Associative Sources and Agents for Zero-input Publishing

David Wolber and Christopher H. Brooks
Computer Science Department
University of San Francisco
2130 Fulton St, San Francisco, CA 94117-1080
{wolber, cbrooks}@usfca.edu

Abstract
This paper presents an associative agent that allows seamless navigation from one's own personal space to third-party associative sources, as well as the personal spaces of other users. The agent provides users with access to a dynamically growing list of information sources, all of which follow a common *associative sources API* that we have defined. The agent also allows users act as sources themselves and take part in peer-to-peer knowledge sharing.

Categories and Subject Descriptors
H.4.3. [**Information Systems**]: Communication Applications

General Terms
Algorithms, Design, Human Factors, Standardization,

Keywords
Web services, Polymorphism, Aggregation, Reconnaissance, Agents, Context, Associativity

Introduction
The context for our research includes two emerging phenomena: 1) an explosion in the availability of general purpose and domain-specific document collections and 2) the pervasiveness of incredibly powerful computing, storage, and networking capabilities available to ordinary computer users. The purpose of our research is to leverage these phenomena in order to improve the research and creative process.

To this end, we have developed an associative agent that performs information retrieval tasks while the user browses, edits, and manages files. We have also developed an associative sources API that allows our agent (and others) to access a dynamically growing list of information sources.

WebTop: An Associative Client
Associative agents serve as virtual library assistants, peeking over the user's shoulder as the user writes or browses, determining what associative information would be helpful, and then scurrying off to virtual libraries to gather data. The goal of such an agent is to augment the user's associative thinking capabilities and thereby improve the creation and discovery process. We are currently developing such an agent, known as WebTop.

WebTop employs a zero-input interface, which helps to integrate the separate tasks of creation and information retrieval. The agent underlying the interface analyzes the user's working documents, builds a model of the document's content using methods such as TFIDF, and then automatically formulates queries to third-party associative sources. The results of such queries are listed on the periphery of the user's focus. The user periodically glances at the suggested links and interrupts the working task only when something is of interest. Because zero-input interfaces are continually formulating associative queries, impromptu information discovery is facilitated. Users do not need to stop their current task and switch contexts and applications in order to search for related work.

WebTop also takes advantage of hierarchical relationships between documents. Search engines typically provide results in a linear fashion; a user can select a link to view the corresponding page, but there is no way to expand the link to view documents related to it, and there is no mechanism for viewing a set of documents and their relationships. Most search engines and file managers also typically focus on one type of association. For instance, Google's standard search retrieves content-related links. It is difficult to formulate queries that integrate content-related and link-related associations. Similarly, file managers focus on one association type—parent-child relationships of folders and documents—and ignore hyper-link associations and content-related associations.

WebTop integrates various association types, including folder-file, inward and outward link, and content-relation, and displays them in a tree-oriented view. Associations from each type can be listed at each level of the tree, allowing a user to view various multiple-degree associations, such as the documents that point to the content-related links of a document, or the inward links of the outward links of a document (its siblings).

Note that when the source of a link is the personal space of another user, this allows the client user to navigate into the personal space of that user. When a document from another user's personal web is expanded, the system will display outward, inward, and content-related links from that same source. Outward and inward links from a personal web include both folders and documents, allowing the client user to navigate both the folder hierarchy and the links within the personal space of the other user.

One of WebTop's primary design criteria is the breakdown of the boundary between external and internal documents. In the traditional desktop, there are tools that work with web documents (search engines) and tools that work with local documents (file managers and editors), typically with little integration between the two. WebTop integrates local and external documents into a single context tree view and explicitly considers links from local to external documents. For instance, if a local document contains a hyperlink to a web document, the agent will display that relationship. If an external document has similar content to that of a local document, that association will be displayed.

Such integration of previously separated tools is beginning to occur in commercial systems. For example, one can now blog using the Google toolbar. In that case, searching and blogging (annotated, published bookmarking) are integrated, but saving to the user's personal space is not. WebTop integrates all of these features—when a user links a document into the personal web, it is saved locally and, if within the shared personal web space,

WWW 2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

made available to other users. We call this feature *zero-input publishing*—just by bookmarking and saving documents, the user can disseminate knowledge.

Associative Sources

In many domains, web service providers are agreeing on standard programmatic interfaces so that information-consuming applications need not re-implement client code to access each particular service. For instance, Microsoft has published a WSDL interface to which securities information providers can conform.

Our system provides standardization in a *cross-domain* fashion by considering web services that provide similar "associative" functionality but are not generally within the same topic domain. In particular, we consider a class of web services which we call associative information sources. These services associate documents based on document characteristics, such as content, keyword, or author. The Google and Amazon web services are examples of services in this class, as are domain-specific information sources such as FindLaw, the Modern Languages Association database, CiteSeer and the ACM Digital Library.

Currently, such services either provide only a web page interface that must be scraped by an agent or a web service based on their own API.. Because of this non-uniformity, a client application must talk to each associative source using a different protocol. Code written to access Google's Web Service API cannot be reused for Amazon..

More importantly, the lack of a uniform API prohibits the use of polymorphic lists of associative sources. Without polymorphism, the choice of which sources a client application should use must be set at development time. An end-user cannot access a newly created or discovered source without changing the client code.

A standardized API and registry system is clearly the solution. Initiatives for standardization of protocols exist in both the web meta-search areas, with START, SDLIB, and SDART, and in the digital library world, including the OAI, OCI, and XLink. We have defined an API using Web Services with SOAP. Because the WebTop API contains various associative methods, not just keyword search, we have chosen to develop our own protocol.

The API contains methods that for keyword and citation search that allow the client to specify and restrict the number and type of links and elements that should be considered. Results are returned in a generic list of Metadata objects, where the Metadata class is defined to contain the Dublin Core fields and a URL.

With this system, any organization or individual can expose a digital collection as an associative information source. If there is already a web service for the collection, the owners or a third party can write a wrapper (adapter) service that conforms to the associative sources API but makes calls to the existing service.

After the source is implemented and deployed, it can be registered using a provided web page interface. Clients can then access this registry to dynamically update their list of associative sources.

To bootstrap the system, we developed a number of associative source web services, including ones that access data from Google, Amazon, Citeseer, the Meerkat RSS feed site, and FindLaw. We have also developed sample C# and Java web service code that can be downloaded and used to build new associative sources

Peer-to-Peer Knowledge Sharing

A key component of our project is the idea of a personal web. A personal web consists of the collection of documents and bookmarks on the user's local hard disk or server space. On initial startup, WebTop users specify the root folders to be analyzed for the personal web. WebTop identifies hyperlinks between documents and the characteristic words of each document, and builds an inverted index for full-text search of the space. As a user works, this personal web metadata is updated so that it is always consistent with the file system. This can help a user to discover unrealized relationships between new and previously existing documents.

A more interesting use of the personal web is for peer-to-peer knowledge sharing. It should be noted that the personal web is not just a collection of documents that can be searched. Instead, it consists of relations between documents and associations, including hyperlinks and folder-document relationships. Thus, each time the user categorizes a document by placing it in a folder or adds a hyperlink within a document, he is adding to the richness of the information.

To expose personal webs for sharing, we are implementing a web service that conforms to the associative source API and returns information from a personal web. On initial start-up of WebTop, users are asked if they want to expose their personal spaces as information sources, and, if so, specifications as to which folders should be shared with whom. If a user chooses exposure, the system will automatically register the personal web as an associative source and, each time the user logs on to the system, deploy the web service exposing the methods to the outside world.

Once the user specifies the shared folders, he or she will be able to update shared information automatically—all document saving, bookmarking, and link creation will update the personal web, thereby creating additional knowledge to share. We hypothesize that zero-input publication will lead to more sharing, since much of the information a user creates or annotates is hidden from others not for privacy reasons, but because publishing the information as a web page or a blog takes effort.

We realize that privacy is a complicated issue in both the corporate and academic settings. The challenge will be to provide a privacy specification mechanism that is flexible enough to provide for the various needs of individuals and organizations, but easy enough that people actually use it instead of choosing "share all" or "share none". We plan to take advantage of existing efforts in this area, such as the W3C *P3P* effort. Our plan is to implement a fairly simple mechanism, make the system available, and then use user feedback to iteratively refine the privacy mechanism.

Summary

We are building a system that will allows users to more easily manage and share the knowledge they create and discover. Based on the ideas of the zero-input interface and zero-input publishing, WebTop allows users to discover and share relationships between documents with minimal effort.

References

[1] Wolber, D., Kepe M., Ranitovic, R., Exposing Document Context in the Personal Web,. *Proceedings of the International Conference on Intelligent User Interfaces* (IUI 2002), San Francisco, CA.