WAIS: The Wide Area Information Server or Anonymous What???

Peter Marshall <peter@julian.uwo.ca>
Manager, Academic Networking
Computing and Communications Services
The University of Western Ontario

August 29, 1992

Abstract

Anonymous FTP isn't a tool that most users find easy or natural to use. Finding out *where* to retrieve data and then using Anonymous FTP to actually *do* the retreival is just too difficult. Other mechanisms need to be developed to make accessible the information available on the Internet. One approach to this problem is the Wide Area Information Server or WAIS[1]. WAIS is a developing network application that allows queries to be made of multi-media (but usually text-based) databases using a standard query and retrieval protocol (Z39.50). One of its great benefits is that it provides a common user interface to a wide range of information sources that can be resident anywhere on the Internet. WAIS provides good mechanisms to flexibly handle diverse and unstructured data. It also encourages the data to reside in a single place, "close" to its maintainer, thus allowing near-realtime updates.

Currently, over 225 publicly registered databases have been made available by Internet sites from around the world. These databases are diverse, instantly searchable and retrievable.

WAIS is an early example of the kind of network applications that will help make the Internet a useful resource for non-computing-oriented users. Its easy to use "natural language" access makes it much more "user-friendly" than many network applications.

[1]Pronounced "ways".

WAIS also is a powerful tool that allows information providers to enter the world of electronic publishing. We all generate mountains of text within our organizations. WAIS provides some excellent facilities to make that text available in a reasonable and useful way.

The University of Western Ontario has implemented a number of WAIS databases both for internal and external consumption. Through the experiences at UWO with WAIS, this paper explores the concepts of network-searchable databases and the WAIS implementation in particular. It introduces the WAIS system, touches on related projects like "gopher" and other text retrieval systems, presents a picture of the current state of the WAIS community and discusses current problems and limitations in the software. The paper concludes with an examination of the future for information servers.

1 Introduction

When I first started thinking about this paper, I was reminded of an incident from my distant computing past. It was in the early seventies when Geoff Collyer (some of you may know him as one of the authors of C-news) invited me down into the dark and eerie basement of St Joseph's Hospital in London to check out an interesting computer. He was doing some work on a PDP-11 system for Nuclear Medicine and was running Unix on it. He gave me a quick overview and then as system managers always do with novice users, told me that all the commands were in a directory named `/bin` and left me alone.

Geoff had left me with one perplexing concept: the pipe. "Pipes", he told me, "were one of the powerful features of Unix." But, working there on my own I really couldn't figure out what Geoff was talking about. Once I had mastered the idea of a pipe (a long time later) I realized that one of the difficulties that I had had with pipes was that they were represented by the strange (to me) vertical bar symbol: "`|`". I (like most machines of the time) had been upper-case oriented and this was a very new part of the keyboard. I just couldn't get past the symbol to its meaning.

I think that to a large extent there is a similar conceptual problem with anonymous FTP. *Anonymous* isn't the easiest word in the world to spell. It isn't even that easy to pronounce. I think that its choice may have had a more profound effect on the development of networks than is generally realized.

Nevertheless, difficulties with using networks have spawned a great deal of Internet activity to try to make the access of computer-based information simpler, easier and more natural. I suppose it could be argued that it might have been easier to teach people to spell!

2 WAIS Overview

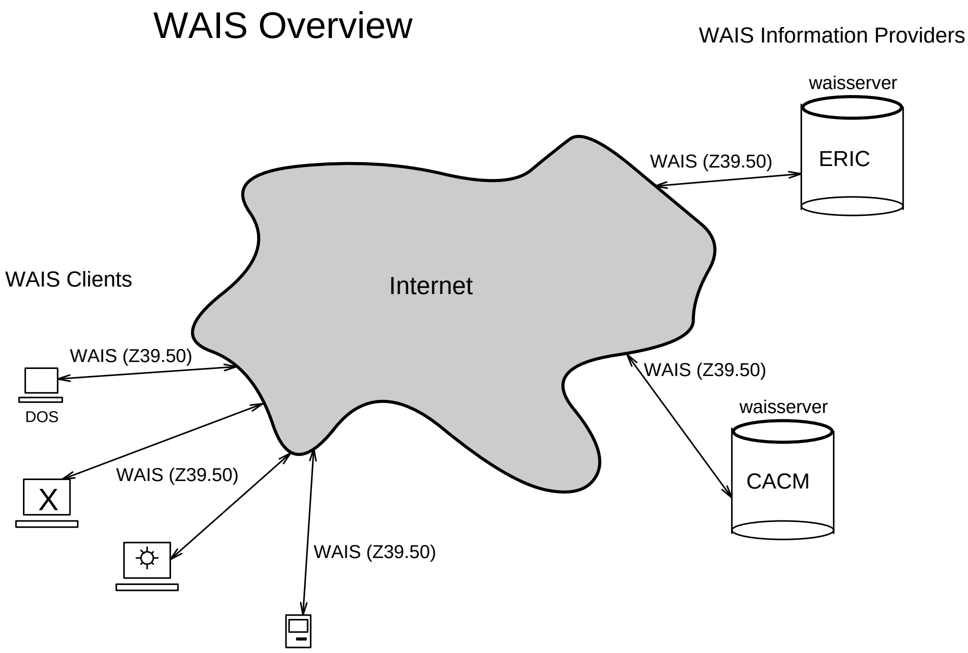

Figure 1: WAIS Overview

One of those attempts at making network-available information more accessible is the WAIS project begun by Brewster Kahle at Thinking Machines Corporation[2].

The basic idea is to separate the information-provider process from the information-seeker process and define a protocol that permits these two to communicate. (See figure 1 on page 3.) WAIS can be thought of in the abstract sense as the protocol[3] between the client and the server. More

[2] Thinking Machines manufactures and sells the massively parallel Connection Machine computers.

[3] The protocol that links these two is an extended version of ANSI (or its successor) Z39.50 1988. Z39.50 was defined as a common protocol for querying library catalogues.

concretely, you can think of WAIS as encompassing both a server process and a client process as well as the mechanism used to communicate between them. The server does the searching while the client is used to compose searches and display the results.

To get the whole project off the ground and to prove the concept, Thinking Machines wrote sample client and server software and made them publicly available during the spring of 1991. Current WAIS implementations (no longer all from Thinking Machines) include clients for

- Unix: shell commands; curses-based screens; X-windows; NeXtstep
- MS-DOS: packet-drivers; Windows (real memory hogs)
- Macintosh
- VMS: Wollongong; Multinet; Digital's Ultrix Connection

Servers are available for Unix systems, Connection Machines, VMS and soon Macintoshes. Most of this software is based on the free-ware implementations. While this code is still considered to be in "beta" test it is reasonably bug free.

Thinking Machines is also involved in some more commercial ventures like Dow Jones News Retrieval. Dow Jones has recently implemented a service on their private network that uses a Connection Machine implementation of WAIS for searching.

3 What's Out There?

The free software approach has proven to be a very successful one. Currently there are more than 225 publicly registered WAIS databases on the Internet. The following is just a very small sampling of what is currently being offered (I've included some sample questions to help give some idea about the contents.):

- ERIC (Educational Resources Information Center) Digests: Does taking notes help students in information retention?
- Communications of the ACM (full-text): Show me that recent article on user interfaces to computer programs.

The WAIS community is actively involved in the 1992 version of Z39.50 specification and future WAIS clients and servers will probably not require extensions.

- Weather satellite pictures (updated every hour): Show me the current ground conditions in North America.
- Human DNA sequences (updated as they are discovered).
- The Gutenberg Project archives: How about checking out a copy of Brontë's *Wuthering Heights*?
- Poetry archives: Find me a poem about roses.
- The KJV of the Bible: Who was the guy with all the boils?
- The Koran: Is the Garden of Eden mentioned?
- FTP'able README files: Where would I find a program that converts scribe to TeX?
- An index to journalism periodicals: Does the reporting of John Crosby's speeches pose an ethical dilemma?

For the most part, current servers' information is largely text based. Searches are made using words and the documents are returned as ASCII text. This is not a restriction in the protocol since the documents retrieved can be an arbitrary byte stream. Indeed, the weather-map server provides very detailed and up-to-the-minute satellite and ground condition maps in colour GIF format for automatic display by many WAIS clients. Much work is currently on-going to develop mechanisms for searching and distributing PostScript, SMGL texts and other data formats.

Another measure of the initial success of WAIS is the wide-spread and active use of the existing servers. Recent surveys show 6,000 hosts with an estimated 10,000 users accessing WAIS servers. These users are scattered all over the world and are using not only WAIS client software but also gateways from other systems like the University of Minnesota's gopher.

4 Searching

One of the key elements of the WAIS system is that queries are posed in a non-threatening, very natural way. Rather than expecting users to understand Venn Diagrams and AND, OR and NOT operations, searches are typically performed by asking an English language question.

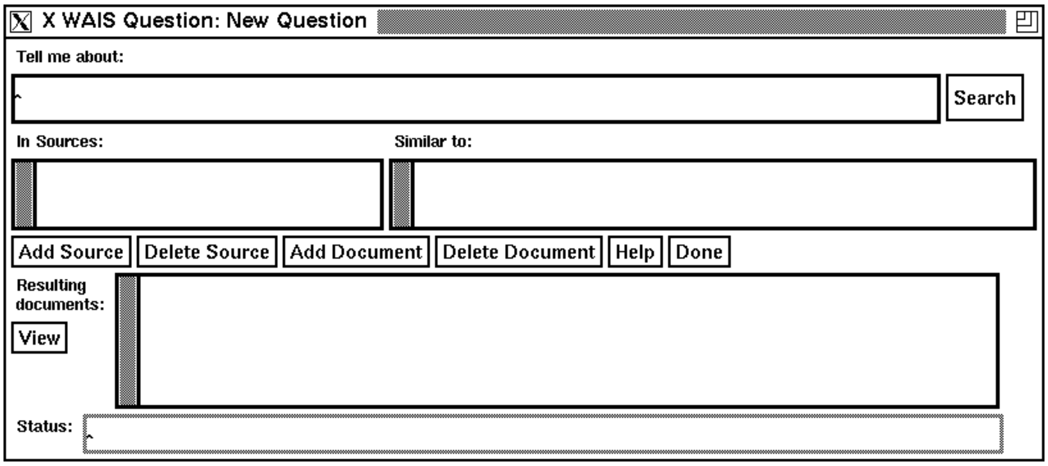

Figure 2: X-Windows WAIS Query

4.1 A Simple Search

The following list is tightly tied to the example illustrated in the following series of figures from the X-windows WAIS client. A general explanation of each step is followed (in parentheses) by the X-windows specific actions.

1. Start up the querying program (See figure 2 on page 6)

2. Choose the database(s) that you think are relevant. (Press Add Source to get a display of the ones currently available. Double click to select a database from the list. Repeat to add multiple databases. See figure 3 on page 7.)

3. Type in your search terms and ask your client to send off the searches to the various servers. (Fill in the Tell me about: field and press the Search button — See figure 4 on page 8.)

4. The server (after a suitable delay) responds by returning a sequence of ranked headlines. Each headline typically includes a ranking (from 1000 to 1), an indication of the size of the document (Do you really want to retrieve all of *Wuthering Heights*?), perhaps a date and then some title information. The ranking is intended to give you some indication about how well a particular document matched your search question. (See figure 5 on page 9.)

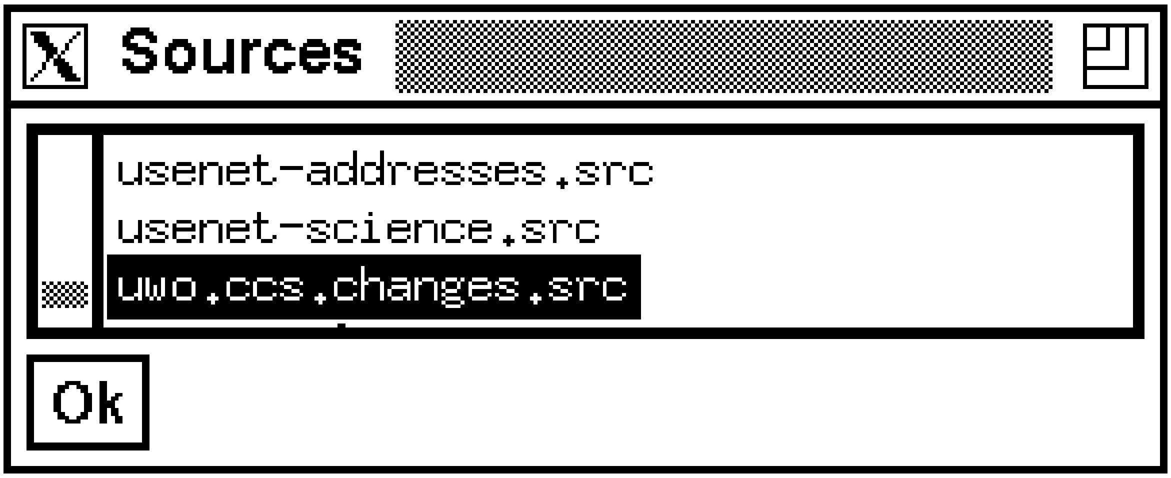

Figure 3: Choose the Database(s)

5. Scan these headlines and choose those appropriate for full text retrieval. (Double click on one of the headlines and a new window pops up (after a suitable delay!) to display the document. See figure 6 on page 10. The text can be read on-line or saved to a file.)

4.2 Relevance Feedback

Searches are not really interpreted as English language constructs. In current implementations the words in the search question are used merely as a list of search terms to be tested against the database. Each occurrence of a search term in the document is counted, perhaps with some weighting and the documents with best scores are ranked near the top.

"But", you may say, "if I can essentially only do a 'term1 OR term2 OR term3' style of search, then how can I ever narrow down the search? Adding extra terms only *widens* the search."

This is handled by a couple of mechanisms: First, the results returned are ranked. Documents that seem to fit your question better get a higher score. This means that queries are not really strings of ORed terms. A much more complex boolean operation is taking place. Secondly, searches can be refined by a process known as "relevance feedback". The idea here is that your first key word search returns a number of "headlines". From

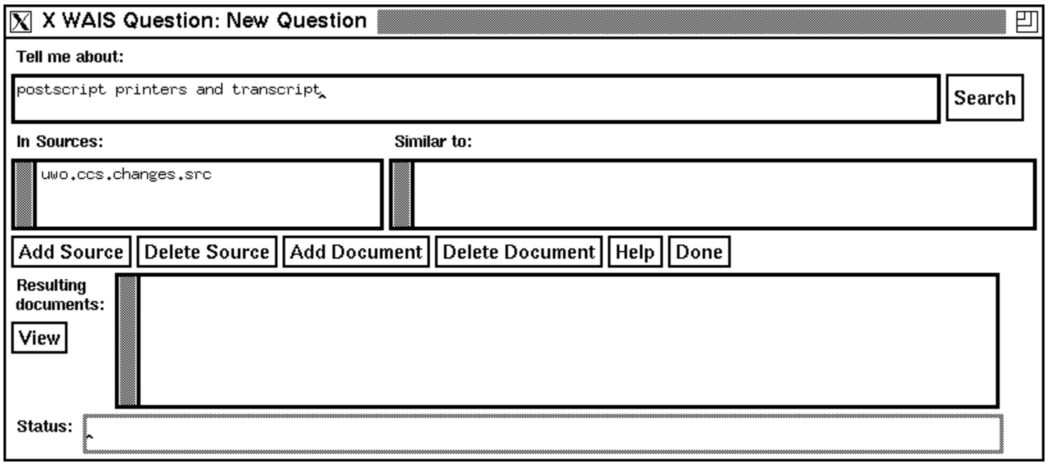

Figure 4: Ask the Question

those headlines, you may be able to choose a document that really does fit your question or you may retrieve a few to see if you can find one that does fit. Once you have located a relevant document you can ask WAIS to find all documents that are similar to *that* document.

When using the X-windows client software, using relevance feedback is as simple as selecting one of the documents retrieved via an ordinary keyword search and then pressing the Add Document button to place it into the Similar to: list. (See figure 7 on page 11.) A new search will then use the selected documents to guide it to a very precise set of documents.

Dow Jones on their DowQuest2 database has found that relevance feedback is a very powerful, yet very easy-to-learn mechanism for searching large databases[4]. Non-computer literate people grasp this concept much more easily than they do boolean algebra! Unfortunately relevance feedback doesn't exist in all WAIS clients yet. Current implementations consider that two documents are similar if they share a large number of common words. Other more intelligent approaches are certainly feasible. Current implementations also don't allow a document from one server to be used on another — another

[4]Brewster Kahle and Art Medlar, *An Information System for Corporate Users: Wide Area Information Servers*, 8 April 1991, Version 3, TMC Tech Report TMC199. A text version of this report is available via anonymous FTP on **think.com** as `/wais/wais-corporate-paper.text`

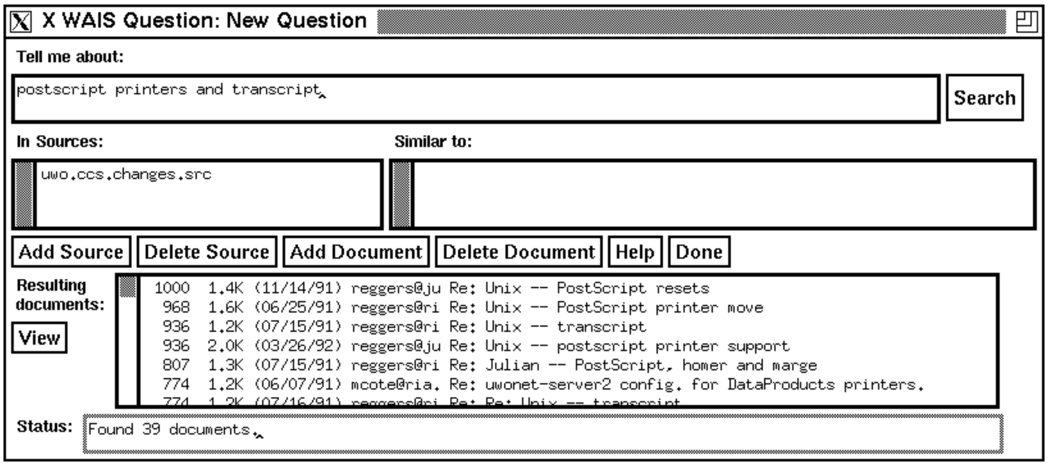

Figure 5: The Returned Results

severe limitation. We have to remember that WAIS is still in its infancy. It is very useful now, but there is still much to be done.

4.3 Searching Multiple Databases

There are a couple of major advantages of using a common protocol like WAIS as the mechanism for communicating with multiple databases. Not only can a single query action on a user's part scan a wide body of information but your results will represent the overall best answers from the entire group of searched information sources. This has the advantage of interspersing answers from a number of sources and rating them on the same scale.

For example, if you got 40 responses from database A and 40 from B it might well turn out that these should be rated such that B's were all better than A's. Using separate searches that used different rating schemes would make such an ordering impossible. With WAIS the proper ranking is automatic.

While current WAIS implementations actually make the connections to each database server sequentially, there is the future possibility of doing the searches in parallel. This could speed up the searching of large numbers of databases.

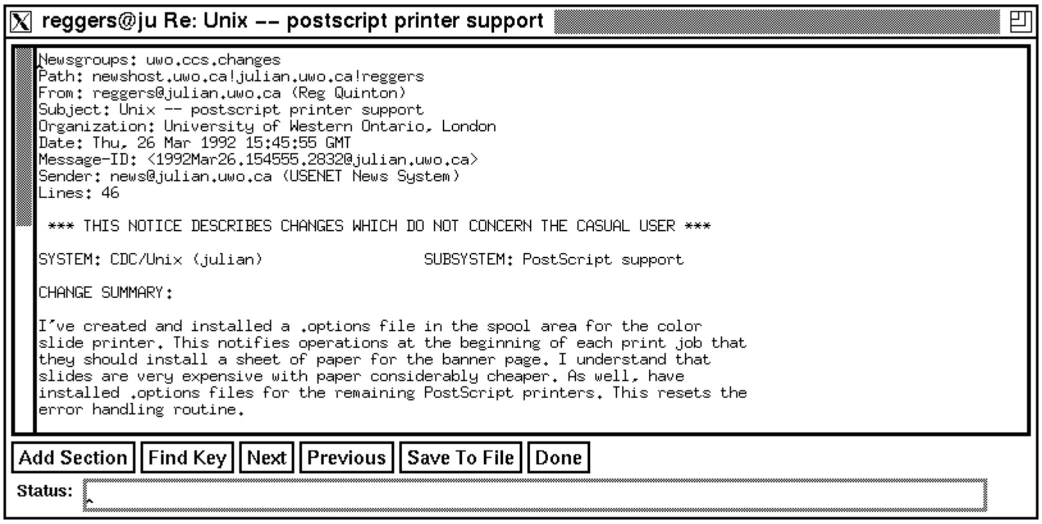

Figure 6: Display a Document

4.4 Finding a Data Source

Up to this point we have assumed that the user just selects the databases to be searched by choosing from a menu. This is certainly a feasible approach while the number of possible database sources is fairly small. Already, with over 200 databases servers now operating, a menu is starting to become difficult to manage. It also means that on every client machine, a copy of the files that point to all the databases must exist — clearly not a scalable approach.

The current approach in the WAIS community is to implement a special server named the `directory-of-servers` which is a WAIS database that contains all of the database description files. These descriptions contain pointer information like IP number and TCP port to use for access to the server and a comment field that is meant to describe the database in a Natural Language like English.

A search now becomes a little more complicated. First a search is made to the `directory-of-servers`. This returns a list of possible database sources. These can be browsed and when a likely one is found, it can be added to your local menu of databases to be searched with the touch of the Add Section button.

The second phase of the search is to select this new database and do the search as outlined in the simple search above.

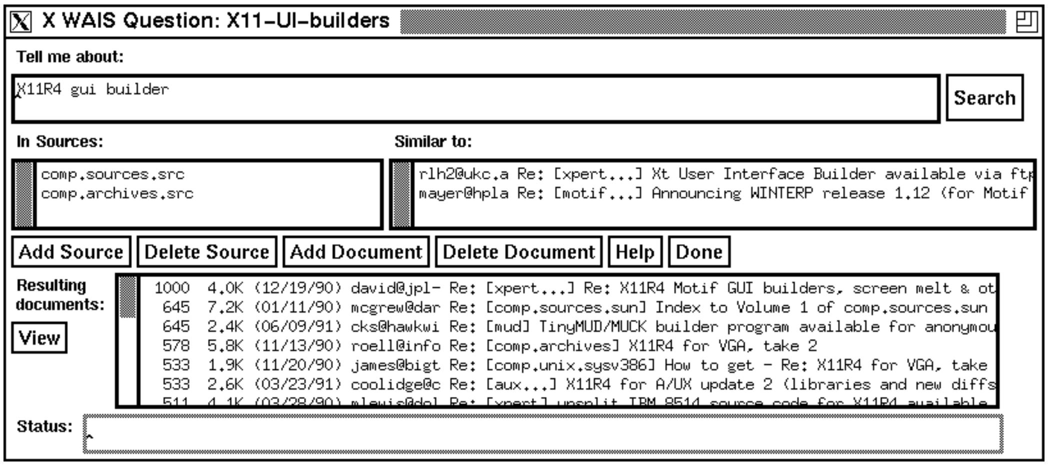

Figure 7: Relevance Feedback

"Power" WAIS users have been known to keep two WAIS windows active. One for searching the directory and adding new database sources and one for actually asking the data questions.

Once you have added a database description to your personal list of databases the `directory-of-servers` search can be skipped. This description can only be considered to be a temporary cache since there is no automatic mechanism to update your description when the database supplier makes a change. Currently databases descriptions don't change too much.

4.5 Saving a Search

Many WAIS clients provide a mechanism to "save" a search. This packages up the current query with all its database sources and any relevant feedback documents so that it can be "run" periodically as databases change. For example, you might be interested in programming environments for X-windows applications. Every week you might perform a search on a group of Usenet news archives to see if anything new has been mentioned. This has some obvious advantages if you have ever tried to follow a few active newsgroups!

5 Setting up a Server

If the WAIS technology is going to make electronic publishers out of all of us, the procedures to set up a public WAIS server must be simple and straightforward. It isn't quite that yet, but it also isn't extremely difficult. The key players in this setup are `waisindex` the indexing routine and `waisserver` the network server routine. Both of these come with the standard WAIS software package for Unix.

The steps involved in setting up a public access server with examples from my setup of a local database are as follows:

1. Find the data. We chose the Index to Journalism Periodicals (IJP) as a test.

2. Massage the data into a format currently understood by `waisindex` or (often much easier) make the modifications (in C code) to `waisindex` to handle the existing format of the data. Since IJP already existed as a database, it was simple enough to write a report that wrote it in "paragraph" mode for `waisindex`. Paragraph mode treats every blank-line separated paragraph as a document. `Waisindex` considers the first line of the paragraph as the title. Since the IJP didn't have proper dates, these were inserted to the beginning of the first line of each paragraph.

   `waisindex` currently supports over 25 different document formats with more being added frequently.

3. Set up a directory with lots of space to store the indexes and perhaps the source data. You cannot move or change the source data once the index is built (without re-building) since the index contains explicit references to the path and character positions in the file.

4. Run `waisindex` on the data, directing the indexes into a disk area that has lots of free space. (Note `waisindex` produces a full text inverted index so that running it will at least double the disk space required.) Since I was working on a machine with 128MB of memory I set the memory parameter to 40MB and the indexing went very quickly.

   There is an append capability in `waisindex` for adding new records to an index without re-building the whole thing. On early releases of the software this tends to expand the database very quickly and it is recommended that the index be rebuilt from scratch periodically.

5. Add in a `z3950` entry to `/etc/services` on TCP port 210.

6. Add a line like the following to `/etc/inetd` to cause `waisserver` to be called when a connection comes in on the `z3950` port:

```
z3950   stream  tcp     nowait  nobody  \
   /usr/ccs/bin/waisserver waisserver.d \
   -d /usr/Local/lib/wais-data/public   \
   -e /usr/spool/syslog/wais/wais-public
```

   Make sure that absolute paths are used to specify file locations. It is best to run the server under an innocuous user-id (like `nobody`).

7. You should set up facilities to monitor and scroll the log files. I wrote a simple shell script to summarize the very wordy log files for IJP. Other more sophisticated ones are available on the network.

8. Edit the *database*`.src` file to include a good, keyword-rich description of the database. To register it for public access, send a copy of *database*`.src` to **wais-directory-of-servers@quake.think.com**.

6 WAIS at UWO

At Western we have been gradually increasing the awareness of WAIS as a network information retrieval tool. We have also started to promote it as a mechanism for electronic publication of local information. We have been treading fairly carefully in this area since the software can be a little on the unstable side.

While there is a wealth of information out on the net that could be useful to faculty, staff and students at UWO, this section concentrates on the sorts of services that we have been able to provide locally via WAIS.

6.1 Index to Journalism Periodicals

The Index to Journalism Periodicals is a bibliographic index of about forty journals about journalism. This information has been maintained by the UWO Graduate School of Journalism (GSOJ) for the past ten years, contains over 15,000 entries and is published primarily as microfiche. The fiche are sold by subscription to clients all over North America.

In a flat file the data occupies about 1MB. Each entry is about 5 lines long and gives typical bibliographic details along with some subject headings. The fiche version of the Index is accessed solely through these subject headings.

The WAIS version has been installed on a central campus unix machine as an experiment in providing this information on-line and to find out if anyone would be willing to pay for such access. It regularly receives queries from as far away as Australia and France (there are some French language articles indexed). In all, with no charging in effect, we are seeing about 350 queries per month from about 250 different machines to this data. In May of this year the new WAIS access was advertised to the existing fiche subscribers.

A recent project has been to make the data easily available to the students in the GSOJ from their network of PCs. This access is expected to lower the demand for help in using the paper and fiche versions from students.

GSOJ is now looking at making some of their other databases WAIS-searchable for their students. An index to The London Free Press (the local daily) and a research papers database are under consideration. This would allow students to search a topic in a number of databases with one operation instead of sequentially as they now must with the more manual paper and fiche based facilities.

The School is also considering a faster cycle time on the updates to their databases — Moving from 6 months to 1 month for the IJP, for example.

While this project is still in its infancy and the jury is still out, it shows encouraging signs of success. It remains to be seen if people will actually *pay* for WAIS access.

6.2 Change Notices

For the past few years we have been gradually introducing and extending the idea of producing formal *Change Notices* for modifications done to systems at CCS. This has been implemented as a local Usenet newsgroup to which staff who modify any of the CCS systems post a notice that describes the change (what, when, why and how). The intent is to improve communication between team members as they work on various projects and to keep the Operations staff aware of changes to the systems as an aid to tracing problems. Problems, as we all know, follow changes (without fail)!

News isn't very good for archiving messages. We started keeping the

Change Notices for a few weeks in the news system but also stored them in `mh` accessible archive directories, split by months. It was still awkward. A WAIS database proved to be an excellent way to handle this archival information.

One of the primary uses of the WAIS Changes database has been to help us to solve problems that have resulted (in possibly a seemingly unconnected area!) after a change has been made. Another use is to remind ourselves, perhaps months after the original occurrence how a problem was solved. The following example illustrates how WAIS was useful in that later case.

A member of the Workstation Support Team had moved the unix mail disk area to a new part of the disk. After making the switch she noticed that the ucb mail program on the Sun workstations was taking a very long time to start up— it was being locked out. She remembered that something like this had happened before. She started by bringing up the New Question window on her X-display using the command `xwaisq`. She selected `uwo.ccs.changes.src` as the database to search and then added a few words into the search box: `mail lock ucb`. She pressed the search button and was quickly rewarded with a list of Change Notices ranked from 1000 down. In this case, the title line of the top Notice seemed familiar. She double clicked on that entry and a window displaying the text of the change notice appeared on her screen. The change had been written by another member of the Team a few months previously. It exactly described the current symptoms and the fix. The "sticky" bit was set on the new mail directory and the problem was quickly solved.

Having a searchable archive of information has begun to change the way we write our change notices. Rather than posting a terse note that just describes or marks a change, we now encourage writers to explicitly document the steps performed to implement the change. This means that the change notice database can serve as a very quick (and fairly informal) manual for how to solve or fix problems.

6.3 Newsletters

UWO publishes two computing newsletters and imports the Merit *LinkLetter* and the *CA*Net Newsletter* into a Usenet news group. Archival access to articles is enhanced by making a WAIS index of this data. People always vaguely remember an article that they read some where. Searching based on the full text will usually turn it up.

6.4 Frequently Asked Questions

A large number of Frequently Asked Questions (FAQ) files on a wide-range of topics, mainly computing related, have been gathered into the news.answers newsgroup. Some of these are currently available as WAIS databases. We hope to index some more of them and also to develop, maintain and index our own local FAQ. We hope to make this into a valuable tool for our Help Desk maintainers.

6.5 Personal Uses

I index all of my e-mail weekly. Indexing your e-mail makes it easy to find a message that you sent out or received 6 months ago. It provides a filing system that is informal and therefore works for people for whom maintaining a rigid filing system remains an impossibility.

6.6 Other possibilities

- Senate and Board of Governors minutes.
- Campus newspapers: student and administration.
- Restaurant and movie reviews.
- Course outlines and calendars.
- Departmental minutes (using a private server).
- Usage write-ups Unix man pages and local manuals.
- Library catalogues.

7 Problems with WAIS

The problems with WAIS tend to be deficiencies in the current implementations rather than flaws in the architecture. Given enough interest, many of the implementation problems will be solved in future versions of the software.

7.1 Implementation Deficiencies

- While there is a concept of charging built into the protocol there is virtually no use being made of it. If you have a publicly accessible database, how do you make sure that you can charge on a per use basis. That is, how do you get someone's credit card number?

- Most current implementations return a maximum of 40 documents from a search. In some applications this isn't enough. The current implementation limits the size of the returned headline information to one packet (size negotiated, but fixed). Since the server is stateless there is no concept of getting the next 40 documents. Extending the limit to say 100 wouldn't be a problem but changing the statefulness of the server would be a major change.

- New data formats require C programming and a re-compile of the indexing program. A simpler, interpretive language could be defined to handle most cases, but I haven't heard of any work going on in this area.

- Many people find that the searching model is very imprecise and not adequate. When you give the current software a question it doesn't do any fancy natural language interpretation on the question. WAIS just searches using the words in the query and hopes that common words, variants, phrase locality and variants on words will not be too important. In-house, commercial servers at Thinking Machines do handle some of these problems. Newer versions of the public domain software also promise to fix some of these deficiencies. Librarians and people who are familiar with boolean-based systems find WAIS really restrictive. Boolean capabilities are slated for the next release of WAIS.

  While there is still much work to be done in this area, it is well underway. For example, the most recent release of the indexer produces word proximity information that will be used by future searching routines.

7.2 Architectural Problems

A major architectural problem with WAIS is how to keep track of where databases are being maintained. While the number of databases is small, it is reasonable to have a central (and very reliable) site that archives this information and allows it to be searched. This is currently being handled well by Thinking Machines. As the numbers grow, various other sites will offer a cloned service. At some point the managing of all of these directories is going to become very difficult. We may then see the emergence of a third level: a directory of directory-servers. Each level makes searching that much more difficult and time-consuming.

Brewster Kahle envisions[5] servers that will rate databases on the quality of their information and other complex meta-services. Pretty soon in that world, getting at the information starts to become almost as difficult as the current Internet labyrinth.

As X.500 databases become more common, they might serve as the "proper" place to store information about WAIS services. The pointers to the databases are fairly static and structured and so they fit smoothly into the database model supported by X.500. A lot of work has gone into the recent X.500 standard to solve replication and referencing problems. This is work that could be used by WAIS rather than re-invented. The great volumes of unstructured data held *in* a typical WAIS database will probably never be coerced into an X.500 database. The marriage of these two systems could have major advantages for network users. Instead of trying to make one system do everything, the appropriate tool can do the part of the job for which they are best suited.

8 The Future

The future is always a nebulous thing and it is notoriously hard to predict. Nevertheless, I see a bright future for WAIS as one of a growing number of networking applications that will help to make computer-stored information resources a little more accessible and a little more useable. Whether WAIS itself survives is a harder question. The current implementations are far from perfect but there are dedicated and talented people working hard to improve them. The core idea is essentially correct and I expect that for the next few years great thing will come from the WAIS project.

The following is a summary of the some of the directions that I believe WAIS development will proceed.

- Support for more flexible and powerful searching:
  - stripping suffixes like "ing", "ed" and "s" for English databases

[5]In *An Information System for Corporate Users: Wide Area Information Servers*, ibid

  - boolean searching
  - more powerful, cross-server relevance feedback
  - proximity search capabilities, phrasing
  - better understanding of natural language queries
  - non-word searching

- improved index density for space savings
- improved client implementations (especially MS-DOS)
- specialized imbedded client applications perhaps as part of a larger information presentation environment
- Commercial databases, charging and access control. There is the potential for money to be made here so this might attract commercial interests.
- Merging a WAIS front-end with an with existing (commercial) high-speed index and retrieval systems like Open Text's PAT/Lector system as the back-end.
- Multi-media support: sound, video, graphics. One researcher has proposed a music database where the search is formed by humming or tapping out a few bars! So you would expect "da da – da DUM" to place Beethoven's *Fifth* fairly high— Could be a very useful tool to help win prizes in "Goodies for Oldies" contests.

9 For Further reading

WAIS is still young and the project is dynamic. It just recently graduated from an `alt.` newsgroup to a mainline one! Much of the documentation is still incomplete or non-existent. Here's a few pointers to bits that I have come across.

- A WAIS Bibliography from **think.com** in `/wais/bibliography.txt`
- Kahle's papers from **think.com** in the files `/wais/*.txt`.
- Various text files (`*.txt`) in the `./doc` directory of the Unix distribution. The distribution is available as `/wais/wais-8-b4.tar.Z` from **think.com**.

- Many useful items in the WAIS discussion archives. Try searching `wais-discussion-archive` and `wais-talk-archive` using WAIS and keywords like `wais relevance feedback` and `z3950`.
- The mailing list wais-discussion@think.com has weekly postings on progress and new releases. To subscribe, send an e-mail message to **wais-discussion-request@think.com**.
- A mailing list for developers can be subscribed to by sending a message to **wais-talk-request@think.com**.
- The Usenet news group, **comp.infosystems.wais** has recently been created. A Frequently Asked Question document is currently under construction by the members of this group.