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. 7,370,035
Issued: May 6, 2008
Application No. 10/654,588
Filed: September 3, 2003

Title: METHODS AND SYSTEMS FOR SEARCH INDEXING

# DECLARATION OF DELAINEY TANNER

MICROSOFT
EXHIBIT 1038

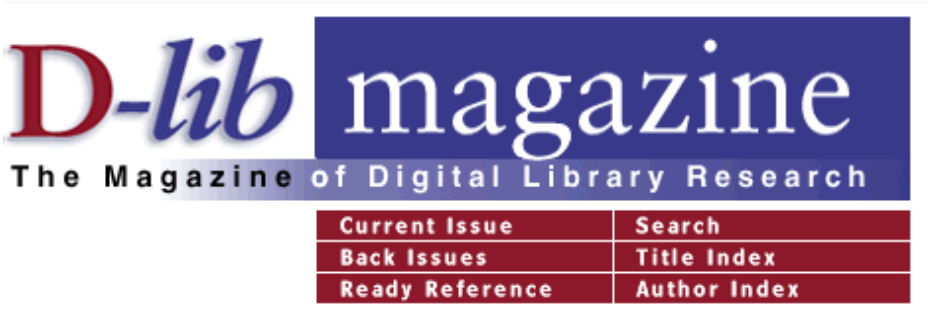
I, Delainey Tanner, hereby declare:

1. I am a Litigation Project Assistant with the law firm of Klarquist Sparkman, LLP, counsel for Petitioner Microsoft Corporation. I make this declaration in support of Microsoft's request for *inter partes* review of U.S. Patent No. 7,370,035.

2. I have personal knowledge of the matters described herein unless otherwise indicated, and if called to testify, I could and would testify competently thereto.

3. The facts set forth in this declaration are known to me personally or are based upon the information I obtained on the web.archive.org and other websites referenced below, which facts I believe to be true.

4. Attached as Appendix A to this Declaration is a printout of www.dlib.org, as captured by the Internet Archive (web.archive.org) on October 10, 1997, available at https://web.archive.org/web/19971010042226/https:/www.dlib.org/. I captured Appendix A on November 1, 2024.




**Appendix A excerpt of the "Search" and "Title Index" navigation links available at https://web.archive.org/web/19971010042226/https:/www.dlib.org/**

5. As shown in the above excerpt, the www.dlib.org webpage (Appendix A) includes navigation links to "Search," and "Title Index." The "Search" link, as captured by the Internet Archive on October 10, 1997, is available at https://web.archive.org/web/19971010042304/http:/www.dlib.org/Architext/AT-dlib2query.html, and hereby attached as Appendix B to this Declaration. The "Title Index" link, as captured by the Internet Archive on October 10, 1997, is available at https://web.archive.org/web/19971010042316/http:/www.dlib.org/title-index.html, and hereby attached as Appendix C to this Declaration. I captured both Appendix B and C on November 1, 2024.

6. Attached as Appendix D to this Declaration is a printout of the article *Clarifying Search: A User-Interface Framework for Text Searches* by Ben Shneiderman from the January 1997 issue of D-Lib Magazine. The link to

this article, as captured by the Internet Archive on October 10, 1997, is available at https://web.archive.org/web/19971010043650/http:/www.dlib.org/dlib/january97/retrieval/01shneiderman.html. I captured Appendix D on November 6, 2024.

7. Attached as Appendix E to this Declaration is a printout of the D-Lib Magazine summary page for the January 1997 issue. This webpage, as captured by the Internet Archive on October 10, 1997, is available at https://web.archive.org/web/19971010042806/http:/www.dlib.org/dlib/january97/01contents.html. I captured Appendix E on November 1, 2024.

8. Attached as Appendix F to this Declaration is a printout of www.eudora.com. This webpage, as captured by the Internet Archive on December 4, 2000, is available at https://web.archive.org/web/20001204092900/http:/www.eudora.com/. I captured Appendix F on November 1, 2024.




**The Best Email Software Just Got Better and - It's Free!**

Email traffic is on the rise. Eudora is the best email program for people who get lots of email. If you're not one of those people today, you will be soon!

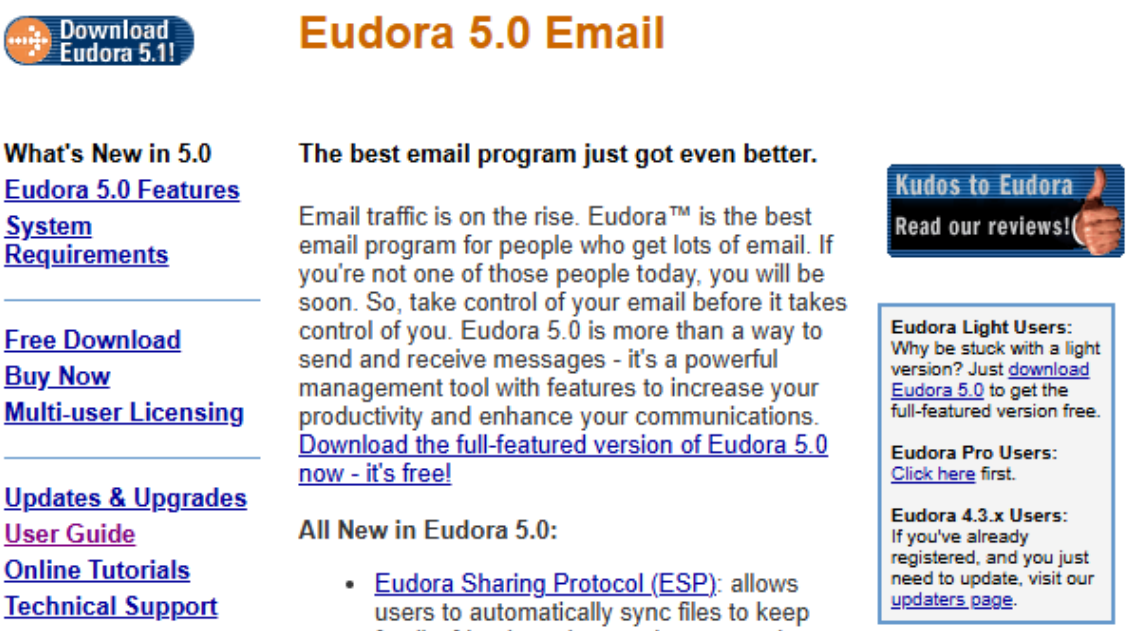
Learn More

So take control of your email before it takes control of you. Eudora 5.0 is more than a way to send and receive messages - it's a powerful management tool for your email and it's free!

Download Now

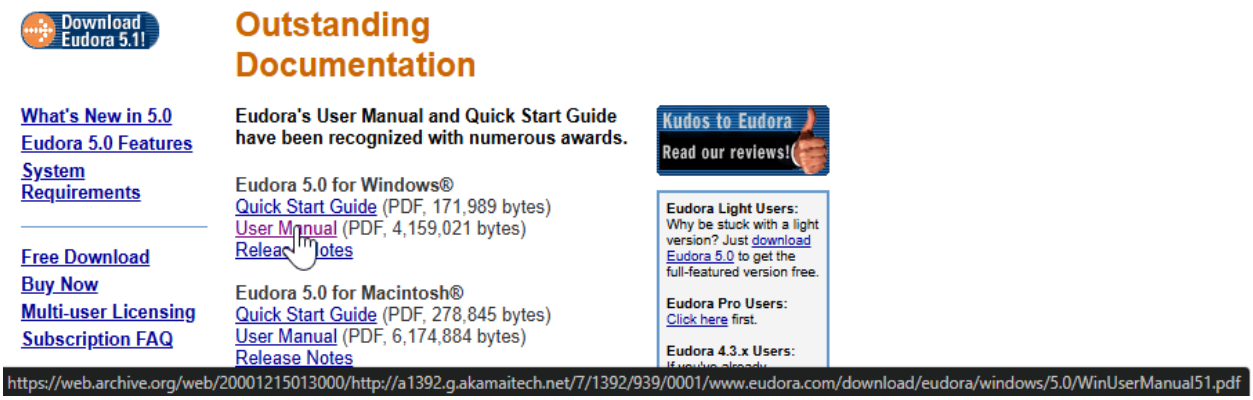
**Appendix F excerpt of the "Learn More" navigation link available at https://web.archive.org/web/20001204092900/http:/www.eudora.com/**

9. As shown in the above excerpt of the www.eudora.com webpage (Appendix F), the page includes a navigation link titled "Learn More." This link leads to the *Eudora 5.0 Email* webpage, as captured by the Internet Archive on December 6, 2000, available at https://web.archive.org/web/20001206023800/http:/www.eudora.com/email/index.html and attached as Appendix G to this Declaration. I captured Appendix G on November 1, 2024.




**Appendix G excerpt of the "User Guide" navigation link available at https://web.archive.org/web/20001206023800/http:/www.eudora.com/email/index.html**

10. As shown in the above excerpt of the *Eudora 5.0 Email* webpage (Appendix G), the page includes a navigation link titled "User Guide." This link leads to the *Outstanding Documentation* webpage, as captured by the Internet Archive on December 15, 2000, available at https://web.archive.org/web/20001215013000/http://www.eudora.com/email/docs/index.html and attached as Appendix H to this Declaration. I captured Appendix H on November 1, 2024.

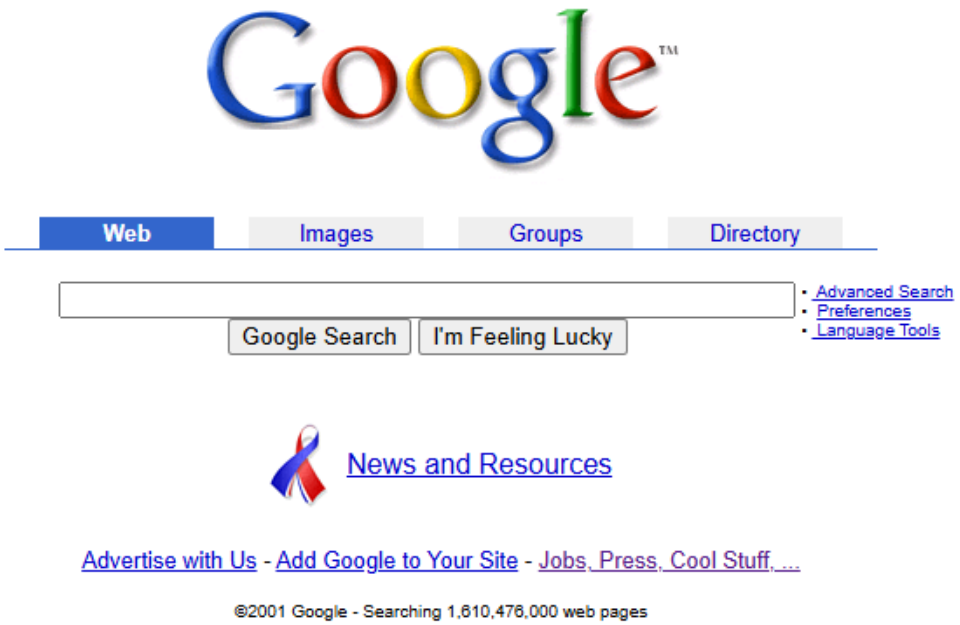



**Appendix H excerpt of the "User Manual" navigation link available at https://web.archive.org/web/20001215013000/http:/www.eudora.com/email/docs/index.html**

11. As shown in the above excerpt of the *Outstanding Documentation* webpage (Appendix H), this page includes a link titled "User Manual." Hovering the cursor over the "User Manual" link shows the URL https://web.archive.org/web/20001215013000/http://a1392.g.akamaitech.net/7/1392/939/0001/www.eudora.com/download/eudora/windows/5.0/WinUserManual51.pdf as can be seen at the bottom of the screenshot above. Clicking on this link leads to a PDF document titled *Eudora Email User Manual for Windows Version 5.0*. A copy of this PDF document is attached to the request for *inter partes* review as EX1007 ("Eudora").

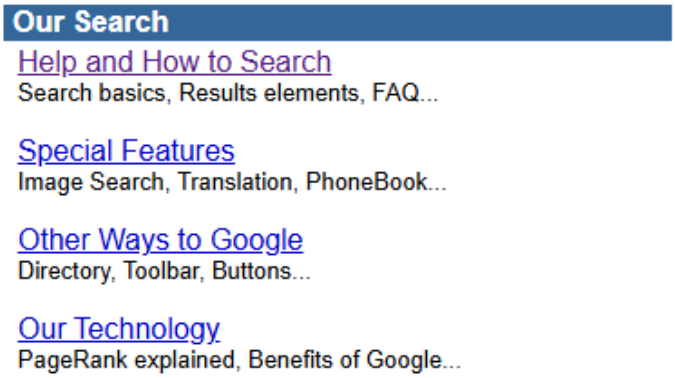
12. Attached as Appendix I to this Declaration is a printout of www.google.com, as captured by the Internet Archive on October 2, 2001, available at https://web.archive.org/web/20011002213742/http:/www.google.com/index.html.

I captured Appendix I on November 1, 2024.



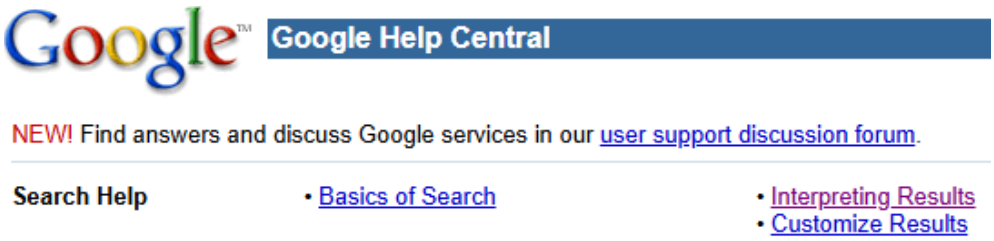

**Appendix I excerpt of the "Jobs, Press, Cool Stuff, …" navigation link available at https://web.archive.org/web/20011002213742/http:/www.google.com/index.html**

13. As shown in the above excerpt from Appendix I, this webpage includes a navigation link titled "Jobs, Press, Cool Stuff, …". The link leads to the *All About Google* webpage, as captured by the Internet Archive on October 2, 2001, available at https://web.archive.org/web/20011002213453/http:/www.google.com/about.html and attached as Appendix J to this Declaration. I captured Appendix J on November 1, 2024




**Appendix J excerpt of the "Help and How to Search" navigation link available at https://web.archive.org/web/20011022 13453/http:/www.google.com/about.html**

14. As shown in the above excerpt of the *All About Google* webpage (Appendix J), the webpage includes a navigation link titled "Help and How to Search." This link leads to the *Google Help Central* webpage, as captured by the Internet Archive on October 2, 2001, available at https://web.archive.org/web/20011002221357/http:/www.google.com/help/index.html and attached as Appendix K to this Declaration. I captured Appendix K on November 1, 2024.




**Appendix K excerpt of the "Interpreting Results" navigation link available at https://web.archive.org/web/20011002221357/http:/www.google.com /help/index.html**

15. As shown in the above excerpt of the *Google Help Central* webpage

(Appendix K), the webpage includes a navigation link titled "Interpreting Results." This link leads to the *How to Interpret your Search Results* webpage, as captured by the Internet Archive on October 3, 2001, available at https://web.archive.org/web/20011003013526/http:/www.google.com:80/help/interpret.html and attached as Appendix L to this Declaration. I captured Appendix L on November 6, 2024.

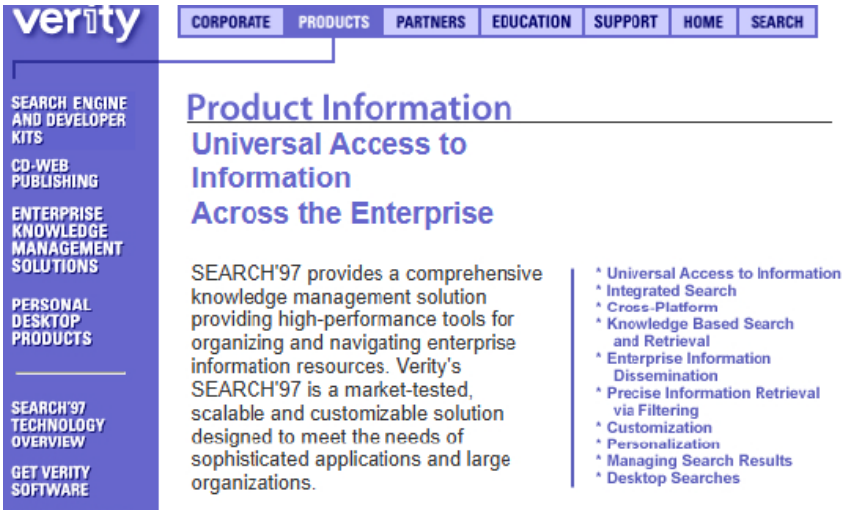
16. Attached as Appendix M to this Declaration is a printout of the www.verity.com webpage, as captured by the Internet Archive on October 16, 1997, available at https://web.archive.org/web/19971016004801/http:/www.verity.com/index.html. I captured Appendix M on November 1, 2024.




**Appendix M excerpt of the "Products" tab available at https://web.archive.org/web/19971016004801/http:/www.verity.com/index.html**

17. As shown in the above excerpt, the www.verity.com webpage (Appendix M) includes a tab titled "Products." This tab leads to the *Product Information* page, as captured by the Internet Archive on October 16, 1997,

available at https://web.archive.org/web/19971016001242/http://www.verity.com/products/prd.html and attached as Appendix N to this Declaration. I captured Appendix N on November 1, 2024.




**Appendix N excerpt of the "GET VERITY SOFTWARE" navigation link available at https://web.archive.org/web/19971016001242/http:/www.verity.com/products/prd.html**

18. As shown in the lower left corner of the above excerpt from the *Product Information* webpage (Appendix N), the page includes a navigation link titled "GET VERITY SOFTWARE." This link leads to a subpage of the *Product Information* webpage, as captured by the Internet Archive on October 16, 1997, available at https://web.archive.org/web/19971016003821/http://www.verity.com/products/software.html and attached as Appendix O to this Declaration. I captured

Appendix O on November 1, 2024.




**SEARCH'97 Personal V 1.1 for *Netscape Navigator* and *Microsoft Internet Explorer***
Platform: Win95/NT
Purchase Now!
Available in Download or CD-ROM version
View the Product Datasheet

**Appendix O excerpt of the "Product Datasheet" tab available at https://web.archive.org/web/19971016003821/http:/www.verity.com/products/software.html**

19. As shown in the above excerpt from Appendix O, it includes a navigation link titled "Product Datasheet." This link leads to the SEARCH'97 Personal Data Sheet, as captured by the Internet Archive on October 16, 1997, available at https://web.archive.org/web/19971016032849/http:/www.verity.com/products/datasheets/personal.html and attached as Appendix P to this Declaration. I captured Appendix P on November 1, 2024.




**Appendix P excerpt of the "PDF" link available at https://web.archive.org/web/19971016032849/http:/www.verity.com/products/datasheets/personal.html**

20. As shown in the above excerpt of the SEARCH'97 Personal Data Sheet webpage (Appendix P), the webpage includes a link to a PDF version of the Datasheet where it can be viewed, downloaded, or printed. A download of this document is attached as Appendix Q to this Declaration. Appendix Q, as captured by the Internet Archive on November 15, 1997, available at https://web.archive.org/web/19971115160249/http:/www.verity.com/products/PDF/s97perso.pdf. I captured Appendix Q on November 1, 2024.

21. Attached as Appendix R to this Declaration is a PDF copy of the Search'97 Personal Data Sheet downloaded from the Internet Archive PDF viewer (Appendix Q), available at https://web.archive.org/web/19971115160249/http:/www.verity.com/products/PDF/s97perso.pdf. I downloaded Appendix R on November 6, 2024.

22. Attached as Appendix S to this Declaration is a printout of www.peachpit.com, as captured by the Internet Archive on January 18, 2001,

available at https://web.archive.org/web/20010118212400/http:/www.peachpit.com/. I captured Appendix S on November 1, 2024.

23. Attached as Appendix T to this Declaration is a printout of the *Books A-Z* webpage from www.peachpit.com, as captured by the Internet Archive on January 24, 2001, available at https://web.archive.org/web/20010124175100/http://www2.peachpit.com/books/titles_a-z.html. I captured Appendix T on November 1, 2024.



E

Elements of Web Design, 2nd Edition, Darcy DiNucci with Maria Giudice and Lynne Stiles

Soon! Entourage 2001 for Macintosh: Visual QuickStart Guide, Steve Schwartz

Eudora 4.2 for Windows and Macintosh: Visual QuickStart Guide, Adam Engst

Eudora for Windows and Macintosh: Visual QuickStart Guide, Adam Engst

Excel 2000 for Windows: Visual QuickStart Guide, Maria Langer

Soon! Excel 2001 for Macintosh: Visual QuickStart Guide, Maria Langer

Excel 98 for Macintosh: Visual QuickStart Guide, Maria Langer

**Appendix T excerpt of the "Entourage 2001 for Macintosh: Visual QuickStart Guide" link available at https://web.archive.org/web /20010124175100/http:/www2.peachpit.com/books/titles_a-z.html**

24. As shown in the above excerpt of the *Books A-Z* webpage (Appendix T), the page includes a navigation link titled "Entourage 2001 for Macintosh: Visual QuickStart Guide." This link leads to an overview page for the *Entourage 2001 for Macintosh: Visual QuickStart Guide* book attached as Appendix U to

this Declaration, as captured by the Internet Archive on January 29, 2001, available at https://web.archive.org/web/20010129021800/http:/www2.peachpit.com:80/books/catalog/73398.html. I captured Appendix U on November 1, 2024.

25. Attached as Appendix V to this Declaration is a printout of the www.oreilly.com webpage, as captured by the Internet Archive on May 9, 2001, available at https://web.archive.org/web/20010509183427/http:/www.oreilly.com:80/. I captured Appendix V on November 1, 2024.




**Appendix V excerpt of the "Product List" and "Office 2001 for Macintosh: The Missing Manual" links available at https://web.archive.org/web/20010509183427/http:/www.oreilly.com:80/**

26. As shown in the above excerpt from www.oreilly.com (Appendix

V), the webpage includes navigation links titled "Product List" and "Office 2001 for Macintosh: The Missing Manual." The "Products List" link leads to the webpage attached as Appendix W, as captured by the Internet Archive on April 13, 2001, available at, https://web.archive.org/web/20010413112343/http:/www.oreilly.com/cgi-bin/productindex. The "Office 2001 for Macintosh: The Missing Manual" link leads to the webpage attached as Appendix X, as captured by the Internet Archive on June 4, 2001, available at https://web.archive.org/web/20010604052859/http:/www.oreilly.com/catalog/msoffmacmm/. I captured Appendix W and Appendix X on November 1, 2024.

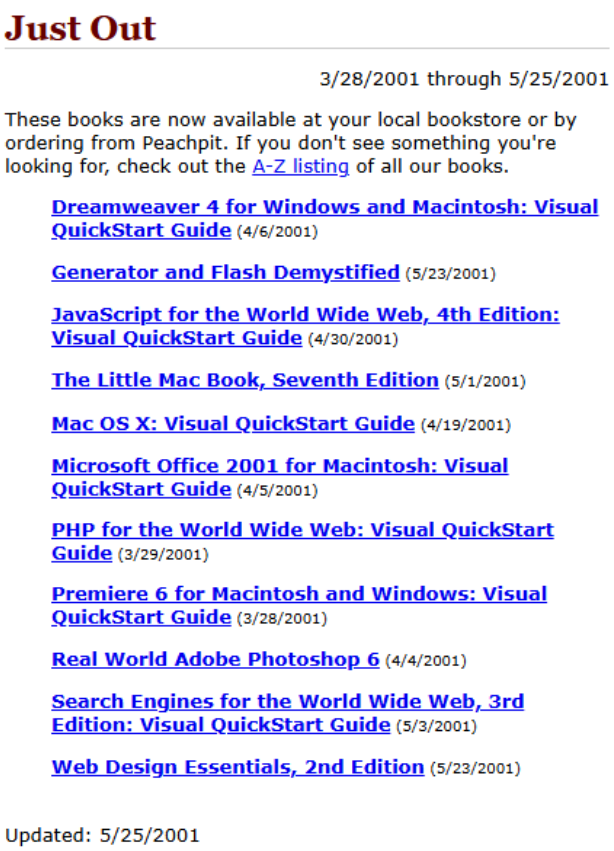
27. Attached as Appendix Y to this Declaration is a printout of the www.pearson.com overview webpage, as captured by the Internet Archive on February 3, 2001, available at https://web.archive.org/web/20010203182800/http:/www.pearson.com/overview/ped.htm. I captured Appendix Y on November 6, 2024.

28. Attached as Appendix Z to this Declaration is a printout of a webpage from www.amazon.com showing a 1st edition book for purchase authored by Nan Barber and David Reynolds titled *Office 2001 for Macintosh: The Missing Manual*. I captured Appendix Z on November 1, 2024, at https://www.amazon.com/Office-2001-Macintosh-Missing-

Manual/dp/0596000812.

29. Attached as Appendix AA to this Declaration is a printout of the www.peachpit.com "Books" webpage showing the "Coming Soon" section, as captured by the Internet Archive on April 9, 2001, available at https://web.archive.org/web/20010409204844/http:/www.peachpit.com/books/just-soon.html. I captured Appendix AA on November 18, 2024.

30. Attached as Appendix BB to this Declaration is a printout of the www.peachpit.com "Books" webpage showing the "Just Out" section, as captured by the Internet Archive on June 4, 2001, available at https://web.archive.org/web/20010604024604/http:/www.peachpit.com:80/books/just-soon.html. I captured Appendix BB on November 18, 2024.



## Just Out

3/28/2001 through 5/25/2001

These books are now available at your local bookstore or by ordering from Peachpit. If you don't see something you're looking for, check out the A-Z listing of all our books.

**Dreamweaver 4 for Windows and Macintosh: Visual QuickStart Guide** (4/6/2001)

**Generator and Flash Demystified** (5/23/2001)

**JavaScript for the World Wide Web, 4th Edition: Visual QuickStart Guide** (4/30/2001)

**The Little Mac Book, Seventh Edition** (5/1/2001)

**Mac OS X: Visual QuickStart Guide** (4/19/2001)

**Microsoft Office 2001 for Macintosh: Visual QuickStart Guide** (4/5/2001)

**PHP for the World Wide Web: Visual QuickStart Guide** (3/29/2001)

**Premiere 6 for Macintosh and Windows: Visual QuickStart Guide** (3/28/2001)

**Real World Adobe Photoshop 6** (4/4/2001)

**Search Engines for the World Wide Web, 3rd Edition: Visual QuickStart Guide** (5/3/2001)

**Web Design Essentials, 2nd Edition** (5/23/2001)

Updated: 5/25/2001

**Appendix BB excerpt of the "Search Engines for the World Wide Web, 3rd Edition: Visual QuickStart Guide" link available at https://web.archive.org/web/20010604024604/http:/www.peachpit.com :80/books/just-soon.html**

31. As shown in the above excerpt from www.peachpit.com (Appendix BB), the webpage includes a navigation link titled "Search Engines for the Word Wide Web, 3rd Edition: Visual QuickStart Guide." This link leads to the webpage attached as Appendix CC, as captured by the Internet Archive on April 18, 2001, available at https://web.archive.org/web/20010418101001/http://www.peachpit.com/books/catalog/73401.html. I captured Appendix CC on November 18, 2024.

I declare under penalty of perjury that the foregoing is true to the best of my information, knowledge and belief.

Executed this 18th day of November 2024 at Portland, Oregon.

Delainey Tanner

Delainey Tanner

# APPENDIX A

https://web.archive.org/web/19971010042226/https:/www.dlib.org/ November 1, 2024 at 11:11 AM PDT

https://www.dlib.org/ Go JUL OCT FEB
2,297 captures 10
19 Jul 1997 - 9 Oct 2024 1996 1997 1998
About this capture

# D-lib magazine

The Magazine of Digital Library Research



Welcome to D-Lib Magazine, a single site with monthly stories, commentary, and briefings and a collection of resources for digital library research. Visit D-Lib Magazine this month for new material; visit Ready Reference for pointers to projects and collections concerning digital library research.

**Current Issue:** September 1997

**D-Lib Ready Reference**

---

**Mirror sites for D-Lib Magazine are graciously maintained by:**

UKOLN: The UK Office for Library and Information Networking

The Australian National University Sunsite

---

These activities are coordinated by the Corporation for National Research Initiatives and are sponsored by the Defense Advanced Research Projects Agency (DARPA) on behalf of the NSF/DARPA/NASA Digital Libraries Initiative.

---

*Graphics on D-Lib's pages have been created by Robert Kinneary and Catherine Rey.*
*wya/af*
*Last revised: September 15, 1997*

# APPENDIX B

https://web.archive.org/web/19971010042304/http:/www.dlib.org/Architext/AT-dlib2query.html November 1, 2024 at 11:11 AM PDT

http://www.dlib.org/Architext/AT-dlib2query.html Go SEP OCT FEB
310 captures 10 Oct 1997 - 27 Sep 2024 1996 10 1997 1998 About this capture

D-lib magazine
The Magazine of Digital Library Research

# Search D-Lib Magazine

**Welcome to the D-Lib Magazine search page, a tool for locating stories, commentary, briefings, and resources for digital library research that we have identified. Help with formulating queries is available.**

**You may also browse the classification structure of the Ready Reference pages or the contents of back issues of the monthly magazine by title, author, or contents pages.**

**Enter ◉ words describing a concept or ○ keywords you wish to find information about:**

excite Search

Home Magazine

***Search engine by excite. Last updated September 15, 1997.***

# APPENDIX C

https://web.archive.org/web/19971010042316/http:/www.dlib.org/title-index.html November 1, 2024 at 11:12 AM PDT
http://www.dlib.org/title-index.html Go SEP OCT FEB
359 captures 10 Oct 1997 - 27 Sep 2024 1996 10 1997 1998 About this capture

D-lib magazine
The Magazine of Digital Library Research

ISSN 1082-9873

## Access to Stories, Briefings, and Editorials by Title

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

*You may also browse by author.*

The 4th Dublin Core Metadata Workshop Report:, Stuart Weibel, Renato Iannella, Warwick Cathro
1990 Census LOOKUP: Mining a Mountain of Data, Deane W. Merrill, Nathan G. Parker, Harvard H. Holmes, Chris Stuber, Valerie J. Gregg

A

Access and Discovery: Issues and Choices in Designing DIFWICS, Jeremy Hylton
Access to Digital Objects: A Communications Law Strategy, Patrice A. Lyons
Accessing the Visible Human Project, Michael J. Ackerman
Across Languages, Across Cultures: Issues in Multilinguality and Digital Libraries, Carol Peters, Eugenio Picchi
Ad- Hoc Classification of Electronic Clinical Documents, David B. Aronow, Fangfang Feng
Adapting Digital Libraries for Learners: Accessibility vs. Availability, Daniel C. Edelson, Douglas N. Gordin
Advanced Web Presentation through Data Modeling: An Open Architecture for the Personalized Webs of the Future, Leon Shklar
Agent-based Architecture for Digital Libraries, William P. Birmingham
Alexandria Digital Library Testbed, James Frew, Michael Freeston, Randy Kemp, Jason Simpson, Terence Smith, Alex Wells, Qi Zheng
An Architecture for Information in Digital Libraries, William Y. Arms, Christophe Blanchi, Edward A. Overly
Australian Digital Library Initiatives , Renato Iannella (Editor)

B

Berkeley Digital Library SunSITE, Roy Tennant
Beyond Word and Image: Networking Moving Images -- More Than Just the "Movies"
*The first of a two-part examination of networking cultural heritage materials*
David Green

*Briefings:*

- ARTFL, Andrea Doane
- Brief Update on the Alexandria Digital Library Project: Constructing a Digital Library for Geographically- Referenced Materials, Terence R. Smith
- CyberStac ks(sm): A 'Library-Organized' Virtual Science and Technology Reference Collection, Gerry McKiernan
- Digital Libraries in the Classroom, Elliott Soloway
- EduPort< /A>, Miriam Masullo
- Highlights related to the Government Information Locator Service (GILS): Toward a Global Information Locator, Eliot Christian
- Integrated Document Access (IDA) Project, Margaret Colmer
- Making a Digital Library: The Chemistry Online Retrieval Experiment -- A Summary of the CORE Project (1991-1995), Richard Entlich, Lorrin Garson, Michael Lesk, Lorraine Normore, Jan Olsen, Stuart Weibel
- OCLC Internet Cataloging Project, Erik Jul
- Oregon State University's Government Information Sharing Project, Jacquelyn Miller
- Project Muse: 43 Humanities and Social Sciences Journals to Come on the Network
- SuperJourn al, David Pullinger, Christine Baldwin
- TURNIP: The URN Interoperability Project, Renato Iannella
- UK Electronic Libraries Programme, Chris Rusbridge
- Z39.50 and the World Wide Web, Sebastian Hammer, John Favaro

C

Caching for Large Scale Systems, Lessons from the WWW, Robert E. McGrath
The Cigarette Papers: Issues in Publishing Materials in Multiple Formats, Karen Butter, Robin Chandler, and John Kunze
Clarifying Search: A User-Interface Framework for Text, Ben Shneiderman, Don Byrd, and W. Bruce Croft

*Clips:*

- Berkeley Digital Library Technology Assists in California Flood Recovery Efforts, Robert Wilensky
- DELOS Cross-Language Workshop: Summary, Paraic Sheridan
- First European Conference on Research and Advanced Technology for Digital Libraries, Carol Peters
- Journal of Internet Cataloging, Gerry McKiernan
- NSF Awards $3 million to the Internet Scout Project; Expansion Planned, Amy Tracy Wells
- ISO Working Group for Transliteration from Indic to Latin script, Anthony P. Stone

- Webseek: Searching for Images and Videos on the World Wide Web, John R. Smith
- Get Started: Education, Tracy Hammerman
- Managing Access To Digital Information: Some Basic Terminology Issues, Patrice Lyons
- Workshop Summary: Digital Imaging for Libraries and Archives , Stephen Chapman and Anne R. Kenney, Cornell University
- Workshop Summary: Second DELOS Workshop --Interoperability and Metadata, Thomas H. Baker
- Workshop Summary: Technology Issues for Terms and Conditions, James R. Davis, Judith Klavans
- Workshop Summary: UK Serials Group -- Learning to Live with E- Journals: Some Practical Solutions, John Kirriemuir

Coming Soon to Your Favorite Library: Decision Support on Demand, Hemant Bhargava, Bob Norris
Content Ratings and Other Third-Party Value- Added Information: Defining an Enabling Platform, Martin Röscheisen, Terry Winograd, Andreas Paepcke
Creating a Networked Computer Science Technical Report Library, James R. Davis

D

Data Modeling for News Clip Archive: A Prototype Solution, Robert C. Plotkin, Michael S. Schwartz
Digital Conversion of Research Library Materials: A Case for Full-Informational Capture, Stephen Chapman, Anne R. Kenney
Digital Initiatives of the Research Libraries Group, Ricky L. Erway
Digital Labels for Digital Libraries, Robert Thibadeau
Digital Language Access: Scripts, Transliteration, and Computer Access, John Clews
Digital Libraries and Corporate Technology Reuse, Jonathan T. Hujsak
Digital Libraries in the Science Classroom: An Opportunity for Inquiry, Raven Wallace, Joseph Krajcik, and Elliot Soloway
Digital Libraries: Searching Is Not Enough; What We Learned On-Site, Andreas Paepcke
The Digital Library Research Agenda: What's Missing -- and How Humanities Textbase Projects Can Help, Allen Renear
DogitaLS1: A Digital Library System Based on Hyper-G, Klaus Tochtermann, Thomas Alders

> DogitaLS1: Ein digitales Bibliothekssystem auf der Basis von Hyper-G

E

Economic Framework for Pricing and Charging in Digital Libraries, J. Sairamesh, C. Nikolaou, D. Ferguson, Y. Yemini

*Editorials:*

- All Things in Good Time
- Back to School
- Collaboration Dreams, L. M. Simmons, Jr.
- Design by Experience
- Double-Edged Sword of Access
- Future is a Complex Place
- Getting Used to Technology and Revolutions in the Making
- In Memory of Paul Evan Peters, William Y. Arms
- In This Issue [July/August 1996]
- Levels of Abstraction
- New Year's Resolutions
- Of the Parts and the Whole
- On the fly
- Pandora's Box
- Straws in the Wind
- Taking the Measure of the Net
- Telling Time
- Text Is More Than Just Words on a Page, Susan Hockey
- Too Many Meetings?
- Tragedy of the Commons, Revisited (Again)
- Visiting Medical Informatics
- When is Honesty the Best Policy?
- Where Are We in Space?
- Why computer modeling should become a popular hobby, Kenneth D. Forbus
- Word (or two) of welcome

F

Fear of Offending: A Note on Educators, the Internet, and the Bible Browser, Richard L. Goerwitz
Federating Repositories of Scientific Literature: An Update on the Digital Library Initiative at the University of Illinois at Urbana-Champaign, Susan L. Harum, William H. Mischo, and Bruce R. Schatz
Finding Images/Video in Large Archives: Columbia's Content-Based Visual Query Project, Shih-Fu Chang, John R. Smith, Horace J. Meng, Hualu Wang, and Di Zhong
French Minitel: Is There Digital Life Outside of the "US ASCII" Internet? A Challenge or Convergence? Jack Kessler
From Static to Dynamic Surrogates: Resource Discovery in the Digital Age, Carl Lagoze

G

GEOREP: A WWW Customizable Georeferenced Digital Library for Spatial Data Marie- Josée Proulx, Yvan Bédard, Francois Létourneau, Christian Martel
GILS : What is it? Where's it going? Eliot Christian
Global Change Data and Information System-Assisted Search for Knowledge (GC-ASK) Project, Roberta Y. Rand

H

Hand- Made in Iowa: Organizing the Web Along the Lincoln Highway, Gerry McKiernan
Historical Collections for the National Digital Library: Lessons and Challenges at the Library of Congress, *(Part I)*, Caroline R. Arms
Historical Collections for the National Digital Library: Lessons and Challenges at the Library of Congress *(Part II)*, Caroline R. Arms

History Goes Digital: Teaching When the Web is in the Classroom, Bill Tally
How to Build a Digital Librarian, Kirk Hastings and Roy Tennant

I

Image Browsing in the Alexandria Digital Library (ADL) Project, B.S. Manjunath
Image Description on the Internet: A Summary of the CNI/OCLC Image Metadata Workshop, Stuart Weibel, Eric Miller
Informedia Digital Video Library: Technology Outreach, Howard D. Wactlar
Informedia Goes to School: Early Findings from the Digital Video Library Project, Michael G. Christel, Krishna Pendyala
An Integrated System for Distributed Information Services, George H. Brett II

J

Java and Libraries: Digital and Otherwise, Paul Jones
JSTOR: An IP Practitioner's Perspective, Sarah E. Sully
JSTOR: From Project to Independent Organization, Kevin M. Guthrie
Just- in-Time Conversion, Just-in-Case Collections: Effectively Leveraging Rich Document Formats for the WWW, John Price-Wilkin

K

Key concepts in the architecture of the digital library , William Y. Arms

M

Making of America: Online Searching and Page Presentation at the University of Michigan, Elizabeth J. Shaw, Sarr Blumson
Markup and Conversion of Japanese Classical Texts Using SGML in the National Institute of Japanese Literature, Shoichiro Hara, Hisashi Yasunaga
Meta- Information Environment of Digital Libraries, Terence R. Smith
Metadata: The Foundations of Resource Description, Stuart Weibel
Model Editions Partnership: Historical Editions in the Digital Age, David Chesnutt
The Model Editions Partnership: "Smart Text" and Beyond, David R. Chesnutt
Multi-Media, Multi-Cultural, and Multi-Lingual Digital Libraries, Or How Do We Exchange Data In 400 Languages?, Christine L. Borgman

N

National Digital Library of Theses and Dissertations: A Scalable and Sustainable Approach to Unlock University Resources, Edward A. Fox, John L. Eaton, Gail McMillan, Neill A. Kipp, Laura Weiss, Emilio Arce, and Scott Guyer
Need for a Common Infrastructure: Digital Libraries and Electronic Commerce, Daniel Schutzer
Netlib Mathematical Software Repository, Shirley Browne, Eric Grosse, Tom Rowan
New Center at Columbia University for Digital Library Research: Fostering Interdisciplinary Research and Bridging Cultural Clashes, Judith Klavans
The New Zealand Digital Library MELody inDEX, Rodger J. McNab, Lloyd A. Smith, David Bainbridge, and Ian H. Witten
The New Zealand Digital Library Project, Ian H. Witten, Sally Jo Cunningham, and Mark D. Apperley
News- on-Demand: An Application of Informedia® Technology, Alexander G. Hauptmann, Michael J. Witbrock and Michael G. Christel

O

OhioLINK: A Consortial Approach to Digital Library Management, David Barber
Options for the Future, Joshua Lederberg

P

Paying Their Way: Commercial Digital Libraries for the 21st Century, Innes A. Ferguson, Michael J. Wooldridge
Pricing Electronic Journals, Hal R. Varian
The Professional Magazine and Parallel Publishing, John MacColl
The Professional Web-zine and Parallel Publishing: The Web Version, John Kirriemuir

R

Recent Developments in GALEN II: Evolution of a Digital Library for the Health Sciences, John A. Kunze, Brian N. Warling
Electronic Records Research Working Meeting, May 28-30, 1997: A Report from the Archives Community, David Bearman, Jennifer Trant
The Red Sage Project: An Experimental Digital Journal Library for the Health Sciences, A Descriptive Overview, Richard E. Lucier and Peter Brantley
Relaxing Assumptions about the Future of Digital Libraries: The Hare and the Tortoise, William Y. Arms
Relaxing Assumptions . . . Stretching the Vision: A Modest View of Some Technical Issues, Ronald L. Larsen
Repository Interoperability Workshop: Towards a Repository Reference Model, William L. Scherlis
Research in Support of Digital Libraries at Xerox PARC; Part I: The Changing Social Roles of Documents, Marti A. Hearst
Research in Support of Digital Libraries at Xerox PARC: Part II: Paper and Digital Documents, Marti Hearst, Gary Kopec, Dan Brotsky
ROADS to Desire: Some UK and Other European Metadata and Resource Discovery Projects, Lorcan Dempsey
Roles for Digital Libraries in K-12 Education, Miriam Masullo, Robert Mack

S

SGML Creation and Delivery: The Humanities Text Initiative, Christina Kelleher Powell, Nigel Kerr
Safeguarding Digital Library Contents and Users: Assuring Convenient Security and Data Quality, Henry M. Gladney, J. B. Lotspiech
Safeguarding Digital Library Contents and Users: Digital Images of Treasured Antiquities, Henry M. Gladney, F.C. Mintzer, and F. Schiattarella
Safeguarding Digital Library Contents and Users: Document Access Control, Henry M. Gladney
SCAM Approach to Copy Detection in Digital Libraries, Narayanan Shivakumar, Hector Garcia- Molina
Secure Repository Design for Digital Libraries, Carl Lagoze
SOS- DS: Système optimal de sélection des données spatiales couplé à un data warehouse, François Létourneau, Yvan Bédard, Marie-Josée Proulx
SOS- SD: A Data Warehouse-Based System for the Optimized Selection of Spatial Data, François Létourneau, Yvan Bédard, Marie- Josée Proulx
Sink or Swim? The U.S. Navy Virtual Library (NVL), Bob Norris, Denise Duncan
*Sui Generis* Database Protection: Has Its Time Come?, Jonathan Band, Jonathan S. Gowdy
Summary of Stanford's Digital Library Testbed Design and Status, Andreas Paepcke

SunSITE: Serving Your Internet Needs Since 1992, Judson Knott, Paul Jones

T

Task Force on Archiving of Digital Information, John R. Garrett
Testbed Development for the Berkeley Digital Library Project, Virginia Ogle and Robert Wilensky
Text Is More Than Just Words on a Page, Susan Hockey *(guest editorial)*

U

Uniform Resource Names: A Progress Report, The URN Implementors
University of Michigan Digital Library Project: The Testbed, Daniel E. Atkins
User- Centered Iterative Design for Digital Libraries: The Cypress Experience, Nancy A. Van House, Mark H. Butler, Virginia Ogle, Lisa Schiff
The User Community as Responsibility and Resource: Building a Sustainable Digital Library, David Seaman

V

The VARIATIONS Project at Indiana University's Music Library, David E. Fenske, Jon W. Dunn

W

W3C and Digital Libraries , James S. Miller
Warwick Framework: A Container Architecture for Diverse Sets of Metadata, Carl Lagoze
Warwick Metadata Workshop: A Framework for the Deployment of Resource Description, Lorcan Dempsey, Stuart L. Weibel
Web for Your Head: The Design of Digital Resources to Enhance Lifelong Learning, Christopher M. Hoadley, Philip Bell
What Do People Want from Information Retrieval? (The Top 10 Research Issues for Companies that Use and Sell IR Systems), W. Bruce Croft
What's Wrong with Internet Searching, Annabel Pollock, Andrew Hockley
Why computer modeling should become a popular hobby, Kenneth D. Forbus, *guest editorial*
Working Towards an Understanding of Digital Library Use: A Report on the User Research Efforts of the NSF/ARPA/NASA DLI Projects, Ann Peterson Bishop
Workshop Report: The Technology of Terms and Conditions, James R. Davis, Judith L. Klavans
Workshop Summary: UCLA-NSF Workshop on Social Aspects of Digital Libraries, Christine L. Borgman

Y

Yale University Library's Project Open Book: Preliminary Research Findings, Paul Conway

Z

The Z39.50 Information Retrieval Standard: Part I: A Strategic View of Its Past, Present and Future -- *The first of a two-part story appearing this month and next*, Clifford A. Lynch
Z39.50: The User's Perspective, Sandra D. Payette, Oya Y. Rieger

**Copyright © 1997 Corporation for National Research Initiatives**



*hdl:cnri.dlib/september97-title.index*

# APPENDIX D

https://web.archive.org/web/19971010043650/http:/www.dlib.org/dlib/january97/retrieval/01shneiderman.html November 6, 2024 at 1:50 PM PST
http://www.dlib.org/dlib/january97/retrieval/01shneiderman.html Go SEP OCT FEB
269 captures 10 1996 1997 2000
10 Oct 1997 - 10 Jun 2024 About this capture

MAGAZINE

# Clarifying Search

## A User-Interface Framework for Text Searches

Ben Shneiderman
Department of Computer Science and Human-Computer Interaction Laboratory
University of Maryland
College Park, MD 20742
*ben@cs.umd.edu*

Don Byrd and W. Bruce Croft
Center for Intelligent Information Retrieval
University of Massachusetts
Amherst, MA 01003
*dbyrd,croft@cs.umass.edu*

**D-Lib Magazine**, January 1997

**ISSN 1082-9873**

**Abstract:** Current user interfaces for textual database searching leave much to be desired: individually, they are often confusing, and as a group, they are seriously inconsistent. We propose a four- phase framework for user-interface design: the framework provides common structure and terminology for searching while preserving the distinct features of individual collections and search mechanisms. Users will benefit from faster learning, increased comprehension, and better control, leading to more effective searches and higher satisfaction.

---

*[Note added at the request of the Authors, January 15, 1997. The Library of Congress has recently improved the interface that we were using in Case Study 1 as the "before" example. We are pleased with their improvements, but the advantages of our "after" example are now less striking. We hope others will also improve their interfaces as a result of this article.]*

- Introduction
  - The problem
  - Towards a Solution
- The four-phase framework for search
  - Formulation
  - Action
  - Review of results
  - Refinement
- Building Effective User Interfaces
- Case Studies
- Conclusions
- Appendix 1: Definitions
- Appendix 2: Feedback from Web Search Tools
- Appendix 3: Terminology Survey
- Acknowledgements
- References

---

### Introduction

**The problem.** We believe that an opportunity exists to improve user interfaces for textual database searching dramatically. The ideal user interface is as simple as possible, and it makes key features as clear as possible. But many of the current text-search interfaces -- especially on the World Wide Web -- are neither simple nor clear: they are often needlessly complex, and they very often obscure key features. The result is confusion, frustration, and failure for intermediate and advanced users as well as novices. Zero-hit outcomes occur on 30% of searches at some services, while huge numbers of hits distract users in many other cases. While online search services such as Infoseek, AltaVista, Lycos, WebCrawler, and Open Text are widely used, public and professional concern about the difficulty of finding information is great (Flynn, 1995; Hatlestad, 1996; Somerson, 1996). But we believe that improved designs can lead to a far greater number of positive outcomes. Recent studies appear to confirm that when users are given more information on and control over their searches, their performance and satisfaction increases (Koenemann and Belkin, 1996).

Improved user-interface design is clearly part of the solution; yet design has its own set of challenges. Consider the diversity of the user community a broadly accessible resource like the Web proposes to serve. One of the main sources of diversity, though by no means the only one, is differences in experience among users (Shneiderman, 1992):

> First-time users need an overview to understand the range of services. . .plus buttons to select actions. Intermittent users need an orderly structure, familiar landmarks, reversibility, and safety during exploration. Frequent users demand shortcuts or macros to speed repeated tasks and extensive services to satisfy their varied needs.

Poor user-interface designs can be improved. But even then, as users move from one search service to another, inconsistencies can cause slower performance, uncertainty, mistaken assumptions, and failures to find relevant documents. For example, the search string "Hall effect" could produce (among many other possibilities) a:

- search on the exact string "Hall effect"
- case-insensitive search on the string "hall effect"
- probabilistic search for "Hall" and "effect"
- probabilistic search for "Hall" and "effect", with higher weights if "Hall" and "effect" are in close proximity
- error message indicating missing AND/OR or other operators/delimiters
- Boolean search on "Hall" AND "effect"
- Boolean search on "Hall" OR "effect"

Many existing systems give little or no indication of which interpretation they are using. Nor does the above list hint at all the query-processing transformations in common use: there are also questions of stemming, stop words, relative weights of fields, etc. Finally, in many systems the results are displayed in a relevance ranking whose meaning is a mystery to many users (and sometimes a proprietary secret).

The suggestions given here are designed to be complementary to ongoing research in information retrieval interfaces and visualization. See, for example, Rao et al (1995), Shneiderman (1994), and Van House et al (1996), as well as various papers in the annual ACM SIGIR proceedings (ACM).

**Towards a Solution.** Based on experience with many systems (Shneiderman, 1992) as well as recent efforts with the Library of Congress's THOMAS (Croft, Cook and Wilder, 1995) and American Memory projects, we propose a four-phase framework for thinking about text-search user interfaces. We expect the framework to be of interest to information retrieval specialists who are concerned about user interfaces. Note, however, that this paper addresses only interfaces for finding information by searching. Browsing -- of indices, alphabetical lists of terms, news articles, etc. -- may be equally important in some applications, but it has its own set of challenges.

Admittedly, user-interface differences sometimes result from functionality differences: no Boolean system (in the usual sense of the term) can do probabilistic searches. Nonetheless, inconsistencies can and should be reduced greatly, and those that remain can and should be made clear. The basic automobile user interface is something we now take for granted, but it took many strange meanders over several decades to reach this level of standardization (Oliver and Berkebile, 1968; Buxton, 1989); and remaining inconsistencies like left/right variations from country to country still cause serious problems for travelers. As software designers, we should be able to do much better than we are doing now, and thereby to spare our users millions of conceptual fatalities.

In the remainder of this paper, we outline the four phases; make recommendations as to how to implement the phases, based on the user's perspective; and show how two existing systems (one Web-based, one standalone) could be redesigned in accordance with our recommendations. We believe that designers armed with this information will be in a good position to satisfy the needs of all users, both in standalone situations and over networks including the World Wide Web.

**The four-phase framework for search**

The four-phase search process gives great freedom to designers of specific systems to offer a variety of features in an orderly and consistent framework. The phases are: formulation (what happens before the user starts a search); action (starting the search); review of results (what the user sees resulting from the search); and refinement (what happens after review of results and before the user goes back to formulation with the same information need).

Actually, before performing a search, users must consider their information need and clarify their search goals. But this is just what a computer system cannot help with.

**1. Formulation:** This is the most complex phase in that it involves decisions of several types, each of which may itself be complex. These decisions include the *sources* of the search, i.e., where to search; which *fields* of documents to search; what actual *text to search for;* and what *variants* of that text to accept. Some systems actually walk the user through each of these decisions in succession, but they cannot always be made in a predetermined order; nor are we convinced they exhaustively cover the query-formulation possibilities.

> **a. Sources:** The first step in performing a search is normally to decide where to search (Marchionini, 1995). This is often a single physical database, but increasingly it is multiple and distributed databases, accessed across a network.
>
> Even if technically and economically feasible, searching all libraries or all collections in a library is frequently not the preferred decision. When users are confident they know where the truly relevant material is, they often prefer to limit the scope of their searches to a specific library (say, NASA, Princeton, or the DIALOG system), a specific collection in a library, or a specific range of documents in a collection.
>
> In most cases, users decide "by inspection" where to search. However, this decision can also be made by an instance of exactly the same procedure as the final search. Specifically, some systems support a process called "collection selection", in which the user's query is run against a special database that describes the contents of all known databases (Callan et al, 1995). The result, instead of a list of best-matching documents, is a list of best-matching databases. The user can then run their query against one or more databases in the list.
>
> **b. Fields:** Each document in a collection may have multiple fields (sometimes called attributes, components, or tags). Users may wish to limit their search to specific fields of documents within a collection. For example, users searching on common terms might prefer to retrieve only documents whose title contains that term, or at least to give a higher rank to documents whose title contains that term: see for example the elaborate weighting algorithm used by THOMAS (Croft, Cook, and Wilder, 1995).
>
> Searches may also be restricted by structured fields (year of publication, volume number, language, media type, publisher, etc.). For example, searchers in the Congressional Record may wish to restrict searches to items involving a specific member of Congress.
>
> **c. What to search for:** There are various ways to express what to search for in full text; the most important are probably (1) unstructured text, (2) text with embedded operators, and (3) text with operators specified separately. Pure unstructured-text interfaces are unusual: most of the popular Web search services (AltaVista, Infoseek, Lycos etc.) and other systems such as INQUERY accept either unstructured text or text with embedded operators. An example of the latter is Infoseek's "city-guide +Boston" (the words "city" and "guide" must appear in close proximity, and the word "Boston" is required). Finally, Open Text's Power Search uses text with separate operators. All three ways can be effective, but only if they are used properly.
>
> A key issue here is that of phrases. In many situations, especially with short queries, searches on meaningful phrases are much more effective than searches on the words of the phrase. Using phrases will generally increase precision at the expense of recall. For example, for someone searching for information on air pollution, the phrase "air pollution" is likely to find fewer irrelevant documents (higher precision) than the pair of words "air" and "pollution" -- though it will tend to overlook relevant documents that refer to "air quality" or "atmospheric pollution" (lower recall). In particular, phrases facilitate searching on names: for example, a search on "James Billington" should not turn up "Jesse James". It should be easy for users to specify, and easy for them to know if they have specified, that a series of words should be considered a phrase.
>
> Unstructured text, approach (1), is often called "natural language", and indeed, it *looks* like natural language, but this can be extremely misleading. For example, many systems treat "and" and "not" as stop words. In such a system, the query "bees and not honey" means the same thing as just "bees honey": compared to the query "bees", it is *more* likely to retrieve information about honey, not less. Even if the system pays attention to words like "and" and "not", it may parse a complex query differently from the user's intention. Similarly, a user might well assume that semicolons between words will be taken as an indication that the words are not part of a phrase, but AltaVista has exactly the opposite interpretation. The only real solution to this kind of ambiguity is with feedback informing users of how the

system interpreted their queries, but it is very difficult to give this kind of feedback in a way that will be clear to nontechnical persons.

In theory, text with embedded operators, approach (2), can be completely unambiguous, including specifying phrases; it can also specify fields. However, our experience has shown consistently that a great many users will have trouble with this approach. One reason is lack of standardization: the syntax and meaning of embedded operators vary considerably from one system to another, so it is easy to get confused. For example, in Alta Vista advanced search and Yahoo!, a "wildcard" (matching anything) is indicated with "*"; in Lycos, it's "$". In Infoseek, Magellan, and Yahoo!, "- " preceding a word means it's forbidden, and "+" preceding a word means it's required; in Lycos, "-" means forbidden, but there seems to be no way to mark a word as required. Another problem is the danger of inadvertent activation: innocently using text that the user thinks of as unstructured, but which contains characters or strings that will be interpreted as embedded operators. For example, some systems (WebCrawler and INQUERY among them) use parentheses to delimit phrases or other groupings; in such a system, pasting in text containing parentheses might result in prematurely ending a grouping.

Other than embedded operators, the only way we know of to specify phrases unambiguously is with an implementation of approach (3), text with operators specified separately: the program considers the contents of every text-entry box as a phrase, and clearly says so on the screen. Then multiple entry boxes must be provided to allow for multiple phrases. (Of course, a text-entry box must also accept a single word.) If choices of Boolean operations, proximity restrictions or other strategies for combining the boxes are available, then users should be able to express them; regardless of whether any choices are available, users must be told what combining technique is being used. Ideally, users and/or service providers should have control over stop lists (common words, single letters, etc.); at a minimum, users should be warned when they try to search for a stop word.

The basic issue is always this: Does the program interpret the query the way the user intended it, and -- even if it does -- does the user *know* that the program interprets it that way? A significant advantage of approach (3) is that, correctly implemented, it is probably the easiest for users to understand.

It is important to allow searching structured fields (controlled-vocabulary text, dates, etc.) in databases that also contain unstructured text at the same time as text fields are searched. The user-interface issues for specifying "what to search for" in structured fields are the same as for standard database systems.

**d. Variants:** Users are very often unsure of the exact value of the field they want; indeed, there may not be any single value that is appropriate. As a result, users may want variants to be accepted. In structured fields of text databases, as in traditional databases, this may include a range on a numeric or date field. In unstructured text fields, interfaces may allow user control over:

- capitalization (case sensitivity)
- stemmed versions: searching for "teach" finds words like "teacher", "teaching", "teaches"
- partial matches: searching for "biology" retrieves "sociobiology" and "astrobiology"
- phonetic variants, e.g., from N-grams or soundex-like methods: searching for "Johnson" finds "Jonson", "Jansen", and "Johnston"
- synonyms: searching for "cancer" finds "malignant tumor"
- abbreviations/acronyms: searching for "Digital Equipment Corporation" finds "DEC"
- broader or narrower terms from a thesaurus (searching for "New England" finds "Vermont", "Maine", "Rhode Island", "New Hampshire", "Massachusetts", and "Connecticut", and vice versa).

In addition, this item should include stop words. (Of course, the fact that some words are stopped has nothing to do with variants in the normal sense.)

In all cases, the user interface should make it clear how variants are handled.

**2. Action:** Searches may be started explicitly or implicitly. The typical usage process in current systems is to have users click on a Search button to initiate the search and then wait for the results. But 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. See, for example, the commercial system Folio Viewer, or research prototypes like Ahlberg and Shneiderman (1994) or Shneiderman (1994). This approach requires adequate screen space and high bandwidth, plus, for a large database, very rapid processing: whether it is feasible depends very much on the situation. However, when it is practical, the advantages are great: users can broaden, narrow, or refocus their search several times in as many seconds. Designers may also allow users to choose between approaches.

In situations where it is not practical to re-run the query and update results continuously -- for example, when the database and the user are connected by a network with limited bandwidth -- the "query preview" approach is worth considering (Doan et al, 1996). In this approach, changes to the query simply update a display (perhaps just an estimate) of the number of hits. The query is not actually re-run until the user requests the full results, presumably when they are satisfied that the number of hits is neither zero nor so high as to be cumbersome. However, it is not yet clear how such an approach can be applied to full-text information retrieval.

A final comment: users should have an obvious way to stop the search in case they feel it is taking too long. Most if not all popular Web browsers have a Stop button, so this should not be an issue for Web interfaces. In addition, many window systems have a standard way of doing this that does not rely on any visible part of the user interface (henceforth "UI"), but less sophisticated users may not know that or may not remember how to activate it. (One would hope that this function would rarely be needed in dynamic-query systems, whose design response times must be very brief.) If the search interface includes both Search and Stop buttons, they should be close together.

**3. Review of results:** For some time, information retrieval interfaces have let users specify result set size (for example, a maximum of 100 documents), contents (which fields are displayed), sequencing of documents (alphabetically, chronologically, relevance ranked, . . .), and, occasionally, clustering (by field value, topics, . . .). All of these capabilities can be valuable, but they all simply try to make a list of documents easier to handle. A query against a large database, even a query that is well focused, can produce so many potentially-useful hits as to be overwhelming -- say, several hundred or more. Fortunately, much more can be done to display results in a useful form.

Recent work in information retrieval interfaces, capitalizing on general information-visualization research, has dramatically expanded the limited traditional palette of display techniques. For example, LyberWorld (Hemmje et al, 1994) displays document icons in a circle, with terms around the circumference "pulling" the documents towards themselves; the terms can be moved and the strengths of their pulls varied. Rao et al (1995) describes such techniques as tilebars, perspective walls, cone trees, and document lenses. Swan and Allan (1996) discuss three-dimensional network displays, with the viewpoint adjustable in real time, to support clustering. Finally, "virtual reality" flythroughs of simulated document spaces are being explored intensively. For example, the Web page for Apple Computer's HotSauce says "Download the HotSauce fly-through plug-in to fly through 3D representations of Web space right away."

Search interfaces should also provide helpful messages to explain search results and to support progressive refinement. For example, if a stop word or misspelling is eliminated from a search input window, or stemmed terms, partial matches, or variant capitalizations are included, users should be made aware of these changes to their query. If the two words in a phrase are not found proximally, then feedback might be given about the occurrence of the words individually. If multiple phrases are being sought, then perhaps documents containing all phrases should be shown first and identified, followed by documents containing subsets, but if no documents are found with all phrases, this would be indicated. A fairly elaborate decision tree (perhaps 50 to 100 branches) of search outcomes and messages might be specified.

**4. Refinement:** One of the most important ways in which current information retrieval technology supports refining searches is relevance feedback. A search interface can support relevance feedback in a variety of ways. Koenemann and Belkin (1996) describe a user test of several ways, and suggests that users should be able to see and manipulate the words relevance feedback adds to their query. (Using the term "relevance feedback" in a user interface is not very satisfying: it is rather unintuitive, and it refers to the system's perspective, not the user's. But, at this point, it is almost completely standard. The only alternatives we have seen are "more like this" and Infoseek's and WebCrawler's "similar pages". Another possibility might be "high relevance".)

Another aspect of refinement is supporting successive queries. As searches are made, the system might keep track in a history buffer to allow review of earlier searches. In any case, progressive refinement should be convenient.

Finally, a system might make search results and the settings of each phase objects that can be saved, sent by e-mail, or used as input to other programs, for example visualization

or statistical tools. But these are mostly matters of convenience, not refining the search.

**Building Effective User Interfaces**

Our goal in this section is to provide designers with general guidelines for effective user interfaces for information retrieval.

User-interface design is a large topic and a growing discipline (Shneiderman, 1992; Preece, 1994; Baecker et al., 1995). Guidelines documents from commercial providers such as Apple, Microsoft, and IBM are widely available and contain hundreds of good suggestions. Short lists of "golden rules" have been provided by several authors; Shneiderman (1992) offers eight rules that can be useful here. Rephrased for the context of information retrieval and the four-phase framework, they are:

1. Strive for consistency. Ensure that terminology, instructions, layout, color, and fonts are used consistently across search user interfaces. For example, changing the search-initiation button label from "search" to "query" or "browse" has been shown to slow user performance and lower satisfaction significantly (Mahajan and Shneiderman, 1996). Based on the user survey described in Appendix 3 and on our experience, we recommend using the terminology we have used above or the alternatives in Table 1.

***Table 1. Terminology***

| Recommended term | Alternative |
|---|---|
| Sources | "Databases". |
| What to search for | "Phrases" (if the contents of an input box will be treated as a phrase) or just "Search for". |
| Action | Nothing, i.e., omit the label entirely. |
| Search | "Start Search" may be clearer in some contexts, especially on the Web, where a button labeled "Search" might simply jump to a search page. |

2. Provide shortcuts for skilled users. An obvious example here is the keyboard equivalents for menu commands that systems like the Mac OS and Microsoft Windows provide: these are particularly helpful because they're self-documenting. As another example, users who already know a term or document identifier should not have to perform a time-consuming search or navigate through a lengthy series of menus and dialogs. (A separate issue is whether long series of menus and dialogs are desirable at all: see below.)

3. Offer informative feedback. This is the point we have emphasized in the discussion of the four-phase framework. The user should be informed about all aspects of the search they are preparing to do: the sources, fields, what is being searched for, and what variants are being allowed. When the search is complete, it should be obvious to the user what happened and why.

For example, why was a given document retrieved? In "conventional" text databases, presumably because words that appear in the query (or variants of them) also appear in the document, and simply highlighting them in a display of the document text is usually sufficient. But in hypertext (i.e., on the Web), a document may have been retrieved partly or entirely because of documents it links to, and things are not so easy. (Unfortunately, this difficulty is compounded by the fact that Web search tools never offer a "custom" display of pages they retrieve, so there is no way to see the retrieved document with query words highlighted.) Another example: when zero-hit or overwhelming-number-of-hit results are produced, users should be given some suggestions as to what to do next. A final example: it is critical to make clear what is being searched for, but many popular search tools do not. To see this, try the query "and or" -- an extreme case, but one a student of linguistics or logic might conceivably give -- in Infoseek, Lycos, or Yahoo. (See Appendix 2 for details of how this query behaves in a number of Web search tools.)

4. Design for closure. Users should know when they have searched a complete database or have viewed every item in a browse list. Traversing a deep menu tree is disorienting, especially when backtracking and exploration are expected. In most situations, a broader tree with fewer levels is much better, since it allows users to reach their destination in fewer steps. Broad, shallow trees also reduce short-term memory load.

5. Offer simple error handling. Syntax errors should be prevented where possible; all error messages should be specific, constructive, and no more technical than necessary; and changes to search parameters should be easy to apply. For example, one error message from XINQUERY says "eval_query "" 50 "" inq_eval_query called with zero length query.": nearly incomprehensible to most non-programmers. A far preferable statement would be something like "No search text was given. Enter text and try again."

6. Permit easy reversal of actions. Every action should be reversible so users can go back to a previous state in a session. In our context, the best example is probably keeping a history of queries given and letting users re-issue them. This is particularly valuable if the complete context of each query -- for example, relevance feedback -- is captured as well.

7. Support user control. In a well-designed interface, users initiate action, monitor progress of long searches, and always feel in control. Most users greatly prefer interfaces with no enforced sequence of actions; they should be able to set parameters for a search in whatever order they prefer. Another way to give users a sense of control is to provide a visual overview of an entire database (Ahlberg and Shneiderman, 1994): then visual feedback about search outcomes help users gain a better understanding of their progress.

8. Reduce short-term memory load. Keep a session history, so users can always go back and reuse previous effort. While spreading information over several screens may be graphically appealing, the burden of shifting from one screen to another is large. Studies show that more compact presentations on fewer screens are more effective. Compact presentations do take slightly longer to scan, but much less time than scanning several spread-out presentations. Similarly, in web page design, compact vertical presentations -- reducing the need to scroll -- are highly beneficial.

One additional rule specific to text-search interfaces is worth mentioning, especially because to some extent it contradicts rule 8: Allow plenty of space in text-entry boxes. This is particularly important because longer search text very often gives better recall and/or precision, and so users should be encouraged to use long search strings.

**Case Studies: Two User-Interface Redesigns**

To clarify both the framework and the user-interface guidelines, we will now give "before" and "after" examples of two text-search interfaces. One is Web-based; the other is a standalone application for a desktop computer.

**Case Study 1: Web Interface**

The current search page for the Library of Congress's THOMAS system enables users to find text in the Congressional Record by full-text search. It is typical in many ways of search pages currently on the Web. With a modest amount of effort, knowledgeable users should be able to find what they are looking for, but it does leave several features unexplained and could be troubling to first-time users. For example, the page has multiple sets of search and clear buttons that perform the same function and may be confusing. The controls that allow a user to search certain sections are located near the "Word/phrase" box, but are not near the other attribute items such as date range and/or Congressperson. The control for the maximum number of items to return is below the final search button, where it may be overlooked by users scanning from top to bottom. Handling of variants, such as case-sensitivity and stemming, is not mentioned. The valid date range for the date range selector is not given. Finally, a list of Congresspersons would help users when entering a name.

Guided by the four-phase framework, the revised version uses an HTML table to organize the components. It starts out by clearly stating the Sources of the search, including the valid dates of the 104th Congress. The Fields section, whose elements limit the search, contains radio buttons for selecting a section of the text, a drop-down box for choosing a member of Congress, and a date range selector. The inclusion of a drop-down box eliminates the burden of spelling a Congressperson's name correctly. The date range now specifies valid dates for this search. (It would be better to replace the two boxes with a double box slider to specify the date range, allowing for rapid adjustments to the search.) Variants allowed are described for the user. Three phrases can be entered, one to a box (note, however, that in this example only the first box is functional). The maximum number of results to return can be set in the Results section. This section also provides the user with information on the sort order of the results. Finally, only one set of search/clear buttons appears.

As of this writing, 800x600-pixel displays seem to be about average. On such a display, the current search interface takes up about two screens, so scrolling is necessary to view all elements of the search. The revised interface fits on one 800x600 screen.

Overall, the the functionality of both interfaces is the same. However, we believe that the changes will shorten learning times, improve user effectiveness, reduce errors, increase retention, and raise satisfaction.

**Case Study 2: Standalone Interface for a Desktop Computer**

XINQUERY (Fig. 1, below) is a front end for CIIR's INQUERY retrieval engine; it runs under X Windows. XINQUERY supports single-database-at-a-time searches.

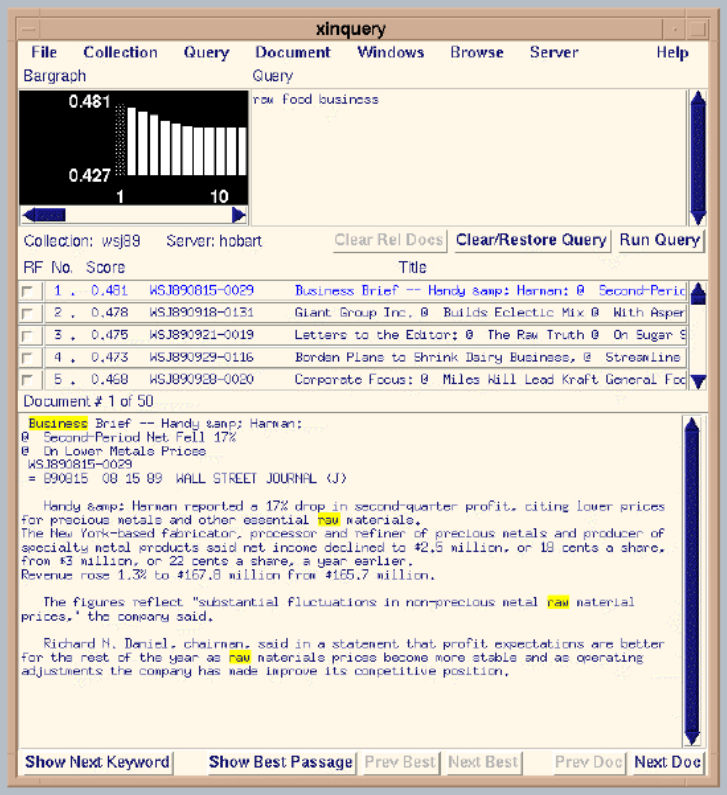



***Figure 1. XINQUERY***

Note that all major user-interface areas appear within a single window:

- In the upper left, a bar graph showing document-by-document scores for the retrieved documents (the area is blank until a query is run). Each bar in the bar graph corresponds to a document title, and the length of each bar corresponds to the relevance-score number to the left of the title.
- In the upper right, a text-entry area, into which an unstructured or structured query is entered.
- Just below those areas and across the window, an area in which a summary of the results (i.e., list of retrieved documents, with numeric scores estimating their relevance) is displayed. This area is also used to control relevance feedback: note the check boxes under the column heading "RF".
- Further down still and across the window, an area in which the text of one of the retrieved documents may be displayed, with text matching the query highlighted. To display a document, the user double-clicks either a document title or its bar in the bar graph.

In addition, the window displays a number of buttons, all of the instantaneous kind; the name of the database currently being used; and some information about the search results.

But XINQUERY's user interface exhibits many deficiencies (not all visible from these screen shots). To point out some of the more serious:

1. The actual numeric scores given for documents are virtually meaningless, even to experienced INQUERY users. Relative scores shown by the heights of the bars are much more useful, but it is not easy to see the close relationship between bars in the bar graph and documents in the results area.

2. The appearance of the entire UI in one window strongly suggests tight coupling between query display and results; but results are not updated until the user explicitly runs a query. As a consequence, the query and the results shown can easily be "out of sync".

3. The only indications of what variants are in use are incomplete, vaguely worded, and available only through menu commands with unintuitive names.

4. Only one document can be viewed at a time, and not much of that document at a time, since much of the window is occupied by other "widgets".

5. There is a "query history" mechanism to support refining queries, but only for the current session: a user can save only individual queries.

6. Many error messages take the form of a beep when they occur: to find out what the problem is, the user must open a special error-list window. Even then, many of the messages are nearly incomprehensible to a non-programmer. The example we gave before, "eval_query "" 50 "" inq_eval_query called with zero length query", is fairly typical.

7. It is not as obvious as it could be whether relevance feedback is in use.

We began work on a new front end for INQUERY early in 1996. The current version, WInquery, is shown in Fig. 2; it is written in Visual Basic, and runs under Windows 95 and Windows NT.

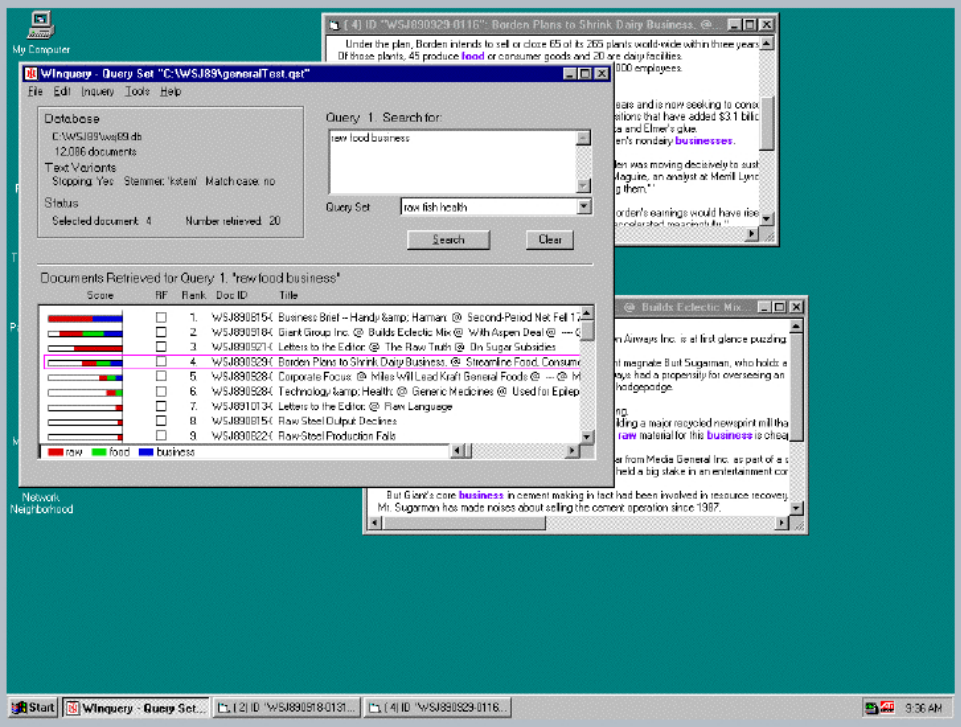



*Figure 2. WInquery*

The version of WInquery shown is similar to XINQUERY in many ways: it also supports single- database-at-a-time searches, and much of the user interface is similar. However, WInquery addresses many of XINQUERY's problems along the lines of the four-phase framework, though it follows the framework less strictly than the THOMAS redesign. Changes to the XINQUERY trouble spots we listed are as follows:

1. There is no separate bar graph area. Instead, the bar representing each document's score appears horizontally, to the left of the document's title in the summary. The numeric scores are normally hidden, but can be displayed if the user wants them.

2. To reduce the chances of a user incorrectly assuming results displayed correspond to the query displayed in the query-text window, the summary is headed by the query whose results it contains. In addition, query sequence numbers above the query-text window and with the summary-area heading indicate whether the two correspond or not.

3. The status of stemming, stopping, and case sensitivity (which are features of the database and cannot be changed) are shown underneath the database name at all times.

4. Several documents (currently a maximum of four) can be shown at once. Each appears in a separate window, which can be as large or small as desired.

5. Files of queries can be opened, modified, and saved. The history of queries issued in a session can be saved.

6. Error messages are straightforward, and they appear in modal windows when the error occurs. For example, trying to run a query with an empty query-text field simply says "No query given. Type or paste something into the query area and try again." (While this is certainly an improvement for most first-time or intermittent users, note that an expert might find XINQUERY's default beep adequate and might prefer not to be bothered with a modal window. This issue could be addressed in several ways, e.g., with a preference to choose between the two methods.)

7. If relevance feedback was used in evaluating a query, the caption above the summary area says so. If relevance feedback is set for the query that will be evaluated if the "Search" button is pressed, the button's label says "Search with RF".

Aside from addressing XINQUERY's deficiencies, WInquery has the unusual feature of breaking the score bars into colored segments to form a "stacked histogram", where each segment shows the contribution of a word in the query to the document's score, according to the legend shown at the bottom of the window. This is reminiscent of Hearst's tilebars (Rao et al, 1995), but the resemblance is mostly superficial. Tilebars provide information that stacked histograms omit, namely where words are used in each document; stacked histograms give information tilebars omit, namely the documents' relative scores and how much each word contributed to each document's score. (It might appear easy to infer a word's contribution from a tilebar, but this is not so, since most systems assign different words very different weights.) For example, the first document's high ranking came mostly from the word "raw", with a smaller contribution from "business". A user can also change the mapping of words to colors, assigning as many words as they want to a single color. This facility, like tilebars, comes under the rubric of offering informative feedback.

A chart showing each user step and the corresponding system support can clarify a given system's strengths and weaknesses, and so may be useful both in designing and evaluating systems. Table 2 is such a chart for WInquery.

*Table 2. Terminology*

| Phase | System support |
|---|---|
| 1. Formulation: | |
| a. Sources | Open menu command; current source is displayed at all times |
| b. Fields | Embedded commands |
| c. Search for | Edit text box; displayed at all times |
| d. Variants | No options; treatment is displayed at all times; Show Parsed Query command |
| 2. Action | Start Search button and menu command |

| | |
|---|---|
| 3. Review of Results | In Preferences command, set maximum items to retrieve, whether to display numeric scores, whether to show colors for query-term contributions; can change assignment of terms to colors; document display highlights matched words; can display several documents at once |
| 4. Refinement | Relevance Feedback; support for query sets (user can choose and re-issue or modify any query from a query set; the name of the current query set is displayed at all times; query sets can be saved) |

Finally, in a complex interface with many options, it might also be useful to break the system support down according to what it does for each category of user -- for example, first-time, intermittent, or frequent.

**Conclusions**

In summary, the four-phase framework focuses on:

**1. Formulation:**

**a. Sources:** specify which libraries and/or collections to search and the search range within them.

**b. Fields:** each document in a collection may have multiple fields. Users specify which text fields are to be searched. Searches may also be restricted by structured fields.

**c. What to search for:** users select or type in text, perhaps as one or more phrases. Users may have control over stop lists (common words, single letters, etc.).

**d. Variants:** searches might allow user control over variant capitalization, stemmed versions, partial matches, phonetic variants from soundex methods, and synonyms, abbreviations, broader, or narrower terms from a thesaurus. In all cases, the user interface should make it clear which variants, if any, are allowed.

**2. Action:** how does a search get initiated -- explicitly (e.g., with a button), or implicitly (e.g., when some aspect of the query is changed).

**3. Review of results:** conventional options are, for example, to specify result set size, layout, sequencing (alphabetically, chronologically, relevance ranked, etc.), and contents (which parts and fields are displayed). Less conventional interfaces might employ a wide variety of techniques, including many based on information-visualization research.

**4. Refinement:** provide feedback on search results with informative messages and clustering of results. For example, enable progressive querying, especially with relevance feedback; history keeping; and extraction of results to files, perhaps for use in e-mail.

The sample Web interface we discussed above requires nothing more advanced than HTML tables and forms, but in some cases, applying the framework successfully over the Web may require more powerful tools such as Java. On the other hand, while we have thought mostly about text situations and both of our sample interfaces are for full-text searching, we suspect our framework will prove appropriate for multimedia as well as text.

Finding common ground will be difficult; not finding it would be tragic. While early adopters of technology are willing to push ahead to overcome difficulties, the middle and late adopters will not be so tolerant. In particular, the future of the World Wide Web as a universally acceptable tool may depend on our ability to reduce the frustration and confusion of the masses of users, while enabling them to reliably find what they need.

**Appendix 1: Definitions**

The following definitions may facilitate discussion.

*Inadvertent activation,* in human factors, means activating a user-interface feature without intending to. In systems that use both single and double clicking extensively, a common mistake of new users is to double click when they intend to single click: the result is inadvertent activation of the double- click behavior.

An *information need* is the perceived need for information that leads to someone using an information retrieval system in the first place.

Two terms are commonly used in the evaluation of information retrieval systems: precision and recall. *Precision* is the ratio of the number of documents retrieved that "should" have been retrieved -- i.e., the number of retrieved documents that were really relevant to the query -- to the total number of documents retrieved. *Recall* is the ratio of the number of relevant documents retrieved to the number of relevant documents in the database. Each can vary from 0 to 1, and the higher the value, the better.

Morris Hirsch (1996) has given an elegant statement of why these concepts are important:

> If you use *any* text search system, you will soon encounter two language-related problems: (1) low recall: multiple words are used for the same meaning, causing you to miss documents that are of interest; (2) low precision: the same word is used for multiple meanings, causing you to find documents that are not of interest.

*Relevance feedback* is the process of taking retrieved documents that have been determined to be good examples of what the user wants, and using them to produce an improved query. Determining which retrieved documents are good ones is normally done by the user (for example, clicking a "Similar Pages" button in Infoseek).

*Stemming* means converting words to their presumed roots. For example, "blacker", "blackest", and "blacks" may all be converted to "black".

*Stop words* are words that the system ignores, normally because they are assumed to be so common as to carry little information useful for distinguishing relevant documents from non-relevant ones, while burdening the system with much larger index files, data structures, and so on.

A *controlled-vocabulary* field in a database is one that accepts only words or other values from a pre-defined list. The term "controlled vocabulary" can also be applied to entire systems.

*Unstructured text* or *free text* (as distinguished from structured text or text involving controlled vocabulary) is text in which any word or sequence of words might appear, and no word is "reserved" to carry some special meaning. For example, in a system in which "and" is a Boolean operator, "cats and dogs" would be structured text; in a system in which it is not a Boolean operator, the same series of words might be unstructured text. Unstructured text is often called "natural language", but this can be extremely misleading: see What to Search For for more details.