

A Rich OPAC User Interface with AJAX

Jesse Prabawa Gozali
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
jessepra@comp.nus.edu.sg

Min-Yen Kan
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
kanmy@comp.nus.edu.sg

ABSTRACT

Open Public Access Catalogs (OPACs) provide patrons with a user interface (UI) to help their information seeking tasks. Even though many OPAC UIs are now web-based, their architectures are often static, which does not allow them to integrate user assistance modules dynamically. We report on a UI that supports integration of such modules, while providing a usable and rich environment. We explore how Asynchronous JavaScript + XML (AJAX) can be employed to create an OPAC UI that offers a better user experience and task support. Our developed UI features a modular architecture that combines several Natural Language Processing (NLP) modules employed to enhance information seeking. Our UI manages queries in a novel way with a tabbed interface featuring an overview/details presentation model, and an AJAX query results data grid. Preliminary user testing results are also presented.

Categories and Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces

General Terms: Design, Human Factors

Keywords: OPAC, Human-Computer Interaction, GUI, AJAX

1. INTRODUCTION

While web-based OPACs have improved accessibility for remote users, usability was sacrificed as OPAC interaction was restricted to clicking hyperlinks or filling forms. With the advent of AJAX, UI designers can now build richer, more dynamic interactions within OPACs. We have capitalized on this design pattern to redesign our local OPAC to incorporate query analysis and expansion modules, while enhancing system usability.

2. THE USER INTERFACE

2.1 Suggestion Bar

A 1998 survey indicated that a major usability problem with OPACs is in finding appropriate keywords for search [5]. As such, our group at NUS has been researching and developing scalable query suggestion modules to close this gap. To date, we have developed modules for spelling correction, subject heading based query expansion [3] and known item detection [2]. We have incorporated these modules' output as part of a suggestion bar in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

JCDL '07, June 17–22, 2007, Vancouver, British Columbia, Canada.
Copyright 2007 ACM 978-1-59593-644-8/07/0006...\$5.00.

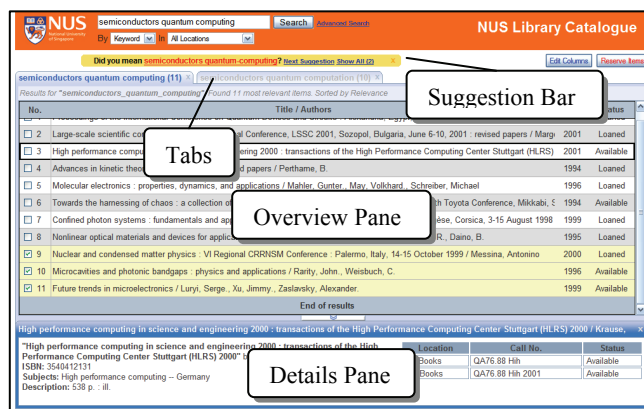


Figure 1. Prototype AJAX-based OPAC User Interface

our prototype OPAC. We believe that users may not distinguish between these different suggestions, hence our UI displays the output of these modules dynamically in a single uniform suggestion bar. Clicking “Next Suggestion” cycles through the suggestions one by one, clicking “Show All” displays the full list. Clicking a suggestion opens a new tab (see Figure 2), where the revised query is shown with its hit count, i.e. query previews are integrated with the suggestion bar.

2.2 Tabbed Interface

To our knowledge, no other OPAC has integrated a tabbed interface. Tabs have often been used to group and organize workflow in a variety of systems, e.g. web-browsers, office applications. Our tabbed interface provides users with a clear overview of all submitted queries in the current session. As the information seeking process may involve several stages of query expansion and rephrasing, tabs provide an effective way to organize these queries transparently in the UI without the clutter of multiple web-browser windows. Tabs also allow users to quickly switch to past queries without having to rely on web-browser history mechanisms. Tabs also allow the UI to be more responsive by caching or pre-fetching the query results. Other UIs have used breadcrumbs to present query histories, but we feel that tabs are more suited. Breadcrumbs work well for hierarchical structures or when trails are long. In contrast, a typical OPAC session lasts only two turns. Furthermore, tabs suggest that multiple query results can be viewed and switched between easily, which is facilitated by AJAX.

2.3 Overview/Details Presentation Model

Users may need to compare many items in the query results. However, typical web-based OPAC UIs zoom directly to a details display, replacing the search results list. This makes it difficult to

evaluate, as users cannot compare two records directly. However, displaying two or more records at once takes up too much space and presents too much clutter.

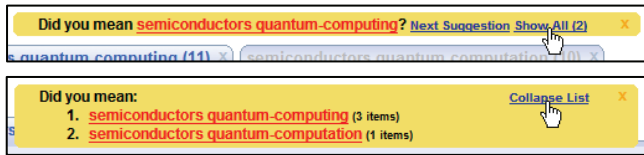


Figure 2. The Suggestion Bar

We believe that users would benefit from keeping the results list visible when viewing item details, as it would help keep track of where they are in the main task [1]. However, other times users may only need an overview of the results, indifferent to the items' details, e.g. in a known-item search. Thus, screen real-estate should be dedicated to query results.

To satisfy this dilemma, our UI uses a hybrid overview/details presentation model. Clicking an item in the results list slides up the details pane from the bottom of the UI to cover the list; hitting "back" on the browser slides the pane down, revealing the results list. Clicking and dragging the partition between the two panes allows the user to arbitrarily reposition it, allowing users to view item details without losing sight of the results list (see Figure 3).

2.4 AJAX Query Results Data Grid

The query search results are organized into a data grid in our UI. This helps users by allowing them to sort by column as well as select columns to be displayed, as recommended in [4]. By default, results are fetched by relevance but may be sorted according to other fields by clicking on the column headings: Title, Call No., Year, and Status. The Title/Author (compound), Year and Status columns are first displayed but may be changed by the "Edit Columns" button. Users assess the query results without pagination to avoid unnecessary navigation between pages (i.e. without a clear overview of results that are already viewed) [6]. Results are asynchronously fetched 25 at a time by clicking on the "more results" button at the end of the table.

3. RESULTS

The preliminary testing results have shown positive feedback on the UI. Five university students with various degrees of OPAC usage were asked to perform two mock tasks on the prototype. The two tasks separately test the Overview/Details model and the Suggestion Bar through non-trivial known-item searches. In both tasks, users were comfortable and familiar with navigating the tabs even within a tabbed web-browser. The dual pane mode of the Overview/Details model was found easy to use and very helpful after a brief one-time demonstration, but was not discovered by users without a prior demonstration. Users also had no problems with the lack of pagination. Users did not sort or edit the columns in either tasks, but when prompted, had no problems in identifying means for achieving those effects.

However, the placement of the Suggestion Bar above the results was problematic. Subjects immediately started scanning from the first line of results, effectively missing the Suggestion Bar. Given this, we are planning to embed the Suggestion Bar directly in the results list. Also, in dual-pane mode, two users tried to use

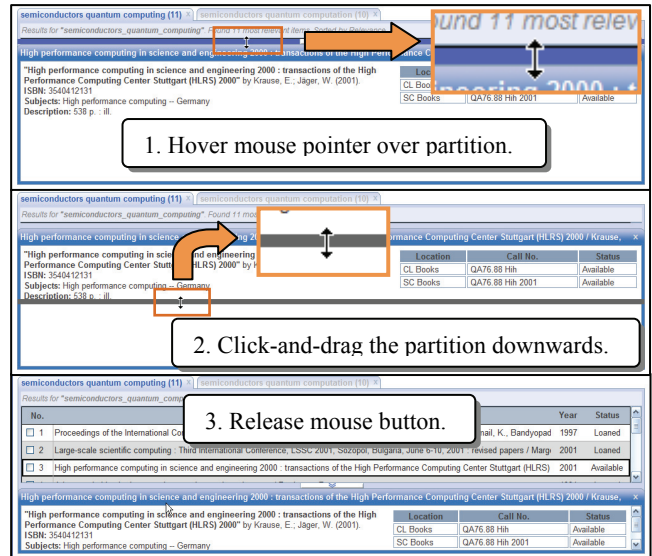


Figure 3. The Overview/Details presentation model

keyboard arrow keys to navigate the list. As such, this feature will be integrated into the next design.

4. CONCLUSION

While the discoverability of the Overview/Details model and the Suggestion Bar is still an issue, users were very positive on their usability and functionality. From the study, we are able to conclude that OPACs have much to gain from an AJAX UI. The preliminary results have shown positive feedback in not only aesthetics but also novelty, functionality, and usability of the UI.

5. ACKNOWLEDGMENTS

We thank the WING group, students of the National University of Singapore and Singapore Management University who have helped with user testing at many stages of system development.

6. REFERENCES

- [1] Ellison, M. Secondary windows in online help: What do users really make of them? *Usability Interface: Newsletter of the STC Usability SIG*. 7, 3, 2001
- [2] Kan, M.Y., & Poo, C.C. Detecting and supporting known item queries in online public access catalogs. *Proc. Joint Conference on Digital Libraries (JCDL 05)*. Pp.91-99. 2005
- [3] Komarjaya, J., Poo, C.C., & Kan, M.Y. Corpus-Based Query Expansion in Online Public Access Catalogs. *Proc. European Conference on Digital Libraries (ECDL 04)*. 2004
- [4] Lergier, R., & Resnick, M. *A framework for evaluating user strategies in internet search and evaluation*. 7th Conference on Human Factors and the Web. 2001
- [5] Rousseau, G., Jamieson, B., Rogers, W., Mead S., & Sit, R. Assessing the usability of on-line library systems. *Behavioral & Information Technology*. 17(5), 1998, 274-281
- [6] Spool, J. As the page scrolls. *UIE eye for design*. Retrieved January 18, 2007, from http://www.uie.com/articles/page_scrolling/. 1998