Chapter 7

A Universal Search Protocol

[Note: This chapter has been written as a self-contained paper]

7.1. Introduction

This note is an attempt to describe a search protocol that can be used by one program or computer process to talk to another process which does Boolean searches on a bibliographic data-base. The searching process will usually interface to a commercial IR service rather than having its own data base and the 'universal' expresses the hope that the protocol should be translatable into the search language on any available IR host. This is probably rather optimistic but the idea is to make it as universal as possible.

The two conversing processes will be called the **User Process** or UP and the **Retrieval Process** or **RP.** These correspond to the user and the IR service in the usual human-machine search situation. The communication medium is two streams of 8 bit bytes, one in each direction. For UNIX processes this would be implemented using pipes. It is also assumed that in emergencies the UP can interrupt the RP and force it to attend to the next command. This interrupt will result in the aborting of any search activity in progress and the flushing of the interprocess channels.

7.2. Description of the USP

Messages from the UP to the RP will be called **commands** and messages the other way are called **replies.** Every command should get a reply, if only a confirmation, and no reply should be sent except in answer to a command. One field of each command is an eight bit command-number which is incremented each time a command is sent, and should be returned as the reply-number of the reply. This (possibly superfluous) procedure makes it possible to detect any mismatch between command and reply. If the RP finds it impossible to execute a command it should send an error-reply instead of the reply expected.

7.2.1. Search commands and replies.

There are three sorts of search command but only one reply (except error). If the search is successful the reply contains the number of matching documents found as well as a search number, that can be used in future searches and in print commands. The three search commands are: the **string search** which searches all the documents which match a given string, the **boolean search** which matches the docs which match a boolean combination of previously done searches and the **document-number search** which searches a combination of document identifiers (see 'the print command' below).

7.2.2. The Print command

The **print** command tells the RP to send to the UP the text of a retrieved document. The arguments passed would be a search number as returned to a search command and a number between 1 and the number of matching docs for that search, which tells the RP which document is wanted.

The reply to the print command is a bit of a problem. It should contain a document identifier which is unique to that document and some standard paragraphs should be delimited. In particular it should be possible for the UP to detect the abstract, title and authors.

An alternative would be for the RP to put the whole document on a file, and simply to reply with a **print done** command containing the document identifier. This would allow for the possibility of the UP displaying some parts of the document to the user, and then fetching other parts on request from the user. On the other hand, it would make the protocol somewhat less universal, as it would require a file accessible to both processes as well as the pipe.

It might prove useful to have another print command which retrieves more than one document.

7.2.3. Other commands

It would probably be necessary to have a **talk** command which tells the RP to talk directly to the user. The reply would not be sent until direct comunication has finished.

Another command that might be needed to recover from hang-ups is **reset**. This would be accompanied by an interrupt (in UNIX a kill). When the RP receives the interrupt it will stop what it is doing and clear data from its input pipe until it finds the reset command. Then it will send a reset received reply and wait for further commands. The UP will clear data from its input pipe until it finds the reset received reply.

Other commands that might be needed are a **login** and **logout** but calling up the host will usualy be done automatically when the RP is loaded. Something else that might be useful is an **are you ready** command.

7.3. Message Format

All commands and replies use the following format

byte	use	notes
0	message type	
1	message number	(i)
2,3	message length	some message types (ii)
2- or		
4-	other arguments	

(i) This field is used to enable the matching of commands and replies. A reply to a command should use the same message number as the command.

(ii) This field is only used in variable length message types. It is a 16 bit unsigned integer giving the total message length in bytes. It might be possible to do without this field if it is possible to find the end of every message without it.

7.4. Some actual message formats

Below are a few examples of suggested message formats. The list does not pretend to be complete.

7.4.1. Search Commands and Replies

String-Search

type: command

purpose: search the data base for the documents which contain a given string.

reply: Search-Done

format:

byte	contents	notes
0	01	command
1	no.	message number
2,3	length	
4-	string	ascii string

Boolean-Search

type: command

purpose: search the data-base for the documents which match a boolean combination of previously done searches.

reply: Search-Done

format:

byte	contents	notes
0	02	
1	no.	
2,3	length	
4-	search statement	terse, host-independent format for boolean combinations of search set numbers, e.g. (1|2)&3

Document-Search

type: command

purpose: search the data-base on a list of document identifiers

reply: Search-Done

format:

byte	contents	notes
0	03	
1	no.	
2,3	length	
4-	document identifiers	list of document identifiers occupying 4 bytes each.

Search-Done

type: reply

purpose: reply to search commands

format:

byte	contents	notes
0	01	
1	no.	
2,3	search set no.	for use in further searches.
4-7	the number of docs retrieved	

Print

type: command

purpose: get complete document from search set specified

reply: Print-Done

format:

byte	contents	notes
0	04	
1	no.	
2,3	search set no.	
4-7	number of required doc. within this set	between 1 and the number of documents retrieved

Print-Done

type: reply

purpose: reply to Print command

format:

byte	contents	notes
0	02	
1	no.	
2-5	document identifier	document held in file