Access to Archival Finding Aids: Context Matters

Junte Zhang, Khairun Nisa Fachry, and Jaap Kamps

University of Amsterdam

Abstract. We detail the design of a search engine for archival finding aids based on an XML database system. The resulting system shows results—which can vary in granularity from individual archival items to the whole fonds—within the context of the archive. The presentation preserves the archival structure by providing important contextual information, and all individual results can be "clicked", warping the user to the full finding aid with the selected part in focus.

1 Introduction

An archival finding aid describes a particular archive in a long and complexly structured document, which is now commonly available in XML using the Encoded Archival Description (EAD) standard. It is an open question how to provide access to the information in such archival finding aids. Previous research [1,2] found that users are getting lost in the hierarchical structure of (archival) finding aids, and argued for a user interface that could provide contextual navigation. This supports users by providing both the local detail and a global view of the relevant information. More generally, information access to structured text in a realistic use case is identified as an important open problem in [3].

We address these problems by experimenting with archival finding aids collected from the International Institute of Social History, the Archives Hub, and the National Archives of the Netherlands. We have implemented a prototype of the README (Retrieving EADs More Effectively) system, based on MonetDB with the XQuery front-end [4] and the retrieval models of PF/Tijah [5]. The results are directly post-processed for generating our presentation in HTML/CSS.

2 README Approach

We have identified the following key requirements for the README system:

Preserving Provenance. Our system ranks and retrieves the XML elements separately, but we group results from the same archival description. The hit list in Fig. 1 for the query "*onderzoek vereniging armoede bestrijding*" (in English, research association poverty prevention) presents query-sensitive results grouped per finding aid, and the text nodes with keyword highlighting.

B. Christensen-Dalsgaard et al. (Eds.): ECDL 2008, LNCS 5173, pp. 455–457, 2008.

Fig. 1. Results per finding aid showing the document structure with deep linking

Preserving Document Structure. To support the conceptual understanding of the structural dependencies of the finding aids, a global view of the structure of the finding aids is presented that preserves the document structure as a tree-like structure. We traverse the XML tree with the XPath axes in XQuery, and while we traverse this tree, we select the titles, headings, and other structural dependencies belonging to relevant text nodes.

Deep Linking. Using XQuery, we group the top ranking elements by article. As we merge the contextual relevancy, any item that is merged (such as the XML structure) is clickable, and directs the user to the relevant part of the full finding aid. There are several choices that a user can make using this result list, maximally exploiting the found results and their full context.

3 Conclusions

We detailed the design of the README system: results are grouped per finding aid (preserving their provenance) and preserve the document structure (reflecting the original order). The resulting system provides direct access to parts of the archive, while at the same time preserving their local and global context.

Acknowledgments. This research is supported by the Netherlands Organization for Scientific Research (NWO) under grant # 639.072.601.

References

1. Yakel, E.: Encoded archival description: Are finding aids boundary spanners or barriers for users? Journal of Archival Organization 2, 63–77 (2004)
2. Kiesling, K.: Metadata, metadata, everywhere - but where is the hook? OCLC Systems & Services 17, 84–88 (2001)
3. Kamps, J.: Presenting structured text retrieval results. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems (EDS). Springer, Heidelberg (2008)
4. Boncz, P.A., Grust, T., van Keulen, M., Manegold, S., Rittinger, J., Teubner, J.: MonetDB/XQuery: A Fast XQuery Processor Powered by a Relational Engine. In: Proceedings of ACM SIGMOD, pp. 479–490 (2006)
5. Hiemstra, D., Rode, H., van Os, R., Flokstra, J.: PF/Tijah: text search in an XML database system. In: Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR), pp. 12–17 (2006)