptance rate*.
112. Amento, B., Hill, W., Terveen, L., Hix, D., and Ju, P. An Empirical Evaluation of User Interfaces for Topic Management of Web Sites, *Proceedings of CHI 1999*, 552-559. *25% acceptance rate.*
113. Terveen, L.G and Hill, W.C. Evaluating Emergent Collaboration on the Web, *Proceedings of CSCW 1998*, 355-362. *19% acceptance rate.*
114. Whittaker, S., Terveen, L.G, Hill, W.C., and Cherny, L. The Dynamics of Mass Interaction, *Proceedings of CSCW 1998*, 257-264. *19% acceptance rate.*
115. Terveen, L.G and Hill, W.C. Finding and Visualizing Inter-site Clan Graphs, *Proceedings of CHI 1998*, 448-455. *23% acceptance rate.*
116. Terveen, L.G and Hill, W.C. Involving Users in Continuous Design of Web Content, in *Proceedings of DIS 1997*, ACM Press, 137-145. *41% acceptance rate*
117. Terveen, L.G., Hill, W.C., Amento, B., McDonald, D., and Creter, J. Building Task-Specific Interfaces to High Volume Conversational Data, *Proceedings of CHI 1997*, 226-233. *23% acceptance rate.*
118. Hill, W.C. and Terveen, L.G Using Frequency-of-Mention in Public Conversations for Social Filtering, *Proceedings of CSCW 1996*, 106-112.
119. Terveen, L.G. and Murray, L. Helping Users Program Their Personal Agents, *Proceedings of CHI 1996*, 355-361. *23% acceptance rate.*
120. Terveen, L.G. and Tuomenoksa, M.L. DynaDesigner: A Tool for Rapid Creation of Device-Independent, *Proceedings of INTERACT 1995*, 386-389.
121. Terveen, L.G. and Selfridge, P.G. Intelligent Assistance for Software Construction: A Case Study, *Proceedings of Knowledge-Based Software Engineering 1994*, 14-21.
122. Terveen, L.G., Selfridge, P.G., and Long, M.D. From 'Folklore' to 'Living Design Memory', in *Proceedings of INTERCHI 1993*, 15-22. *19% Acceptance Rate.*

123. Terveen, L.G. Interface Support for Data Archaeology, *ISMM International Conference on Information and Knowledge Management* (CIKM'93). *(acceptance rate for 1992 unknown; average is 24%*).

124. Brachman, R. J., Selfridge, P.G., Terveen, L.G., Altman, B., Borgida, A., Helper, F., Kirk, T., Lazar, A., McGuinness, D.L., and Resnick, L. A., Knowledge Representation Support for Data Archaeology, *ISMM International Conference on Information and Knowledge Management* (CIKM'92). *(acceptance rate for 1992 unknown; average is 24%).*

125. Selfridge, P.G., Terveen, L.G., and Long, M.D. Managing Design Knowledge to Provide Assistance to Large-Scale Software Development, *Proceedings of Knowledge-Based Software Engineering 1992*, 154-162.

126. Terveen, L.G. and Wroblewski, D.A. A Tool for Achieving Consensus in Knowledge Editing*, Proceedings of AAAI 1991*, 74-79*. 24% acceptance rate*.

127. Terveen, L.G., Wroblewski, D.A., and Tighe, S.N. Intelligent Assistance Through Collaborative Manipulation, *Proceedings of IJCAI 1991, 9-14*. (*acceptance rate for 1991 unknown; typically in low 20s*).

128. Terveen, L.G. and Wroblewski, D.A. A Collaborative Interface for Browsing and Editing Large Knowledge Bases, *Proceedings of AAAI 1990*, 491-496. *18% acceptance rate*.

Book Chapters

129. Terveen, L., Riedl, J., Konstan, J., and Lampe, C. (2014) "Study, Build, Repeat: Using Online Communities as a Research Platform", in *Human Computer Interaction Ways of Knowing*, edited by Judith S. Olson and Wendy Kellogg, New York: Springer.

130. Amento, B., Harrison, C., Nathan, M., and Terveen, L. (2009), Asynchronous Communication – Fostering Social Interaction with CollaboraTV, in Cesar, P., Geerts, D., and Chorianopoulos, K. (ed.), *Social Interactive Television: Immersive Shared Experiences and Perspectives* (2009), Information Science Reference.

131. Amento, B., Terveen, L.G and Hill, W. From PHOAKS to TopicShop: Experiments in Social Data Mining, in Lueg, C. and Fisher, D. (ed.), *From Usenet to CoWebs: Interacting with Social Information Spaces* (2002), Springer.

132. Whittaker, S., Terveen, L.G, Hill, W.C., and Cherny, L. The Dynamics of Mass Interaction, in Lueg, C. and Fisher, D. (ed.), *From Usenet to CoWebs: Interacting with Social Information Spaces* (2002), (this chapter is a reprint of Whittaker et al 1998), Springer.

133. Terveen, L.G and Hill, W. Beyond Recommender Systems: Helping People Help Each Other, in Carroll, J. (ed.), *HCI in the New Millennium* (2001), Addison Wesley.

134. Terveen, L.G. Computer-Mediated Collaboration, in *More than Screen Deep: Toward Every-Citizen Interfaces to the Nation's Information Infrastructure* (1997), National Academy Press.

Other Publications

135. Konstan, J.A., and Terveen, L.G. Human-centered recommender systems: Origins, advances, challenges, and opportunities. *AI Magazine*, 2021.

136. Terveen, L. "Autonomous" agents? What should we worry about? What should we do? *Human-Computer Interaction*.

137. Kaur, A., Smith, C. E., and Terveen, L. Sway Together, Stay Together: Visualizing Spiritual Support Networks Through the SoulGarden Prototype. In *Extended Abstracts of CSCW 2021*.

138. Kaur, H., Johnson, I., Miller, H., Terveen, L, Lampe, C., Hecht, b., and Lasecki, W. Oh The Places You'll Share: An Affordances-Based Model of Social Media Posting Behaviors. In *Extended Abstracts of CHI 2017* (ACM Conference on Human Factors in Computing Systems). 40% acceptance rate.
139. Masli, M. and Terveen, L. Geographical Social Production: Lessons from Cyclopath, in CHI 2013 GeoHCI Workshop.
140. Masli, M., Bouman, L., Owen, A., and Terveen, L. Geowiki + route analysis = improved transportation planning, CSCW 2013 Interactive Poster.
141. Brady, S. S., Sieving, R. E., Terveen, L. G., Rosser, B. R. S., Kodet, A. J., & Rothberg, V. D. (2012, October). *TeensTalkHealth: An interactive website to promote healthy relationships and prevent STIs*. Paper presented at the annual meeting of the American Public Health Association, San Francisco, CA.
142. Society for Personality and Social Psychology, San Antonio, TX. January, 2011. "The role of community orientation in promoting online participation" (with J. S. Kim, P. C. Dwyer, J. Filson Moses, P. T. Fuglestad, C. A. Mannino, R Davies, & M. Snyder)
143. Society for Personality and Social Psychology, San Antonio, TX. January, 2011. "Applying a functional approach to participation in online groups" (with P. T. Fuglestad, P. C. Dwyer, J. Filson Moses, J. S. Kim, C. A. Mannino, R Davies, & M. Snyder)
144. Society for Personality and Social Psychology, San Antonio, TX. January, 2011. "Past volunteerism predicts amount of content contributed in an online community" (with P. C. Dwyer, J. Filson Moses, P. T. Fuglestad, J. S. Kim, C. A. Mannino, R Davies, & M. Snyder)
145. Society for Personality and Social Psychology, San Antonio, TX. January, 2011. "Social motives and personality as predictors of online participation" (with J. Filson Moses, P. C. Dwyer, P. T. Fuglestad, J. S. Kim, C. A. Mannino, R Davies, & M. Snyder)
146. Kapoor, N., Frankowski, D., Konstan, J., and Terveen, L. Lessons Learned in Implementing the CHIplace Online Community, in *Proceedings of Human-Computer Interaction International 2005*.
147. Zhou, C., Ludford, P., Frankowski, D., and Terveen, L. An Experiment in Exploring How People Describe Places, Short Paper in *Proceedings of Pervasive 2005*.
148. Zhou, C., Ludford, P., Frankowski, D., and Terveen, L. An Experiment in Discovering Personally Meaningful Places from Location Data, Short Paper in *Proceedings of CHI 2005*.
149. Kapoor, N., Konstan, J., and Terveen, L. How Peer Photos Influence Member Participation in Online Communities, Short Paper in *Proceedings of CHI 2005*.
150. Amento, B., Hill, W., and Terveen, L. The Sound of One Hand: A Wrist-mounted Bio-acoustic Fingertip Gesture Interface, Short Paper in *Proceedings of CHI 2002*, 724-725. *33% acceptance rate.*
151. Terveen, L., Hill, W., and Amento, B. Collaborative Filtering to Locate, Comprehend, and Organize Collections of Websites, in SIGART Bulletin, 9, 3&4 (1998), 10-17.
152. Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. 1997. PHOAKS: a system for sharing recommendations. *Commun. ACM* 40, 3 (Mar. 1997), 59-62.
153. Terveen, L.G., Stolze, M., and Hill, W. From 'Model World' to 'Magic World', in *SIGCHI Bulletin*, 27, 4 (1995), 31-34.
154. Terveen, L.G., Papavero, E., and Tuomenoksa, M. DynaDesigner: A Tool for Rapid Design and Deployment of Device-Independent Interactive Services, Refereed Formal Demonstration in *Adjunct Proceedings of CHI'95*, 29-30.

155. Terveen, L.G. Person-Computer Cooperation through Collaborative Manipulation. Ph.D. Thesis, University of Texas Department of Computer Sciences, 1991.

Invited Presentations

Carnegie Mellon University

Northwestern University

The University of California Irvine

Carleton College

Twin Cities MetroGIS Policy Board

Hennepin County Bicycle Advisory Committee

University of Minnesota Digital Humanities Collaborative

University of Minnesota Advanced Transportation Technologies Seminar Series

Georgia Tech

University of Minnesota Urban Ecosystems Symposium

University of Minnesota New Media

IBM T.J. Watson Research

University of Illinois

University of Maryland

University of Washington

Carnegie-Mellon University

Swedish Institute of Computer Science / Royal Institute of Technology

Uppsala University, Sweden

New Jersey Institute of Technology

Microsoft Research

Vassar University

Rensselaer Polytechnic Institute

University of Colorado

Virginia Tech

University of Nebraska

Conference Presentations

- ACM Conference on Computer-Supported Cooperative Work (CSCW), 2014.
- ACM Conference on Computer-Supported Cooperative Work (CSCW), 2012.
- ACM Conference on Computer-Supported Cooperative Work (CSCW), 2002.
- ACM Conference on Computer-Supported Cooperative Work (CSCW), 1998.
- ACM Conference on Human Factors in Computing Systems (CHI), 1998.
- ACM Conference on Design of Interactive Systems, 1997.

- ACM Conference on Human Factors in Computing Systems (CHI), 1997.
- ACM Conference on Human Factors in Computing Systems (CHI), 1996.
- IFIP TC13 International Conference on Human-Computer Interaction (INTERACT), 1995.
- IEEE Conference on Knowledge-Based Software Engineering, 1994.
- ACM Conference on Human Factors in Computing Systems (CHI), 1993.
- Conference on Information and Knowledge Management, 1993.
- Conference on Information and Knowledge Management, 1992.
- National Conference on Artificial Intelligence (AAAI), 1991.
- International Joint Conference on Artificial Intelligence (IJCAI), 1991.
- National Conference on Artificial Intelligence (AAAI), 1990.

Workshop Presentations

- CHI 2013 GeoHCI Workshop
    - "Geographical Social Production: Lessons from Cyclopath"
- CSCW 2002 Workshop on "The Role of Place in Shaping Virtual Communities":
    - "Place-Based Community Information Systems".
- CHI 99 Workshop on "Interacting with Recommender Systems":
    - "Visualization Interfaces for Recommender Systems".
- 1999 Human-Computer Interaction Consortium Workshop:
    - "A Reference Task Agenda for Human-Computer Interaction".
- 1998 AAI Workshop on "Recommender Systems":
    - "The PHOAKS Recommender System".
- 1997 Human-Computer Interaction Consortium Workshop:
    - "The PHOAKS Recommender System".
- 1995 Lifelike Computer Characters Workshop:
    - "Hidden Hands, not Talking Heads: The Magic World Interaction Paradigm".
- 1995 Lifelike Computer Characters Workshop:
    - "Moving Agent-User Voice Dialogue towards Natural Conversation".
- CHI 95 Workshop on "'Model World' to 'Magic World': Making Visual Objects the Medium for Intelligent Design Assistance":
    - "The 'Magic World' Approach to Human-Computer Collaboration".
- 1993 AAAI Fall Symposium: "Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice":
    - "A Framework for Human-Computer Collaboration"
- 1993 AI-ED Workshop on "Collaborative Problem Solving: Theoretical Frameworks and Innovative Systems":

- "Collaborative Problem Solving in Interactive Systems".
- 1992 CAIA Workshop on "Applying AI To Software Problems: Assessing Promises and Pitfalls":
  - "Representing and Disseminating Software Design Knowledge".
- 1992 AAAI Spring Symposium on "Cognitive Aspects of Knowledge Acquisition":
  - "In The Footprints of The Masters: Embedding Knowledge Acquisition in Organizational Activity".
- AAAI 90 Workshop on "Complex Systems, Ethnomethodology, and Interaction Analysis":
  - "Resources for Person-Computer Collaboration".
- 1990 AAAI Spring Symposium on "Knowledge-Based Human-Computer Communication":
  - "Tools for Human-Computer Collaboration".

Patents

U.S. Patent # 5,388,188. Apparatus and methods for providing design advice. (with P. Selfridge). Issued February 7, 1995.

U.S. Patent #5,659,724. Interactive data analysis apparatus employing a knowledge base. (with A. Borgida, R.J. Brachman, T. Kirk, and P. Selfridge). Issued August 19, 1997.

U.S. Patent #5,680,530. Graphical environment for interactively specifying a target system (with P. Selfridge). Issued October 21, 1997.

U.S. Patent #5,806,060. Interactive data analysis employing a knowledge base. (with A. Borgida, R.J. Brachman, T. Kirk, and P. Selfridge). Issued September 8, 1998.

U.S. Patent # 5,809,492. Apparatus and method for defining rules for personal agents. (with L. Murray). Issued September 15, 1998.

U.S. Patent # 5,953,393. Personal Telephone Agent. (With P. Culbreth, P. Danielsen, R.J. Hall, E. Papavero, and M. Tuomenoksa). Issued September 14, 1999.

U.S. Patent #6,029,192. System and method for locating resources on a network using resource evaluations derived from electronic messages. (with W.C. Hill). Issued February 22, 2000.

U.S. Patent #6,244,873. Wireless myoelectric control apparatus and methods. (with W.C. Hill, F.C. Pereira, and Y. Singer). Issued June 12, 2001.

U.S. Patent #6,256,648. System and method for selecting and displaying hyperlinked information resources. (with W.C. Hill). Issued July 3, 2001.

U.S. Patent #9,430,043. Bioacoustic Control System, Method, and Apparatus. (with B. Amento and W.C. Hill). Issued August 30, 2016.

CONSULTING

I provide details for matters that have become public. In other cases, I provide general descriptions.

- (2023) Retained by Unified Patents. General technology area: online advertising. Writing declaration in support of inter partes review petitions.

- (2023) Retained on behalf a consumer and information technology company. General technology area: graphical and touch-based user interfaces. Writing declaration in support of inter partes review petitions.

- (2023) Retained on behalf of a computer software, hardware, and consumer electronics company. General technology areas: graphical user interfaces for web information access Writing declaration in support of inter partes review petitions.

- (2022) Retained on behalf of a computer software company. General technology areas: map-based user interfaces, mobile computing, messaging. Writing declarations in inter partes review proceedings involving patents owned by the company.

- (2022) Retained as expert witness on behalf of a computer software and hardware company. General technology area: group collaboration systems. Writing declaration in support of inter partes review petitions.

- (2022) Retained as expert witness on behalf of a consumer electronics company. General technology area: graphical user interfaces. Analyzed possible infringements of the company's patents.

- (2022) Retained as expert witness on behalf of a computer software company. General technology areas: Web-based and mobile user interfaces.

- (2022) Retained as expert witness on behalf of a computer software company. General technology areas: Web-based and mobile user interfaces.

- (2021) Retained to write declaration in support of inter partes review petitions. General technology area: information discovery.

- (2021) Retained on behalf of a computer software and hardware company. General technology area: graphical user interfaces. Writing declaration in support of inter partes review petitions.
    - **Deposed: September 2022**

- (2021) Retained on behalf of a computer software and hardware company. General technology areas: web information access, voice interaction, home automation. Writing declaration in support of inter partes review petitions.

- (2021) Retained on behalf of a computer software company. General technology area: multimedia messaging. Analyzing potential infringement of company's patents.

- (2021) Retained on behalf of a computer software company. General technology areas: map-based user interfaces, mobile computing, messaging. Writing declarations in support of patent holder's position in *ex parte* reexaminations.

- (2020) Retained on behalf of a computer software and hardware company. General technology area: information processing, hypertext, inference. Writing expert reports.

- (2020) Retained on behalf of a computer software and hardware company. General technology area: graphical user interfaces. Analyzed potential invalidity of patents.

- (2020) Retained by on behalf of a consumer electronics company. General technology area: graphical user interfaces. Analyzed possible infringements of the company's patents.

- (2020) Retained by Erise IP on behalf of Apple
    - Worked on *Inter Partes* review declaration
    - **Deposed: December 2020**

- (2019) Retained on behalf of Comcast. General technology area: interactive program guide systems. Writing expert reports.
    - **Deposed July 2021**
    - **Testified July 2022**
    - **Testified at jury trial April 2022:** Judge William Jung of the U.S. District Court for the Middle District of Florida granted judgment as a matter of law of no infringement for Comcast.

- (2019) Retained on behalf of a computer and internet software company. General technology area: information search. Did not write expert reports.

- Retained by O'Melveny & Myers on behalf of Google (2019)

- Retained by Erise IP on Behalf of Unified Patents, Inc. (2018)
    - Worked on *Inter Partes* review declaration
    - **Deposed: September 2019**

- Retained by Maynard, Cooper & Gale for an *Ex Parte* Reexamination (2018)

- Retained by Klarquist Sparkman on behalf of LinkedIn (2018)
    - Worked on an *Inter Partes* review declaration

- Retained by Klarquist Sparkman on behalf of Microsoft (2017-2018)
    - Worked on multiple *Inter Partes* review declarations and a Post Grant Review declaration

- Microsoft Corporation and Microsoft Mobile, Inc. vs. Koninklijke Philips N.V.
    - Retained as expert on behalf of Microsoft by Perkins Coie (2017-2019)
    - PTAB Case No. IPR2018-00023
    - Wrote an *Inter Partes* Review declaration
        - **Deposed: July 2018**
    - District Court Case

- **Deposed: July 2019**

- Retained by Perkins-Coie on behalf of Microsoft (vs. Cypress Lake Software)
  - Wrote a Post Grant Review declaration

- TiVo Inc. v. Samsung Electronics
  - Retained as expert on behalf of TiVo by Irell & Manella (2016)
  - Case No. 2:15-cv-1503 (E.D. Tex.)

- Netflix v. Rovi 4:11-cv-06591
  - Retained as expert on behalf of Netflix by Keker Van Nest (2014-2015)
  - **Presented technology tutorial to Judge, March 2015**

- Clear with Computers vs. Vermeer
  Clear with Computers, LLC vs. Vermeer Corporation
  Civil Action No. 13-cv-00167 (E.D. Texas)
  - Retained by Faegre, Baker, and Daniels
- Clear with Computer vs. Manitowoc Cranes, LLC.
  Clear with Computers, LLC vs. Vermeer Corporation
  Civil Action No. 13-cv-00167 and related cases (E.D. Texas)
  - Retained by Baker Botts
- Clear with Computers vs. Terex Corporation
  Clear with Computers, LLC v. Terex Corporation
  Civil Action No. 6:12-cv-00634 (E.D. Texas)
  - Retained by Fish & Richardson

- Clear with Computers vs. Valmont Industries, Inc.
  Clear with Computers, LLC v. Valmont Industries, Inc.
  Civil Action No. 6:13-cv-00166 (E.D. Texas)
  - Retained by Fish & Richardson
  - During the first half of 2014 I was retained by three different firms on behalf of four different clients to work on invalidity. I also worked on non-infringement on behalf of Vermeer.

- Intellectual Ventures vs. U.S. Bancorp and U.S. Bank
  Intellectual Ventures II LLC v. U.S. Bancorp and U.S. Bank
  Civil Action No. 13-2071 (U.S. District of Minnesota, Fourth Division)
  - Retained by Winthrop & Weinstine on behalf of U.S. Bancorp and U.S. Bank
  - Consulted on non-infringement and invalidity from late 2013 through mid 2014

- "Products Containing Interactive Program Guide and Parental Control Technology": Rovi vs. LG Electronics, Mitsubishi, Netflix, Roku, and Vizio.
  - Case No. ITC 337-TA-845
  - International Trade Commission
  - Retained as expert on behalf of respondents Netflix and Roku by Keker Van Nest (2012-2013)

- **Deposed: 2013**
- **Testified in court: 2013**

- Motorola Mobility vs. Microsoft and Microsoft vs. Motorola Mobility (Counterclaim)
  - Case No. 1:10-CV-24063-MORENO
  - United States District Court Southern District of Florida
  - Retained as expert on behalf of Microsoft (2011-2012) by Sidley Austin
  - **Deposed: 2011**

- Interval Licensing LLC vs. Aol, Inc.; Apple, Inc., eBAY, Inc.; Facebook, Inc.; Google, Inc.; Netflix, Inc.; Office Depot Inc.; OfficeMax Inc.; Staples, Inc.; Yahoo, Inc.; and YouTube, LLC.
  - Case No. 2:10-cv-01385-MJP
  - United States District Court Western District of Washington at Seattle
  - Retained as expert of behalf of four of the respondents – eBay Inc.; Netflix, Inc.; Office Depot, Inc.; and Staples, Inc. – by Klarquist Sparkman (2011)

- Microsoft vs. Tivo
  - Case No. 5:10-cv-00240-RS
  - United States District Court Northern District of California
  - Retained as expert on behalf of plaintiff by Perkins Coie (2010-2011)

- Elsevier B.V., Elsevier Inc., and Mosby, Inc. vs. UnitedHealth Group, Inc. etc.
  - 09 Civ. 2124
  - United States District Court Southern District of New York
  - Retained as expert on behalf of respondent by Dorsey & Whitney (2010)

- NORTHBROOK DIGITAL LLC vs. Vendio Services, Inc.
  - Civil File No. 07-CV-2250
  - United States District Court District of Minnesota
  - Retained as expert on behalf of plaintiff by Dorsey & Whitney (2009)

SERVICE

University of Minnesota Committees

- College of Science & Engineering Consultative Committee: 2015-2018
- College of Science & Engineering Honors & Awards Committee, 2017-2020

External Professional Activities

ACM Special Interest Group on Computer Human Interaction

- President: 2015-2018
- Executive Committee: Adjunct Chair for Awards, 2012 – 2015.

- Executive Committee: Vice President for Membership and Communication, 2009 – 2012.

Conference Chair

- CHI 2002: ACM Conference on Human Factors in Computing Systems
- IUI 1998: ACM Conference on Intelligent User Interfaces.

Program Committee Chair

- CSCW 2004: ACM Conference on Computer-Supported Cooperative Work.
- CSCW 2013: ACM Conference on Computer-Supported Cooperative Work.

Computer Supported Cooperative Work (CSCW) Steering Committee

- Chair, 2012-2014 (First elected Chair of the committee that oversees the CSCW Conference and the general CSCW and Social Computing research community.)

Awards Committee Chair

- CSCW 2008: ACM Conference on Computer-Supported Cooperative Work.

Journals Edited

- Special Issue of *Knowledge-Based Systems* on Human-Computer Collaboration, Vol. 8, No. 2-3, 1995.

Proceedings Edited

- Proceedings of the 2002 ACM Conference on Human Factors in Computing Systems (CHI 2002).
- Proceedings of the 1998 ACM Conference on Intelligent User Interfaces (IUI 1998).

Editorial Boards

- Communications of the ACM, 2009-present.
- ACM Transactions on CHI, 2000-2006.
- Knowledge-Based Systems, 1993-2009.
- ACM *intelligence*, 1998-2001.

Program Committees

- ACM Conference on Human Factors in Computing Systems (CHI): 1999-2004, 2006, 2016.
- ACM Conference on Computer-Supported Cooperative Work (CSCW): 2000, 2006, 2008, 2010, 2011, 2012, 2015.
- ACM Conference on Intelligent User Interfaces (IUI): 1997-2000, 2004.
- ACM Recommender Systems Conference: 2007, 2008, 2009, 2010, 2013.
- ACM GROUP Conference: 2007, 2009, 2010.

- ACM SIGIR Conference: 2008.
- AAAI Conference on the Web and Social Media: 2011, 2012, 2016.
- User Modeling, Adaptation, and Personalization: 2011, 2016.
- NordiCHI 2008, 2010.
- Communities and Technology: 2009.
- Computer-Supported Cooperative Learning 2002.
- User Modeling: 2001.
- International Conference on Knowledge Capture 2001.
- Knowledge-Based Software Engineering (KBSE): 1994-1999.
- Intelligent Data Analysis: 1997.
- National Conference on Artificial Intelligence (AAAI): 1996-1997.
- IEEE Conference on Artificial Intelligence for Applications (CAIA): 1994.
- International Workshop on Privacy-Aware Location-based Mobile Services: 2007.
- SIGIR Workshop on Future Challenges in Expertise Retrieval: 2008.

Other conference leadership positions

- ACM Conference on Computer-Supported Cooperative Work Doctoral Consortium Co-Chair, 2016 and 2011.
- The 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2014) Doctoral Consortium Co-Chair.
- International Symposium on Wikis and Open Collaboration Doctoral Symposium Chair, 2011.
- ACM Conference on Recommender Systems Doctoral Consortium, Co-Chair, 2007.
- International Joint Conference for Artificial Intelligence Doctoral Consortium, Co-Chair, 1997.
- American Association for Artificial Intelligence Doctoral Consortium, Co-Chair, 1997.
- American Association for Artificial Intelligence Doctoral Consortium, Co-Chair, 1996.

Reviewer

Journals

- ACM Computing Surveys.
- IEEE Transactions on Data and Knowledge Engineering.
- IEEE Expert.
- Information Systems.
- International Journal of Human-Computer Studies.
- Journal of Computer-Supported Cooperative Work.

Conferences

- Ninth IFIP International Conference on Human-Computer Interaction (INTERACT): 2003.
- ACM Conference on Human Factors in Computing Systems (CHI): 1995-1998.
- ACM Conference on Computer-Supported Cooperative Work (CSCW):1998.
- ACM Symposium on User Interface Software and Technology (UIST): 1996.
- ISSM Conference on Information and Knowledge Management: 1993.

National Science Foundations Panels

Served on panels in 1996, 1998, 2001, 2003, 2005, 2010, 2012, 2013.

Workshops Organized

- ACM Wikis and Open Collaboration Doctoral Symposium, 2011.
- Social Computational Systems Community Workshop, 2011.
- ACM Computer Supported Cooperative Work Doctoral Consortium, 2011.
- International Conference on Ubiquitous Computing (UbiComp): "Multi-device Interfaces for Ubiquitous Peripheral Interaction", 2003.
- ACM Conference on Human Factors in Computer Systems (CHI): "Interacting with Recommender Systems", 1999.
- International Joint Conference for Artificial Intelligence Doctoral Consortium, 1997.
- American Association for Artificial Intelligence Doctoral Consortium, 1997.
- American Association for Artificial Intelligence Doctoral Consortium, 1996.
- ACM Conference on Human Factors in Computer Systems (CHI): "'Model World' to 'Magic World': Making Visual Objects the Medium for Intelligent Design Assistance", 1995.
- ACM Conference on Human Factors in Computer Systems (CHI): "New Uses and Abuses of Interaction History", 1994.
- AAAI Fall Symposium: "Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice", 1993.
- World Conference on Artificial Intelligence and Education: "Collaborative Problem Solving: Theoretical Frameworks and Innovative Systems", 1993.
- Conference on AI for Applications: "Applying AI To Software Problems: Assessing Promises and Pitfalls", 1992.

Other Professional Service

- Member of SIGCHI Publications Board and Conference Management Committee, 2002-2004.
- ACM Special Interest Group on Artificial Intelligence Conference Chair (1995-1999); originated and co-organized Doctoral Consortia held in conjunction with AAAI and IJCAI.

Internal Service

- Associate Department Head, 2019-
- Chair, Department Head Search Committee, 2015.
- Chair, Strategic Planning Committee, 2013-2014.
- Chair, Social Computing Faculty Search Committee, 2013-2014.
- Curriculum Committee, 2012-2013.
- Chair, Strategic Planning Committee / Faculty Recruiting Committee, 2011-2012.
- Director of Graduate Studies: 2007-2010.
- Communications Committee: 2007
- Newsletter/Brochure Committee (chair): 2006-2007
- Research Opportunities Committee: 2005-2007
- Curriculum Committee: 2003-2005.
- Hosted Robert Kraut, Cray Colloquium speaker: October 2003.
- Participated in recruiting and admission activities.
- Information, Technology, and Everyday Life Initiative: Committee Member.

TEACHING AND ADVISING

Ph.D. Students Advised

- Dan Cosley (co-advised with John Riedl): completed PhD July 2006, currently an Associate Professor at Cornell University; on leave as Program Director at NSF.
- Pamela Ludford: completed PhD September 2007, currently an independent consultant.
- Reid Priedhorksy: completed PhD August 2010, currently employed at Los Alamos National Laboratories.
- Colin Delong: completed PhD October 2013 (co-advised with Jaideep Srivastava)
- Mikhil Masli: completed PhD July 2013, currently employed at IBM.
- Aaron Halfaker (co-advised with John Riedl): completed PhD September 2013, currently employed at Microsoft.
- Katie Panciera: completed PhD August 2014; currently Assistant Professor at Milwaukee School of Engineering.
- Fernando Torre: completed PhD September 2014; founder of a startup.
- Shuo (Steven) Chang: completed PhD August 2016; currently employed at Facebook.
- Tien Nguyen (co-advised with Joe Konstan): completed PhD August 2016; currently employed at Facebook.
- Morten Warncke-Wang (co-advised with Brent Hecht): completed PhD December 2016, currently employed at the Wikimedia Foundation.
- Andrew Sheppard: completed PhD December 2017, currently employed at Houston

Engineering.

- Jacob Thebault-Spieker (co-advised with Brent Hecht): completed PhD December 2017; currently Assistant Professor at The University of Wisconsin.
- Hannah Miller (co-advised with Brent Hecht): completed PhD August 2018; currently Assistant Professor at University of Wisconsin Oshkosh.
- Vikas Kumar (co-advised with Joe Konstan), completed PhD August 2018, currently employed at KenSci.
- Andrew Hall: completed PhD August 2019; currently employed at 3M.
- Bowen Yu (co-advised with Haiyi Zhu), completed PhD December 20918, currently employed at Facebook.
- C. Estelle Smith: completed PhD December 2020; currently Assistant Professor at the Colorado School of Mines.
- Zach Levonian (co-advised with Lana Yarosh): completed PhD July 2022.
- Charles Chuankai Zhang (entered program Fall 2019).
- Mohammed (Mo) Houtti (entered program Fall 2019)
- Moyan Zhou (entered program Fall 2021)
- Ava Bartolome (entered program Fall 2022; co-advised with Stevie Chancellor)

M.S. Students Advised

- Tyler Danielsen – received degree in 2016
- Jie Kang – received degree in 2016
- Zahra Eslami – received degree in 2015
- Yanjie Liu – received degree in 2013
- Renji Yu – received degree in 2012
- Carol Drysdale – received degree in 2011
- Jingwen Zhang – received degree in 2011
- Jisu Oh – received degree in 2010
- Sara Drenner – received degree in 2008
- Anna Rouben – received degree in 2006
- Arjun Sundararajan – received degree in 2006
- Pamela Ludford – received degree in 2005
- Rahul Akolkar – graduated May 2004

Undergraduate Honors / Senior Thesis Students Advised

- Moyan Zhou 2021
- Avleen Kaur 2020

- Harmanprett Kaur, 2016
- Arlo Siemsen, 2014
- Johnathan Frenz, 2013
- David Pitchford - 2012
- Michael Ludwig - 2010
- Jordan Focht – 2010
- Kurt Wilms - 2005
- John Murphy – 2004

Other Committees

- Hanlin Li, PhD: Member of final examination committee (Northwestern University).
- Matheus Araújo, PhD: Member of preliminary and final examination committees
- Hao-Fei Cheng, PhD: Member of preliminary and final examination committees
- Catherine Grevet, PhD: member of preliminary and final examination committees (Gerogia Tech).
- Loxley Wang, PhD: member of preliminary and final examination committees.
- Abigail Bakke, PhD: member of preliminary and final examination committees
- Tahir Sousa, MS: member of final examination committee
- Tony Lam, PhD: member of preliminary examination committee.
- Michael Janseen, PhD: member of preliminary examination committee.
- Haleh Hagh Shenas, PhD: member of preliminary & final examination committees.
- Julie Beilfuss, MS: member of final examination committee.
- Liv Knatterud, MS: member of final examination committee.
- Sean McNee, MS: member of final examination committee.
- Shankar Subrahmanian, MS: member of final examination committee.
- Vamsee Venuturumilli, MS: member of final examination committee.
- Eric Gilbert, Ph.D. Preliminary and Final Examination Committees (UIUC, 2009/2010): member of thesis committee.
- Yi (Jenny) Zhang, Ph.D. (New Jersey Institute of Technology, 2004): member of thesis committee.
- Brian Amento, Ph.D. (Virginia Tech, 2001): member of thesis committee.
- David McDonald, Ph.D. (UC Irvine, 2000): member of thesis committee.

Courses taught

A list of all the classes I have taught, in most cases multiple times.

- CSCI 8115: Human-Computer Interaction and UI Technology

- CSCI 5125: Collaborative and Social Computing
- CSCI 5116: GUI Toolkits and Their Implementation
- CSCI 5115: User Interface Design, Implementation, and Evaluation
- HSEM 2519H: Honors Seminar on Crowdsourcing
- CSCI 1902: Structure of Computer Programming II
- CSCI 1901H: Introduction to Computer Science (Honors)

Tutorials

"Intelligent User Interfaces: Issues, Approaches, Evaluation", offered at 1993 Conference on Artificial Intelligence for Applications.