Chapter 2.

REFINEMENTS AND MODIFICATIONS TO CIRT

2.1. Initial work

In the interval between the development of Cirt and the start of the present project, a new version of the "York Box" Unix-X25 software was installed at City. The first task for the present project was to make the modifications to Cirt which were necessitated by the change. In particular, Cirt makes heavy use of the York Box function library, and the new version of the library had some substantial differences from the old.

One of the characteristics of these differences was an increase in size. We managed to get a version of Cirt working, but it was with some difficulty that we got it within the limitations on program size imposed by our LSI 11 hardware. This left us very little (in fact not enough) scope for the user-oriented modifications to Cirt that we considered necessary (see 2.3). In order to overcome these problems, we eventually decided to attempt a more radical modification as described below.

2.2. Two-process system

In the original version, Cirt consisted of a single program ("process" in Unix terms). A suggestion was made in the original report that it could be split into two processes, the first communicating with the user and the second with the host, clearly with communication between the two processes also. The original reason for this suggestion was to facilitate the development of versions of Cirt to talk to different hosts and/or with different user interfaces.

The process size problems identified above led us to undertake the development of a two-process version of Cirt, prior to introducing the user-oriented modifications. The limitations on process size imposed by the hardware are discussed further in section 5.1 and a detailed

technical discussion of the design of the two-process version is given in Macaskill (Appendix A1). A brief non-technical discussion follows.

Calling Cirt starts one process (the "parent") which in turn starts the second ("child"). Thereafter, the parent handles communication with the user and the child handles the host (except in "talk-through" mode, see below). The parent also operates the search algorithm for translating weighted searches to Boolean; the child undertakes all analysis and interpretation of incoming responses from the host. Communication from parent to child involves control codes, instructions, and search statements in a stylised internal communication format (the child reformats them for the host). Communication from child to parent involves status codes, search results and retrieved references.

When in talk-through mode (chiefly for the Boolean searches in the experiment) the parent goes temporarily to sleep, and the child passes messages verbatim (transparently) between the user and the host.

2.3. Refining useability

From an informal survey of practicing intermediaries it was possible to identify some refinements necessary for Cirt to operate with a degree of facility. The system is recognised as a prototype, so extensive alterations were neither practicable nor feasible. It was understood that improvements beyond a rudimentary state would be the subject of future projects. The following basic changes were therefore instituted.

2.3.1. Search tree

The search tree relating to the search algorithm would not be visible to the searcher. Nevertheless while the machine was processing the request it was necessary to provide some indication that processing was in progress, so the comment "searching" followed by a succession of dots at intervals of a few seconds was displayed.

2.3.2. Limits

A limits facility was designed, for two reasons. Firstly because direct manipulation of search sets by the intermediary on Cirt is incompatible with the search algorithm, and to do so would result in terminating the search. Consequently it is not possible to restrict searches by employing the Boolean "not" interactively. Secondly Medline provides a range of "check tags" such as; human/ animal, female/ male, which are of value in restricting searches and therefore deemed useful

to include in a limits capability. A very basic "limits" facility has been incorporated into Cirt, a series of limiting requests being offered to the searcher at the start of the search (eg year, language, human/ animal, female/ male, and others which can be specified). In effect, these limits serve to define a new collection, a subset of the original, within which weighting, ranking and relevance feedback take place.

2.3.3. Deleting

The original version of Cirt provided a delete command which was only permitted on terms added before the search was executed, or since the last completed search. Some attempt was made to provide a deleting function after the search was complete. This was accomplished by giving the term to be deleted a zero weight, thereby rendering it ineffective to the search. This procedure did not work precisely as planned, in that the searching algorithm continues to distinguish between documents with or without the term, even if they have the same matching value. Further work is required on this problem.

2.3.4. Adding terms offline

A facility was provided (for weighted searches only) to key-in a list of search terms before logging into Data-Star. This then permits an abbreviated "add" command. The intermediary can add either a series of set numbers corresponding to the previously keyed-in terms or simply type "add all" and send all the terms downline to Data-Star

2.3.5. Saving searches

This was made possible for two situations, either a temporary save when changing databases during one search session, or a permanent save retaining terms for a subsequent search session. A command was also provided to purge and update permanently saved searches.

2.3.6. Look mode

The most significant refinement was to divide Cirt into two modes (not dissimilar to Data-Star's separate print and search modes). Look mode allows display of titles from the ranked list of search sets. The searcher is offered four options for each set: ignore, print, look, or quit. After looking at an individual title, the user is (as in the original version of Cirt) asked for a relevance judgement; but if the automatically displayed title does not provide sufficient information to make a judgement, it is possible to ask for further fields (eg abstract,

descriptors, year, language, author and source etc) to be displayed.

2.3.7. Printing offline

Cirt automatically merges into a single set all sets for which a "print" request has been made, together with all titles judged relevant online. This merged set forms the basis for the offline evaluation of full-format prints (see section 3.3.2).

2.4. Discussion: Cirt and the relevance feedback model

Cirt as originally developed was a fairly raw implementation of the probabilistic relevance feedback model; little compromise was made to the realities of searching or the habits of searchers. The modifications discussed above, while making Cirt more useable, also moved it somewhat from the original concept. Some of the changes can be seen as fitting within the relevance feedback framework, some not so well. Some of the modifications made are discussed from this perspective.

2.4.1. Limits

The addition of a limiting capability (ie a broad Boolean search giving a large base set, within which the weighting, ranking and relevance feedback take place) might be seen as conflicting with the principles of Weighted searching, as some documents are categorically excluded, and cannot figure thereafter on any rank list. On the other hand one might plausibly argue that choice of base set is strictly equivalent to choice of database. In any case, provided that the base set is large enough, it would still make sense to use weighting and ranking within it.

The implementation of the limit facilities is consistent with this idea, in that once a limiting set has been established, all subsequent operations take this base set to be the entire database. This does imply, however, that it would be much more difficult to establish a consistent method for subsequent limiting (i.e. during the search).

2.4.2. Ignore

The ignore facility, which allows a complete set, as retrieved at high rank by the algorithm, to be skipped in favour of a lower-ranked set, is clearly antipathetic to the model. Indeed, it was introduced in response to a perceived problem with the model, namely the problem with synonyms. If the searcher introduces a number of synonymous terms, the

weighting method (based on an assumption of term independence) can rank highly a document containing several synonyms, but missing some other vital concepts. More generally, the ignore facility provides a mechanism by which the searcher can modify the ranking produced by the system. This is seen as a necessary compromise with useability.

2.4.3. Print off-line

In its simplest manifestations the relevance feedback procedure is seen as one in which the user goes through a ranked list, item by item, assessing relevance; each assessment might be used to alter the rankings of the remaining items. In such a system, all selections by the user would be done on-line; the user would see all the items selected in the course of the search, and when he or she reaches a stopping-point, no further items are retrieved. The introduction of a printoff facility with which the searcher could request off-line prints of a set (ie not look at each item) but then go on to look at a lower-ranked set item-by-item, was seen initially as a purely pragmatic change. There were two strong pragmatic reasons:

(a) To allow for the fact that some searches would retrieve more items than could reasonably be viewed on-line;

(b) To ensure, for experimental purposes, that the weighted search resulted in a well-defined retrieved set, which could be assessed subsequently for evaluation purposes.

However, there is no strong reason in the relevance feedback theory why such a procedure should not be followed. Indeed there is a theoretical advantage, in that it would probably help in the estimation of term weights that the relevance information is gathering from a wider range of documents, not just those at the top of the list.

2.4.4. Delete and save

One outstanding problem in the model is how exactly to use information provided by the user, or from other sources, to assess the value of each term, and how to reconcile this information with that from relevance feedback (10). The introduction of a delete facility, which allows the user to overide any relevance feedback information, can be seen as a rather simple instance of this. A more sophisticated interpretation of the probablistic model would probably include a Bayesian element as a more general method of combining information from different sources.

A second user-oriented modification to Cirt also includes a design element deriving from these considerations. When a search query is saved, any relevance information to hand about the value of the search terms in the database just searched is saved too. This information then contributes to the weight calculation in any new database. In the context of the probabilistic model, the validity of such an operation is debatable (10). It was nevertheless decided to include it.