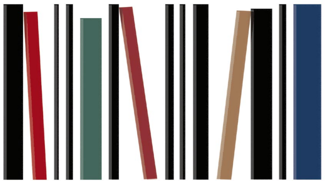
Managing Complexity in a Distributed Digital Library



With multiple collections, languages, and media, digital libraries are becoming more difficult to maintain and develop. Researchers at New Zealand's University of Waikato have developed a software architecture that deals with this complexity.

Ian H. Witten
Rodger J. McNab
Steve Jones
Mark Apperley
David Bainbridge
Sally Jo Cunningham
University of Waikato

As the capabilities of distributed digital libraries increase, managing organizational and software complexity becomes a key issue. How can collections and indexes be updated without impacting queries currently in progress? When users can communicate with the system in different natural languages, how can software cope with the display of various versions of text? With multimedia collections, how can a uniform, extensible software framework accommodate different kinds of search engines required to search various media types? How can the system handle several user-interface clients for the same collections? How do these clients learn about new collections created in different physical locations?

We have developed a software structure that successfully manages this complexity in our own digital library. It dynamically handles additions of new collections in different locations without requiring a centralized list. Its novel, flexible macro language manages interfaces to many collections in many languages, and the structure allows development of experimental user interfaces. It accommodates radically different search engines within a uniform structure and communication protocol. The architecture has met our goal of minimizing maintenance while allowing our digital library to easily expand its offerings.

THE NEW ZEALAND DIGITAL LIBRARY

The New Zealand Digital Library (http://www.nzdl.org) currently offers about 20 collections, varying in size from a few documents up to 10 million documents and several gigabytes of text. These collections

- use different browsing and indexing structures;
- are accessible by computers in different locations;
- support interaction in a variety of languages; and
- contain text, graphics, and audio components.

Some, such as our music collection, require special retrieval engines for nontextual information.

Many collections are updated automatically every few days, but a few—such as the Oxford Text Archive—rarely change. Most collections are created from electronic source texts, but some—notably the Historic New Zealand Newspaper Collection—require scanning and the use of optical character recognition (OCR) to capture the source material. Some collections and their contents are well structured and lend themselves well to browsing, but many have little or no easily discernable structure.

Multiple languages

The library's documents are written in many different languages, including English, French, German, Arabic, Maori, Portuguese, and Swahili. Many collections contain several languages, as do some documents. For example, the Historic New Zealand Newspaper Collection, illustrated in Figure 1, is bilingual, with parallel Maori and English articles.

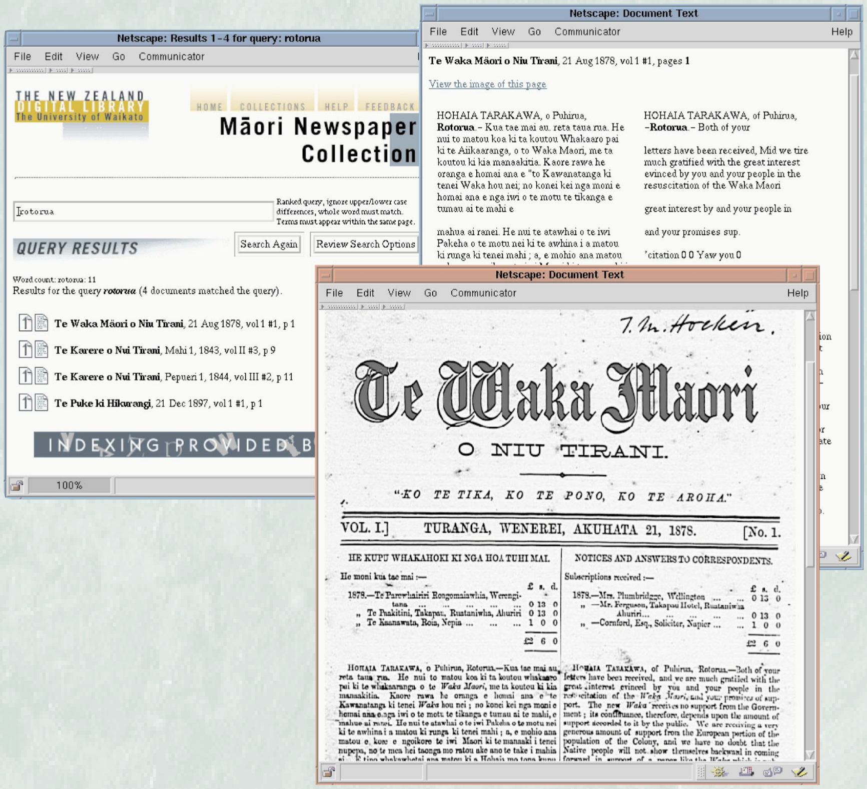
The library also provides interfaces to the collections in several languages. For example, the Computer Science Technical Reports collection has English, French, German, and Maori interfaces. The Arabic interface to the Arabic Library, shown in Figure 2, requires storage and search mechanisms that can handle non-ASCII alphabets. Because the code that produces Web pages is divorced from text strings appearing on them, new languages can be added by merely translating all phrases and menu terms used in the interface.

To accommodate blind users (with speech synthesizers) and partially sighted users (with large-font displays), we provide text-only versions of the interface for each language. For this, we follow standard guidelines for Web-page accessibility.[1]

Multiple media

Some collections contain more than just text. For example, the Local Oral History Collection includes recorded interviews and photographs that illustrate them. Users query summary transcripts. Figure 3 shows results for a query about VE Day.

0018-9162/99/$10.00 © 1999 IEEE


Netscape: Results 1-4 for query: rotorua
File Edit View Go Communicator
THE NEW ZEALAND DIGITAL LIBRARY The University of Waikato
HOME COLLECTIONS HELP FEEDBACK
Māori Newspaper Collection

rotorua

Ranked query, ignore upper/lower case differences, whole word must match. Terms must appear within the same page.

Search Again
Review Search Options
QUERY RESULTS

Word count: rotorua: 11

Results for the query *rotorua* (4 documents matched the query).

Te Waka Māori o Niu Tirani, 21 Aug 1878, vol 1 #1, p 1

Te Karere o Nui Tirani, Mahi 1, 1843, vol II #3, p 9

Te Karere o Nui Tirani, Pepueri 1, 1844, vol III #2, p 11

Te Puke ki Hikurangi, 21 Dec 1897, vol 1 #1, p 1

Te Waka Māori o Niu Tirani, 21 Aug 1878, vol 1 #1, pages **1**

View the image of this page

HOHAIA TARAKAWA, o Puhirua, **Rotorua**.– Kua tae mai au. reta taua rua. He nui to matou koa ki ta koutou Whakaaro pai ki te Aiikaaranga, o to Waka Maori, me ta koutou ki kia manaakitia. Kaore rawa he oranga e homai ana e "to Kawanatanga ki tenei Waka hou nei; no konei kei nga moni e homai ana e nga iwi o te motu te tikanga e tumau ai te mahi e

mahua ai ranei. He nui te atawhai o te iwi Pakeha o te motu nei ki te awhina i a matou ki runga ki tenei mahi ; a, e mohio ana matou

HOHAIA TARAKAWA, of Puhirua, –**Rotorua**.– Both of your

letters have been received, Mid we tire much gratified with the great interest evinced by you and your people in the resuscitation of the Waka Maori

great interest by and your people in

and your promises sup.

'citation 0 0 Yaw you 0

Te Waka Maori

O NIU TIRANI.

"KO TE TIKA, KO TE PONO, KO TE AROHA."

VOL. I.] TURANGA, WENEREI, AKUHATA 21, 1878. [No. 1.

HE KUPU WHAKAHOKI KI NGA HOA TUHI MAI.

He moni kua tae mai :—

£ s. d.

1878.—Te Parewhairiri Rongomaiawhia, Werengitana 0 13 0

" Te Paakitini, Takapau, Ruataniwha, Ahuriri 0 13 0

" Te Kaanawata, Roia, Nepia 1 0 0

£2 6 0

HOHAIA TARAKAWA, o Puhirua, Rotorua.—Kua tae mai au reta taua rua. He nui to matou koa ki ta koutou whakaaro pai ki te whakaaranga o te *Waka Maori*, me ta koutou ki kia manaakitia. Kaore rawa he oranga e homai ana e te Kawanatanga ki tenei *Waka* hou nei ; no konei kei nga moni e homai ana e nga iwi o te motu te tikanga e tumau ai te mahi, e mahue ai ranei. He nui te atawhai o te iwi Pakeha o te motu nei ki te awhina i a matou ki runga ki tenei mahi ; a, e mohio ana matou e, kore e ngoikore te iwi Maori ki te manaaki i tenei nupepa, no te mea hei taonga mo ratou ake ano te take i mahia ai.

NOTICES AND ANSWERS TO CORRESPONDENTS.

Subscriptions received :—

£ s. d.

1878.—Mrs. Plumbridge, Wellington 0 13 0

" —Mr. Ferguson, Takapau Hotel, Ruataniwha Ahuriri... ... ... 0 13 0

" —Cornford, Esq., Solicitor, Napier 1 0 0

£2 6 0

HOHAIA TARAKAWA, of Puhirua, Rotorua.—Both of your letters have been received, and we are much gratified with the great interest evinced by you and your people in the resuscitation of the *Waka Maori*, and your promises of support. The new *Waka* receives no support from the Government ; its continuance, therefore, depends upon the amount of support accorded to it by the public. We are receiving a very generous amount of support from the European portion of the population of the Colony, and we have no doubt that the Native people will not show themselves backward in coming forward in support of a paper like the *Waka* which is

Figure 1. Searching the Historic New Zealand Newspapers Collection. The collection is bilingual; all articles are retrieved in Maori and English.

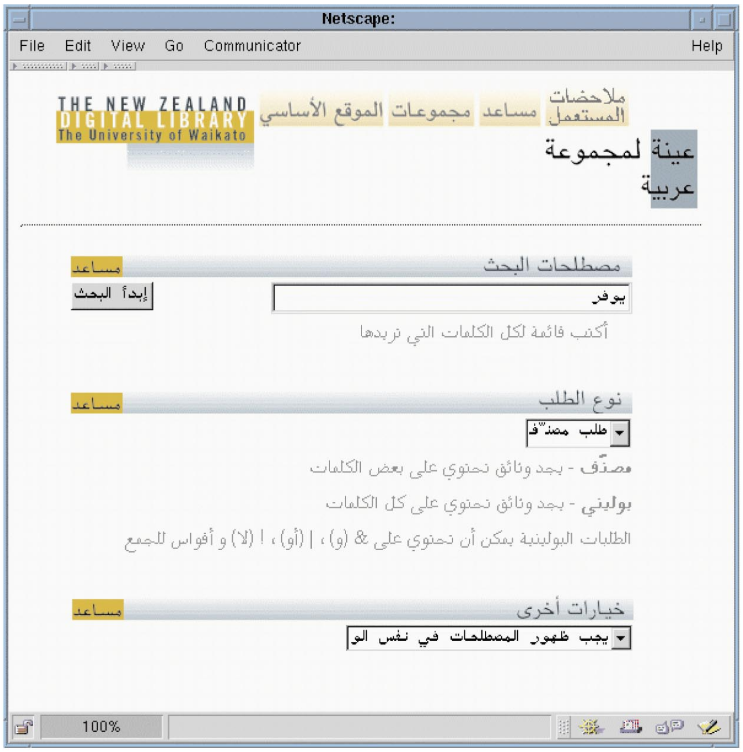
The Music Library demonstrates a technically more significant multimedia capability. Users can sing or hum a tune, then upload their audio file to find a match in a tune database. Figure 4 shows the results when a user sings the first eight notes of "Auld Lang Syne." The transcribed input appears at the top left, and titles of similar items, ranked according to how closely they match the query, appear below. Any of the tunes may be selected for audio replay or visual display, as shown on the right.

The Music Library database includes nearly 10,000 folk tunes from North America, Ireland, Britain, Germany, and China. Users can search the entire database or limit their searches to individual countries. The Music Library is a good example of how the digital library's single unifying architecture can encompass radically different search regimes.

SYSTEM DESIGN

Our design is based on *collections*—sets of like *documents*.[2] These documents come in a variety of formats: plain ASCII, PostScript, PDF, HTML, SGML, and Microsoft Word for textual documents; Refer, BibTex, and USMarc for bibliographic information; and various formats for graphics and sound.

Collections invariably undergo a *building process* to make them suitable for search, retrieval, and display. This often involves converting the documents to a different format and identifying subparts that require their own indexes. Among the current collections are indexes for complete documents, individual pages, paragraphs, articles, abstracts, authors, titles, subjects, publication details, references, and motions in meetings.


Netscape:
File Edit View Go Communicator Help
THE NEW ZEALAND DIGITAL LIBRARY The University of Waikato
الموقع الأساسي
مجموعات
مساعد
ملاحضات المستعمل
عينة لمجموعة عربية

يوفر

أكتب قائمة لكل الكلمات التي تريدها

طلب مصنّف

مصنّف - يجد وثائق تحتوي على بعض الكلمات

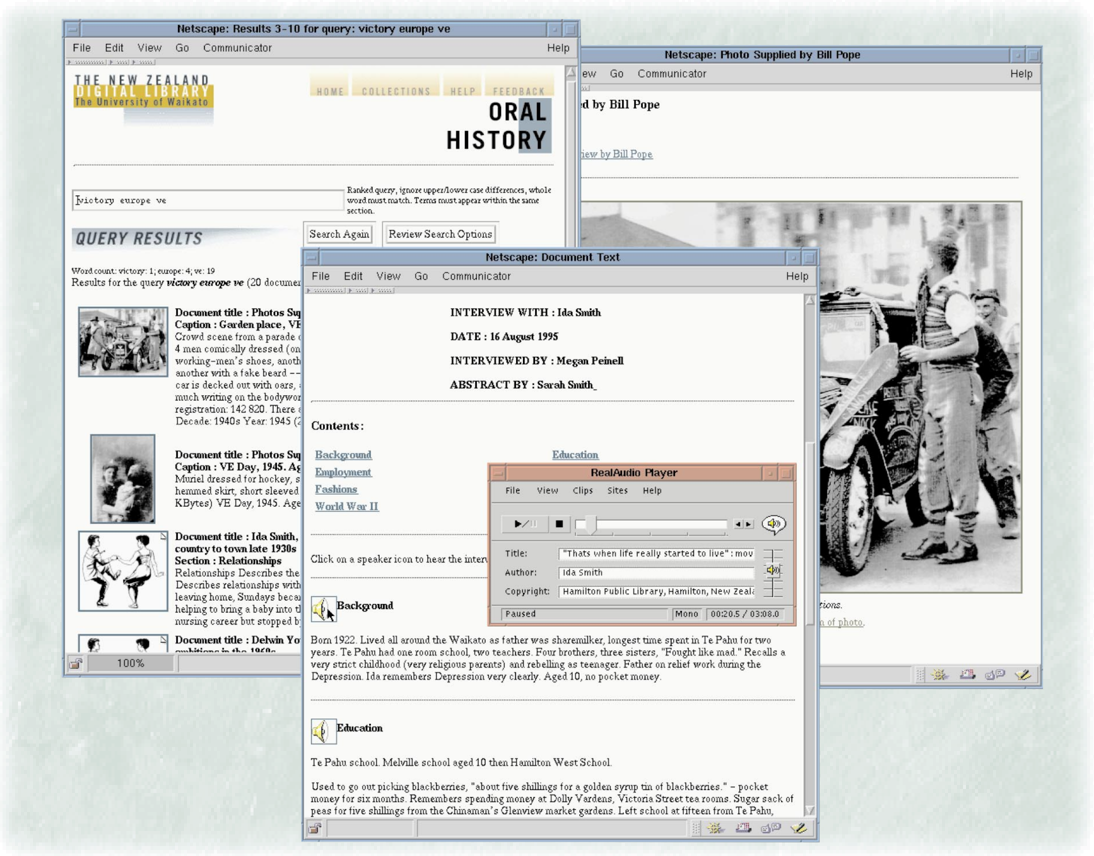
بوليني - يجد وثائق تحتوي على كل الكلمات

الطلبات البولينية يمكن أن تحتوي على & (و)، | (أو)، ! (لا) و أقواس للجمع

يجب ظهور المصطلحات في نفس الو

Figure 2. Arabic interface for the Arabic Library.

Collections are *rebuilt* when new documents are added to them. To avoid disrupting active users, rebuild-


Netscape: Results 3-10 for query: victory europe ve
File Edit View Go Communicator Help
THE NEW ZEALAND
DIGITAL LIBRARY
The University of Waikato
HOME COLLECTIONS HELP FEEDBACK
ORAL HISTORY
Search Again
Review Search Options
QUERY RESULTS
100%
Netscape: Photo Supplied by Bill Pope
ew Go Communicator Help
Netscape: Document Text
File Edit View Go Communicator Help
RealAudio Player
File View Clips Sites Help

victory europe ve

Ranked query, ignore upper/lower case differences, whole word must match. Terms must appear within the same section.

Word count: victory: 1; europe: 4; ve: 19
Results for the query *victory europe ve* (20 documen

Document title : Photos Su
Caption : Garden place, VE
Crowd scene from a parade
4 men comically dressed (on
working-men's shoes, anoth
another with a fake beard --
car is decked out with oars,
much writing on the bodywor
registration: 142 820. There
Decade: 1940s Year: 1945 (2

Document title : Photos Su
Caption : VE Day, 1945. Ag
Muriel dressed for hockey, s
hemmed skirt, short sleeved
KBytes) VE Day, 1945. Age

Document title : Ida Smith,
country to town late 1930s
Section : Relationships
Relationships Describes the
Describes relationships with
leaving home, Sundays beca
helping to bring a baby into t
nursing career but stopped b

Document title : Delwin Yo

d by Bill Pope

iew by Bill Pope

ions.

n of photo.

INTERVIEW WITH : Ida Smith

DATE : 16 August 1995

INTERVIEWED BY : Megan Peinell

ABSTRACT BY : Sarah Smith_

Contents:

Background
Employment
Fashions
World War II
Education

Click on a speaker icon to hear the interv

Title: "Thats when life really started to live" : mov
Author: Ida Smith
Copyright: Hamilton Public Library, Hamilton, New Zeala
Paused | Mono | 00:20.5 / 03:08.0

Background

Born 1922. Lived all around the Waikato as father was sharemilker, longest time spent in Te Pahu for two years. Te Pahu had one room school, two teachers. Four brothers, three sisters, "Fought like mad." Recalls a very strict childhood (very religious parents) and rebelling as teenager. Father on relief work during the Depression. Ida remembers Depression very clearly. Aged 10, no pocket money.

Education

Te Pahu school. Melville school aged 10 then Hamilton West School.

Used to go out picking blackberries, "about five shillings for a golden syrup tin of blackberries." – pocket money for six months. Remembers spending money at Dolly Vardens, Victoria Street tea rooms. Sugar sack of peas for five shillings from the Chinaman's Glenview market gardens. Left school at fifteen from Te Pahu,

Figure 3. Results of a query about VE Day from the Local Oral History Collection. The small gray panel controls audio playback of a sound bite or relevant section from a participant's interview.

ing takes place in a separate file directory, and when it is complete, the directory is moved in one fell swoop to the live file area. The query engine automatically manages the transition to the new version. Current users experience only a brief delay while the query process reinitializes. References on a query results page, however, may become incorrect after the update occurs. To resolve this, we could simply warn users, but our plan is to instead use persistent document identifiers.

Normally, several collections operate on the same server machine—by convention in one directory—and each server machine runs its own copy of the collection server software. Collections are active objects, receiving and processing messages, such as search requests and document display requests. Therefore, they can be instructed to look for other collections in their own directory—which avoids a high-maintenance centralized list. Adding a new collection merely involves inserting it into a directory with an already known collection.

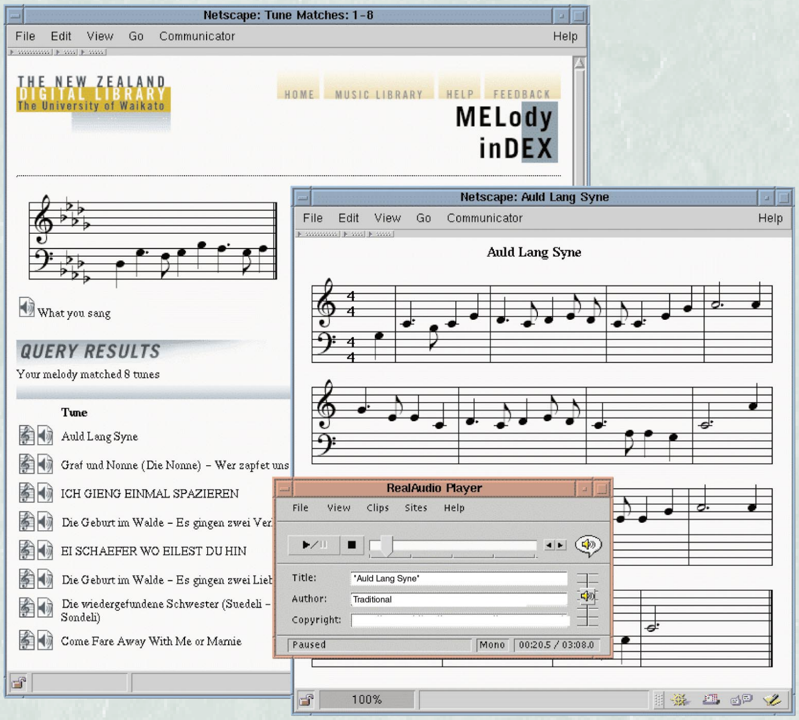
Whenever a user-interface client starts up (and periodically while it is running), it messages collections on all machines, requesting information on other collections in their directories. When a collection is added, all interface clients will eventually realize it. This automatic notification process greatly simplifies the process of adding new collections.

Managing complexity

Managing the complexity of multiple collections, multiple languages, and multiple interface options presents a significant challenge. For example, document items that have not yet been translated to other languages need to default to English. Non-ASCII languages like Arabic and Chinese need special text positioning and justification. Text-only pages require a structure different from corresponding text-and-graphics pages. Page headers and footers must allow some collections to override them. Directory names and URLs should be referenced symbolically, facilitating the system's configuration to different environments.

For easily maintained, easily augmented, and consistent interfaces to collections, commonality must be abstracted out and represented in just one place. We design Web pages, then, with a model based on structure, content, and reusable items:

- The *structure* allows different versions of the content. For example, a page might contain some sort of heading, a main information section, and a footer with navigation aids. The heading in turn might have its own structure, with masthead and navigation aids at the top and an advertisement underneath.
- The *content* comprises the principal information elements of the page—text, graphics, sound clips, and video—separated from the page structure in separate resources. Many content items are language-specific, and all resources pertaining to a particular language are collected together for easy maintenance.
- *Reusable items* include pathnames of certain directories and frequently used icons. Isolating these items renders Web pages more portable.


Netscape: Tune Matches: 1–8
File Edit View Go Communicator Help
THE NEW ZEALAND DIGITAL LIBRARY The University of Waikato
HOME MUSIC LIBRARY HELP FEEDBACK
MELody inDEX
What you sang
QUERY RESULTS
Netscape: Auld Lang Syne
File Edit View Go Communicator Help
RealAudio Player
File View Clips Sites Help
100%

Your melody matched 8 tunes

Tune
- Auld Lang Syne
- Graf und Nonne (Die Nonne) – Wer zapfet uns
- ICH GIENG EINMAL SPAZIEREN
- Die Geburt im Walde – Es gingen zwei Ver
- EI SCHAEFER WO EILEST DU HIN
- Die Geburt im Walde – Es gingen zwei Lieb
- Die wiedergefundene Schwester (Suedeli – Sondeli)
- Come Fare Away With Me or Mamie

Auld Lang Syne

Title: "Auld Lang Syne"
Author: Traditional
Copyright:
Paused | Mono | 00:20.5 / 03:08.0

Figure 4. Using the Music Library to find "Auld Lang Syne."

We meet these requirements using a flexible, specially designed macro language. While standard macro languages are often used to maintain large collections of Web pages,[3] such languages only support different page versions through the conditional inclusion of text. This quickly becomes cumbersome as the number of options increases.

Our macro language, however, allows many variants of the same macro, each with different parameter values. Our three basic parameters are collection name, text-and-graphics or text-only format, and language. At expansion time, the macro processor searches the database for a variant matching all three parameter values for the generated page's environment. If it does not find an exact match, it uses the closest match based on each parameter's priority. For example, if a user seeks a German text-and-graphics format, but the database contains only English and German text-only versions, the processor calls up the German text-only version.

Macros also support essential maintenance operations. Each week an automatic process will scan the macro, locate those that have been flagged as needing translations, and mail the material to appropriate translators.

Facilitating searches

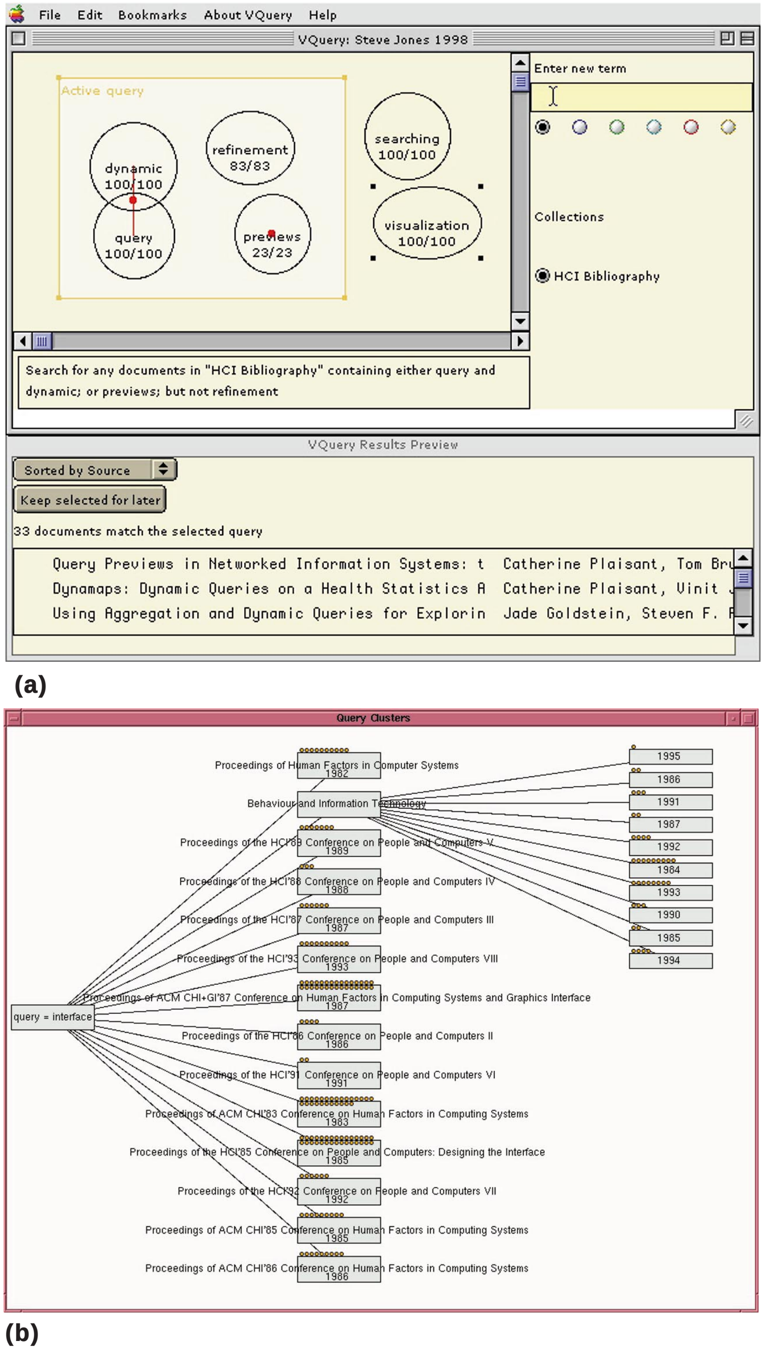
Because collections run on different computers, our system is distributed, but searching is not distributed: Each collection handles its own searches. Cross-collection searching would be relatively easy for collections on the same server, but searching across servers would involve a more difficult distributed search process (which is addressed to an extent by some information retrieval systems[4]). Our goal, however, has not been to advance the state of the art in information retrieval, but to provide a flexible structure that can accommodate the latest advances.

While searches in some multimedia collections can be done on textual descriptions of the media objects, different media types generally require different search mechanisms. Presently the system includes two search and retrieval methods—one for text and the other for music. Our uniform architecture accommodates them and their vastly different document types. This structure is flexible, allowing us to incorporate other retrieval mechanisms in the future—for example, one for accessing an image collection through content-based queries.[5]

For text, we use MG, a full-text retrieval system that efficiently stores text and its index in compressed form.[6] MG typically compresses text to about 25 percent of its original size and compresses indexes to about 7 percent of the original text's size—making the total storage requirement about one-third of the original text's size.

For music, we use MR, a novel scheme for searching musical melodies.[7] MR matches sung (or hummed) input to a database of tunes, with various user options:

- Restricting attention to subsets of the database
- Choosing one of two matching algorithms
- Matching anywhere within a tune or to beginnings only
- Matching using musical intervals or coarser up-down-same pitch differential contour
- Ignoring or taking into account note duration, transcribing using fixed tuning, or trying to adapt to overall drift in the user's intonation
- Specifying minimum rest and note lengths
- Specifying music speed


File
Edit
Bookmarks
About VQuery
Help
VQuery: Steve Jones 1998
Active query
dynamic 100/100
refinement 83/83
searching 100/100
query 100/100
previews 23/23
visualization 100/100
Enter new term
Collections
HCI Bibliography
VQuery Results Preview
Sorted by Source
Keep selected for later
(a)
Query Clusters
query = interface
Proceedings of Human Factors in Computer Systems 1982
Behaviour and Information Technology
Proceedings of the HCI'89 Conference on People and Computers V 1989
Proceedings of the HCI'88 Conference on People and Computers IV 1988
Proceedings of the HCI'87 Conference on People and Computers III 1987
Proceedings of the HCI'93 Conference on People and Computers VIII 1993
Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface 1987
Proceedings of the HCI'86 Conference on People and Computers II 1986
Proceedings of the HCI'91 Conference on People and Computers VI 1991
Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems 1983
Proceedings of the HCI'85 Conference on People and Computers: Designing the Interface 1985
Proceedings of the HCI'92 Conference on People and Computers VII 1992
Proceedings of ACM CHI'85 Conference on Human Factors in Computing Systems 1985
Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems 1986
1995
1986
1991
1987
1992
1984
1993
1990
1985
1994
(b)

Search for any documents in "HCI Bibliography" containing either query and dynamic; or previews; but not refinement

33 documents match the selected query

```
Query Previews in Networked Information Systems: t  Catherine Plaisant, Tom Bru
Dynamaps: Dynamic Queries on a Health Statistics A  Catherine Plaisant, Vinit
Using Aggregation and Dynamic Queries for Explorin  Jade Goldstein, Steven F. R
```

Figure 5. Two experimental interfaces to the New Zealand Digital Library. (a) A Java applet provides a graphical language for Boolean querying. (b) A Tcl/Tk application visualizes clusters of documents that are the result of a sequence of queries.

Many of these options are needed because users differ in their musical ability and in the accuracy with which they remember tunes.

The process of building a collection is heavily dependent on the mechanisms used to search it because so much of the building involves creating an indexing structure. Consequently, for each collection there is a *builder process*, which performs the offline activity of creating the index and other supporting structures, and a corresponding *collection server process*, which manages user interaction.

To accommodate different index structures, often with widely differing needs, the builder and server processes use a flexible, object-oriented program structure. Object classes with all the basic features needed for building, document retrieval, and searching are defined, but the functionality they provide is selectively overridden by defining subclasses for particular search and retrieval systems. These defining subclasses are themselves finally overridden by collection-specific subclasses that accomplish any task peculiar to a collection.

Improving interfaces

Internet search engines and online public-access library catalogs commonly provide textual query languages like Z39.58,[8] but such interfaces present problems for casual users, such as the need to learn the syntax of the language. Using various implementation platforms, we are developing several highly interactive, graphical, and direct manipulation interfaces to the library.

Figure 5a shows one example, a Java applet that provides a graphical language for Boolean querying and updates itself immediately as the query is modified. Figure 5b shows another, a Tcl/Tk application for visualizing clusters of documents resulting from a sequence of queries. A communication protocol defines how user-interface software communicates with collections through network connections established with collection servers.[2] These interfaces have been developed without modifying the collection server software and without close code integration between the user interface and the server.

Collection servers recognize three message types:

- Requests for general information, such as details of available collections
- Requests for document information, such as summaries or the full text of documents
- Requests for documents matching specified search criteria

User interface clients do not necessarily operate on any collection's host machine, and our Java implementations are independent of platform. We have developed an application programmer's interface to a software library that contains routines to form and send requests to collections and manage the resulting responses. This makes it easy to integrate new interfaces, without any impact on collection servers. Both the standard Web interface and our experimental interfaces exploit the protocol, and different interfaces can coexist without any difficulty.

Dynamic query interfaces—in which displayed document-result sets change immediately whenever query parameters change—form the focus of our user interface research. Here, operations must be executed quickly on locally cached document information. The protocol allows close control over when information is sent from the collection server to the cache and how much is transferred. For example, in bibliographic collections, items such as author, title, and keywords can be retrieved individually for single documents. Only the information that is desired is retrieved over the network. This improves response time, minimizes the size of the local data store, and allows information to be displayed incrementally as it arrives. Users can continue interacting while information is still arriving.

Our system successfully manages organizational and software complexity in a large, fully operational, and widely used digital library system. Our fundamental goal has been to minimize the effort required to keep the system operational and yet continue to expand its offerings. We can now easily build new collections with distinctive features and indexes—adding them at any time, on any existing server, while the system is running. We can accommodate new languages by translating phrases and images used in the interface and incorporating them into the collection server software. We can easily incorporate multiple media and initiate nonstandard search engines for them. We can apply new, highly interactive interfaces, implemented on the client side and using a variety of support technologies. The structure we have developed makes it easy to keep a digital library working and growing. ❖

Acknowledgments

We gratefully acknowledge the help of Carl Gutwin, who developed one of the experimental user interfaces. Many thanks are also due to Stefan Boddie, Te Taka Keegan, Craig Nevill-Manning, and Lloyd Smith, who contributed to this work in various ways.

References

1. *Unified Web Site Accessibility Guidelines*, http://www.w3.org/WAI/GL (current Dec. 2, 1998).
2. R.J. McNab, I.H. Witten, and S.J. Boddie, “A Distributed Digital Library Architecture Incorporating Different Index Styles,” *Proc. Advances in Digital Libraries ’98*, IEEE CS Press, Los Alamitos, Calif., 1998, pp. 36-45.
3. A. Peel, *HTML Macros: Easing the Construction and Maintenance of Web Texts*, Tech. Report 4-96, Univ. of Kent, Canterbury, England, 1996.
4. O. de Kretser et al., “Methodologies for Distributed Information Retrieval,” *Proc. Int’l Conf. Distributed Computing Systems*, IEEE CS Press, Los Alamitos, Calif., 1998, pp.66-73.
5. J.R. Smith and S.-F. Chang, “VisualSEEK: A Fully Automated Content-Based Image Query System,” *Proc. ACM Multimedia Conf.*, ACM Press, New York, 1996, pp.87-98.
6. I.H. Witten, A. Moffat, and T.C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*, Van Nostrand Reinhold, New York, 1994.
7. R.J. McNab et al., “Toward the Digital Music Library: Tune Retrieval from Acoustic Input,” *Proc. Digital Libraries ’96*, ACM Press, New York, 1996, pp. 11-18.
8. NISO, *Z39.58: Common Command Language for Online Interactive Information Retrieval*, ANSI/NISO, Bethesda, Md., 1995.

Ian H. Witten is a professor of computer science at the University of Waikato in Hamilton, New Zealand. He directs the New Zealand Digital Library research project. His research interests include information retrieval, machine learning, text compression, and programming by demonstration. He received an MA in mathematics from Cambridge University, England; an MSc in computer science from the University of Calgary, Canada; and a PhD in electrical engineering from Essex University, England. He is a member of the IEEE, and a fellow of the ACM and of the Royal Society of New Zealand.

Rodger J. McNab is a research programmer for the New Zealand Digital Library. His research interests include the architecture of digital libraries, indexing, and digital music libraries. He received a BCMS and an MCMS in computer science from the University of Waikato.

Steve Jones is a lecturer in computer science at the University of Waikato. His research interests include digital libraries, World Wide Web navigation, and computer-supported collaborative work, particularly collaborative writing and information retrieval. He received his PhD in computer science at the University of Stirling, Scotland.

Mark Apperley is a professor of computer science and chair of the Computer Science Department at the University of Waikato. His research focuses on human-computer interaction, and his publications address topics ranging from dialogue design notations to interaction devices and computer-supported collaborative work. Apperley is a fellow of the New Zealand Computer Society and a member of the IEEE Computer Society and of the ACM.

David Bainbridge is a lecturer in computer science at the University of Waikato. His research interests are computer music and digital libraries. He received a BEng in engineering at the University of Edinburgh and a PhD in optical music recognition at the University of Canterbury, New Zealand.

Sally Jo Cunningham is a senior lecturer at the University of Waikato. Her research interests include digital libraries, machine learning, and computing education. She earned a PhD in computer science, with minors in Asian history and library science, from Louisiana State University.

Contact Witten, McNab, Jones, Apperley, Bainbridge, and Cunningham at {ihw, rjmcnab, s.jones, m.apperley, d.bainbridge, s.cunningham}@cs.waikato.ac.nz.