Metasearch tools for desktop search

Paul Thomas
CSIRO
Canberra, Australia
paul.thomas@csiro.au

David Hawking
Funnelback Pty Ltd
Canberra, Australia
david.hawking@acm.org

ABSTRACT

Desktop search, as commonly implemented, is not capable of searching many of the web sites and other sources people use day-to-day. A metasearch model, incorporating desktop search tools and others into a "single box", offers a potential solution. We describe this model and introduce PERS, a library for building desktop metasearch software.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*distributed systems*

1. METASEARCH ON THE DESKTOP

Desktop search is, at least in part, motivated by a desire to provide a single point of access to all of an individual's information. This information could come from any or all of a variety of sources (Figure 1).

As normally implemented, desktop search relies on a local index. Some sources, such as local files, contact lists, or calendars, are little trouble in this model, but other sources pose a greater challenge to desktop search: enterprise resources such as proprietary databases, intranet pages, or LDAP directories require adapters or special processing. Outsourced enterprise applications may be accessible only via limited APIs. The public web, including both pages already visited (and in the browser's cache) and pages not yet seen, is of course too large to process at the desktop. And most troublesome are the large number of online services which are private, are not crawlable, or which charge fees for access, and are therefore not indexed by public search engines. These vary from online catalogues, which are not crawlable, to large fee-for-use databases. Even if there were adapters for every collection, the size of the collection or the rules surrounding access would make local indexing impossible in many cases.

Since it is not feasible—or even possible—to index some sources locally, this precludes access to these sources from

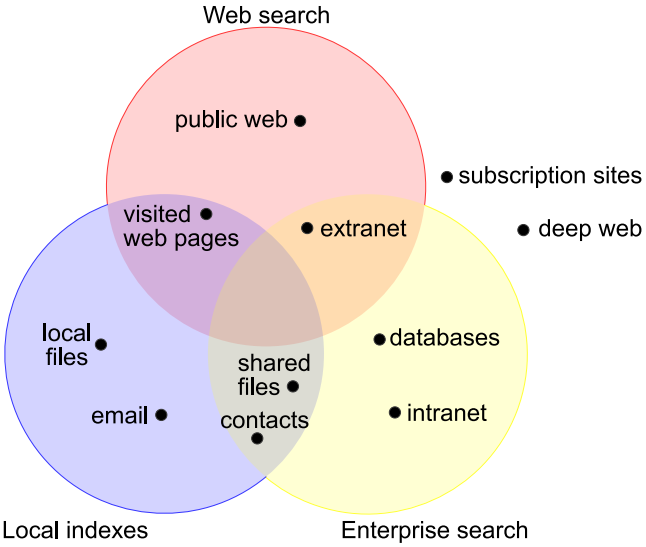

Figure 1: Sources we may wish to search, and approximate coverage of various search technologies. "Deep" web sites and subscription sites are not searchable by any of the three technologies here, although they may present their own search interface.

such local-index tools. Similarly, web search engines are limited to the public web and exclude local or enterprise content; and enterprise search engines exclude local data or the public web. Some desktop search products do offer search of the public web, either through affiliation with a web search engine or in a very limited sense through a search of visited or cached web pages. However, none of the three search technologies cover outsourced enterprise applications, subscription services, web sites which require some sort of credential (e.g. usernames and passwords), or the "deep web".

Without a single search interface to all of a person's information resources, the amount of time taken to find things increases at least linearly with the number of search interfaces used. For example, finding comprehensive background on a significant customer may require separate searches of an employee's desktop, the corporate email archive, externally hosted sales database, subscription financial health reports, and the whole web. There is also a greater chance of missing something useful or important: if a choice of search interfaces is offered, users may simply search in the wrong place.

Metasearch, also known as "distributed" or "federated"

SIGIR'10, Workshop on Desktop Search, July 23, 2010, Geneva, Switzerland.

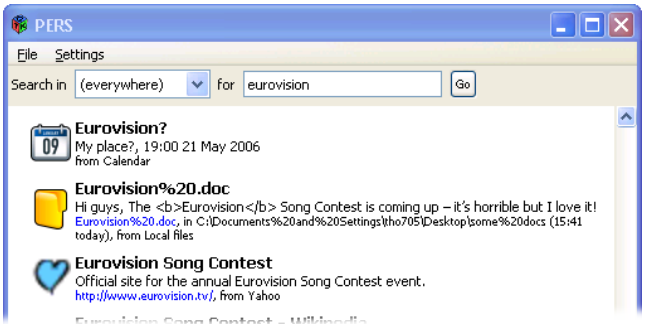

Figure 2: The PERS desktop interface. Shown are partial results from a query covering a calendar, a local file collection, and the public web.

search, offers a way to provide comprehensive, single-search-interface search. Rather than integrating at the data level by building a local index, it integrates the results of multiple search engines, each with their own index.

Desktop metasearch brings its own challenges and opportunities, and it is not yet clear how many lessons from other metasearch applications will carry over. Can a system automatically discover which search engines should be integrated? Can it characterise those engines well enough to enable intelligent selection which subset of available search engines should be consulted for each particular query? Does the query need to be translated prior to transmission to particular engines? Finally, how can the results sets from multiple engines be merged into a single high quality set and presented in a useful way?

In general the metasearch model is unable to rely on any form of cooperation from the underlying search engines, other than returning a result set in response to a query. However, an advantage of the approach is that the metasearcher can rely on the underlying search engines to deal with problems such as indexing; access control; thesauri and query expansion; and ranking.

Uncooperative metasearch has been well studied within artificial test collections [1, 3] but not yet in the desktop context. This is perhaps surprising as desktop search may be seen as a "killer app" for metasearch, but is largely because of difficulties in conducting experiments in this more realistic setting. The PERS metasearch library may overcome this difficulty.

2. PERS, A METASEARCH LIBRARY

PERS (`http://es.csiro.au/pers/`) is a *per*sonal *s*earch library for supporting and evaluating metasearch approaches to desktop search. PERS can handle a wide variety of data types and scales: prototypes have covered collections ranging from small personal calendars, through library catalogues, to the entire public web. It implements a "hybrid" model [2], meaning it can use a combination of existing search services and its own local indexes to search both local files and data on any subscription, corporate, or public service. A variety of merging and presentation options means it can present results from all sources in a single interface (Figure 2).

The PERS library comprises many components for metasearch at the desktop, including:

- A variety of configurable routines for "uncooperative" metasearch. This includes characterising search engines (sampling and estimating size); building language models; selecting sources for a given query; carrying out searches in parallel; and merging results.
- Search for sources including local files, calendars, addressbooks, local and remote email, the public web, particular websites, and anything with a web interface (which includes most subscription services).
- Filters for a number of file formats including HTML, PDF, and Microsoft Office.
- GUI and web front-ends.
- Support for experimentation and a range of utility functions: alternative algorithms can be switched in or out (often with a single line of code), execution can be logged, interface actions can be recorded and reported to a central host, etc.

PERS is implemented in C# and has been run under Windows, Linux, and OS X. Simple uses of the library need only a dozen lines of code.

It is easy to add other routines, other filters, and to harness other search engines. To date PERS has supported experiments in evaluation, interface design, and server selection, in web and desktop search contexts [e.g. 4, 5].

3. CONCLUSION

There is some indication from brochure sites that leading commercial desktop systems may be taking tentative steps down the metasearch path. However, this path is still very much untrodden. Clearly research is needed, and we hope that the PERS library will enable and encourage academic researchers to work on these very interesting problems. Such research can be undertaken without the need for expensive infrastructure or access to massive data held by public search engines. PERS is potentially a valuable way of studying aspects of searcher behaviour, in the context of their everyday search tasks.

4. REFERENCES

[1] J. Callan. Distributed information retrieval. In W. B. Croft, editor, *Advances in information retrieval*, volume 7 of *The information retrieval series*, pages 127–150. Springer, 2000.

[2] N. Craswell, F. Crimmins, D. Hawking, and A. Moffat. Performance and cost tradeoffs in web search. In *Proc. Australasian Document Computing Symposium*, pages 161–170, 2004.

[3] W. Meng, C. Yu, and K.-L. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34(1):48–89, Mar. 2002.

[4] P. Thomas and D. Hawking. Experiences evaluating personal metasearch. In *Proc. IIiX Symposium on Information Interaction in Context*, pages 136–138, 2008.

[5] P. Thomas, K. Noack, and C. Paris. Evaluating interfaces for government metasearch. In *Proc. IIiX Symposium on Information Interaction in Context*, 2010. To appear.