

Project 5 - Improve searching functionality

5.1 Background

Since Archie's launch in 2004, there have been considerable increases in the quantity and complexity of data stored in the system, but the search component has remained structurally unchanged. There have been several minor updates to introduce additional search options, but these have all been within the limits of the original framework.

There are a number of known shortcomings to the current search function. One simple example is that it is not possible to directly search for people who live in a developing country – this would have to be done by specifying each country.

Requests for changes to the features available for Archie are recorded on the Archie wish list¹, and assessed by the EMAG. This project proposal is based on a combination of the recorded requests, and developer analysis of the unexplored potential of the search functionality of Archie.

5.2 Proposal and discussion

5.2.1 Focus area 1: Working with search queries

The primary focus in this area is the Advanced Search function. As the data complexity continues to grow, the need for flexibility in the search framework increases. Investing developer resources in a focused overhaul of the search component will pre-empt having to use even more resources over time in response to individual requests.

Use both AND and OR in same search

Many complex search queries rely on using different Boolean operators within a single query. However, this is not currently possible in Archie. Instead, users have to run multiple searches, and use the selection set to combine the results. This makes the system difficult to use, increases the risk of errors and prevents the saving of such searches for easy reuse.

Remove restraints on query construction

The current interface for Advanced Search in Archie is highly structured with users constructing searches by selecting from predefined lists of terms, constrictors and values. This rigid structure is helpful when users construct queries that fit neatly within the structure, but the converse is that it limits the number of possible queries that can be expressed. Imposing this limitation allowed us to save resources during the initial development of the system but it would now be possible to develop a system that incorporates the best of both worlds by adding a search option where users enter their queries in a free-form text based search language. Users would not have to remember all the available attributes, constrictors and operators, as these would all be listed for easy insertion as part of the interface.

The following example illustrates how a complex search that is not possible in the current framework could be expressed in the proposed search language:

"Reviews where Author has Country where (Income is Low Income or Lower-Middle Income)"

Search in current selection set

The ability to search in the current selection set, allows a user to refine quickly their previous search (for example, if the result set is larger than expected).

Search on relation to selection set resources

The current search options are nearly all limited to attributes on the type of resource being searched for. For example, one can find the *reviews* first published in a specific issue, but one cannot *contact persons* for the reviews first

¹ The Archie wish list can be seen at <http://archie.cochrane.org/fogbugz/archiewishlist.jsp>

published in a specific issue. If it was possible to find and select the reviews first, and then run a second search on the selected reviews for the contact persons, many more advanced queries would be possible.

Add additional search terms

Several feature requests relate to making it possible to search on additional attributes. For example, being able to find documents based on the version description, or linked topics. Additional person searches that have been requested include:

- Role with Specification in Entity
- Authors on published reviews only
- Date a role was assigned
- Country income level rating
- The *Sex* and *Country of Origin* fields
- Reference centre based on *Country of Origin*
- Bulk Mailings setting
- Authors by Review Type

5.2.2 Focus area 2: Presenting and working with results

Improvements within this area would benefit users of all search types.

Provide search results as both counts, groups and individual hits

If search results could be grouped or counted based on different criteria, this would allow a user to group the search results by, for example, reviews that were at the title, protocol and full review stages.

Sort results

Users should be able to specify how their search results are sorted for display. At the moment, the results are divided into pages that are only retrieved one at a time (in order to conserve bandwidth, by avoiding to have to fetch thousands of results), and so it would be best if order for sorting was specified as part of the search query.

Ranking of results

Archie should, where applicable, enable ranking of results based on the number of search criteria met. Ranking is especially relevant in free text searches, such as searches within document text. If we implement sorting (see above), ranking would be one of the sort options.

5.2.3 Dependencies on other projects

This project relates to projects 1.b (updating the database server software) and 4 (replacement for the parent database). Most of the improvements proposed here could be implemented if those two projects do not go ahead, but there is significant synergy to be gained by implementing all three.

5.3 Summary of recommendations

Develop Archie's search functionality to ensure it meets the needs of current users and is better placed to respond to future needs.

5.4 Resource implications

The total research and development time for this project is estimated to be about 13 FTE weeks. We estimate that 4 FTE weeks are needed for testing and 8 FTE weeks for user documentation (see note below).

Note: The FTE required for documentation needs to be higher than that required for development time because a highly versatile search interface can be daunting to use, and user documentation is especially important. In addition to developing comprehensive coverage of the search functions in the standard user documentation (the help file and various user guides), more help information could be added directly in interface. This would include advice to users of the not-so-obvious consequences of relying on a particular search attribute.

5.5 Impact statement

The value of stored information is directly related to the ability to retrieve it. A cornerstone for any information management system is therefore its ability to deliver the required information with the right presentation. Improving search efficiency contributes to maximising the benefit gained from the overall resources invested in the IMS.

If this project is implemented:

- Archie users in general will save time building queries, and working with the results.
- Users will have direct access to information that is currently 'wasted'.
- Entities, especially CRGs, will have faster, more reliable, means of retrieving the information they need when reporting on their activities to the Collaboration, their funder and their host institution.
- The capabilities of Archie's search system will be more obvious to users.

The developers will save future resources, by not having to 'force' requested functionality into a restrictive framework.