15. WAIF: Web of Asynchronous Information Filters*

Dag Johansen[1], Robbert van Renesse[2], and Fred B. Schneider[2]

[1] Dept. of Computer Science, University of Tromsø, Tromsø, Norway
dag@cs.uit.no
[2] Dept. of Computer Science, Cornell University, Ithaca, NY 14853
{rvr,fbs}@cs.cornell.edu

Summary. The Internet is seeing a rapid increase in on-line newspapers and advertising for new products and sales. Yet only primitive mechanisms are available to help users discover and obtain that subset of these news items likely to be of interest. Current search engines are really only first step. For locating news providers, word-of-mouth and mass mailings are still used; for retrieval of news items, users are forced to poll web sites regularly or provide e-mail addresses for follow-up mailings.

WAIF is a new framework to facilitate easy user access for Internet users to relevant news items. WAIF supports new kinds of browsers, personalized filters, recommendation systems, and – most importantly – an evolution path intended to enable efficient deployment of new techniques that enhance the user retrieval experience.

15.1 Challenges

Today's World Wide Web has begun to offer convenient mechanisms for locating and retrieving information. But search engines – like Google and AllTheWeb – and other of current technology only work well for information that is relatively static and remains relevant for long intervals. More and more, we see on-line services providing highly dynamic kinds of information. This information has value for only a short period of time and thus might be stale by the time it has been recorded in a search engine's index. We call such information *news* and are driven to provide high precision access; our goal is an easy way of getting news items to exactly those people who have an interest in that news.

In today's WWW *publishers* have few mechanisms to identify the set of *consumers* for news item dissemination. A publisher either must wait for a subscription or generate an ad hoc mailing list. With subscriptions, news items reach only a small set of the interested parties; email blitzes ("spam") reach many people outside the target audience.

Consumers also have few mechanisms to specify what information they are interested in. The consumer must find, often by chance, and subscribe to publishers whose output has high overlap with the consumer's interests. Interesting information from other publishers is not seen, and unsolicited and irrelevant news items are received from publishers employing ad hoc mailing lists.

* This research was funded in part by the Norwegian Research Council (IKT-2010 Program), in part by DARPA/AFRL-IFGA grant F30602-99-1-0532, and in part by the AFRL/Cornell Information Assurance Institute.

A. Schiper et al. (Eds.): Future Directions in DC 2002, LNCS 2584, pp. 81–86, 2003.

15.2 Future Research

WAIF (Web of Asynchronous Information Filters) is a new project that attempts to address these current inadequacies of WWW by supporting real-time news location, routing, filtering, and analysis. In short, WAIF provides a framework to enable news publishers to reach interested consumers. The architecture offers a standard protocol for users to subscribe to news item streams and for publishers to publish news items. A small set of WAIF mechanisms facilitates the construction of collaborative filtering and recommendation systems. So, subscribers are able to rank publishers and to re-publish news items that interest certain other communities. We are mindful that the success of WAIF thus depends heavily on having a user-friendly browser, so this is a central research concern for us.

15.2.1 An Information Overlay Network

WAIF is essentially an Overlay Network [15.1] where the endpoints are publishers and consumers; the WAIF transport infrastructure contains mechanisms to rank news items for each consumer individually as well as for routing messages to consumers according to this ranking. In WAIF, consumers explicitly subscribe to publishers, and consumers have a convenient way to rate the news stream provided by each publisher. The paradigm can be likened to a sound mixer control panel, with a slide control for each subscription; we are considering the mixer panel as part of our WAIF browser. This is a prototype browser similar in respect to the Curious Browser [15.3], which infer user interests based on a combination of explicit and implicit ratings.

A consumer in WAIF can be a producer as well. If a consumer C receives a news item that C thinks is of interest to other consumers, then C can re-publish this news item. Similarly, if C happens across an interesting web page or receives some interesting e-mail, then C may publish this information as a news item. (We discuss below how re-publication is specified using a simple drag-and-drop interface.) Note how WAIF blurs the distinction between publishers and consumers. Both are called WAIF *principals*.

Each WAIF principal can publish news items to one (or more) *topics*. The "topic hierarchy" can be created as the principal sees fit. For example, the New York Times might create topics like "news/politics/international" and "money/stock". An individual might create "personal/family", "personal/bowling", and "work" topics. For the WAIF browser, we display a tree to represents this topic hierarchy. An individual publishes web pages or re-publishes news items simply by dragging them onto the correct node of this topic hierarchy. This is much like dragging e-mail messages into a folder hierarchy.

If a WAIF consumer does not think the news item is worthy of re-publishing, the consumer can either delete a news item after reading it, or drop the news items into a "garbage can." The latter indicates annoyance with the news item. Such actions enable WAIF algorithms to improve how subsequent news items are ranked. (Other methods to get ranking input include keeping track of reading time, mouse movement, etc.)

A URL is associated with each WAIF principal and with each topic to which the principal posts news items. Examples might include

"waif://nytimes.com/news/politics" or
"waif://aol.com/personal/john/family".

Consumers subscribe to such URLs; a subscription generates a record for the subscriber, including the *mailbox* at which news items will be delivered. The mailbox is similar to a standard SMTP mailbox, and similar (if not the same) protocols may be used to ensure reliable delivery of news items.

Third parties can deploy information *filters* and *fusers*. These news item processors are WAIF principals; they subscribe to WAIF URLs while also publishing information based on their input. We intend to use our TOS system [15.7] so that users can upload new filters into the WAIF infrastructure in a safe manner. Filters might even migrate between TOS servers in order to optimize scalability or other notions of performance.

Not all filters add value by enhancing precision. Some filters might be deployed to improve scalability of WAIF or to support anonymous subscription. Other filters might maintain state and attempt information synthesis. For example, a filter might analyze the news items published by several stock exchanges and publish forecasts.

15.2.2 Personalized Filtering

News-on-demand systems that automatically process news and provide personal presentations are currently being constructed [15.8]. We are developing a personalized filtering system that allows individuals to do content-based filtering. We call such a system of filters a "PONS" (Personal Overlay Network System)[1]. Each user will be able to deploy his or her own PONS using a set of filters, possibly obtained from the web. The PONS infrastructure will automatically place the filters on appropriate TOS servers to minimize network resources, while maximizing sharing between users.

We will use a PONS prototype under construction to illustrate this concept. The goal is that a novice Internet user shall be able to configure and transform the Internet into a highly personalized, asynchronous and autonomous distributed filtering network with high precision and recall. Creating this PONS works as follows.

Initially, the user specifies interest in certain predefined UDDI conformant topics through a *WAIF-browser*. The user does not have to know anything about programming and how to deploy filters, location of remote servers and the like; all that is needed is personalized preferences specified through scroll down menus. The net effect of this dialogue is a file with what we call a *user profile*, a list of topics, some general, some very detailed. This user profile is submitted to a remote *WAIF Deployment server*.

A WAIF Deployment server acts as an advanced match-making server. It keeps track of remote data sources and tries to match a user profile with these. Locating resources is a key problem, and both pull- and push-based techniques are investigated. This includes use of pull-based centralized search solutions [15.9] and peer-to-peer techniques [15.6], to more push-based schemes where data sources update the WAIF Deployment server with new directory information.

Upon successful location of a convenient data source, filters have to be composed for deployment. It is a key requirement that a regular user should not be involved in this specialized task. Therefore, the WAIF Deployment server transforms a user profile into one or a set of filters. We have identified a set of software patterns in this type of computations and have devised a collection of reusable pattern-based *nano-filters*.

[1] A *pons* is also a relay station between the brain and the spinal cord.

Hence, a nano-filter (code) is coupled with specific user data and is deployed close to the data source. This deployment is at one or several *WAIF Filter servers*, which act as advanced mediators [15.10].

A WAIF Filter server either produces data itself or subscribes to data streams from other data sources. This can be traditional topic- or content-based Internet data sources. The nano-filters can now parse the data streams, and specific alerts triggered lead to notifications being sent to the user. This might create a precision problem closely related to how users experience e-mail spam today, and we approach this problem by sticking one or a stack of *spam filters* into the upstream data feed in the PONS. The idea is that the nano-filters do the course grain data filtering, while spam filters do more and more fine grained filtering. Typically, the spam filter is stateful, while the nano-filters are, besides eventual parameters, stateless. Finally, data passed through the spam filters, ends up at a *relay filter*, a highly context sensitive distribution filter much like a MS .Net Passport Alert service.

We have now described how a user can create and deploy his own data fusion and filtering PONS. Other PONS services are also being constructed. For instance, a user profile can be submitted to a *WAIF Profiler*. This is a server that takes a user profile as input and generates an HTML-page of recommendations for that particular profile. In our first naive implementation this is a personal start page for traditional web browsing. This page evolves over time based on user actions recorded by the WAIF browser.

A more elaborate PONS is one using collaborative filtering techniques in a socialware context [15.5] The idea is that captured preferences of multiple users can be used to recommend comprehensive events or items of interest to other users. Multiple PONS from like-minded users are connected together through *WAIF Recommender servers*. A horizontal network of such servers co-operates and exchanges data to predict additional topics or products a user might like.

15.2.3 An Information Market Place

WAIF defines a new web – one in which the links join principals that communicate through subscriptions and thus capture relationships based on how information is being used (rather than how the original author intended it to be used, which is what WWW hypertext links do today). This new web may be crawled and indexed, just like today's WWW. (Each WAIF URL may have a short XML description associated with it to allow for keyword search). And, as with the WWW, information sources may be ranked. Unlike WWW, the links in the WAIF have weights associated with them, hopefully leading to improvements in relevance ranking.

WAIF will go a step beyond passive relevance ranking and notify consumers automatically of news item sources that might interest them. This is similar to what Amazon does when it suggests or recommends other items of interest to a consumer. Such a service can be implemented in WAIF by a filter. We expect many such filters to coexist, just like today there are many search engines to choose from. A consumer can subscribe to one or more of these WAIF filters and relate his or her "profile" (the set of subscriptions of the consumer, along with the corresponding rankings). Based on this information and information obtained by crawling, the filter could then post recommendations as news items which, just like any other news item source, the user can rank using the mixer control panel, and/or re-publish.

We envision that WAIF will form a market place of information. Consumers negotiate contracts with publishers for information. Such a contract specifies not only what a consumer will pay the publisher for information, but also restrictions on what the consumer is allowed to do with said information. Subsequent re-publishing of received information, for example, may be restricted by copyright protection, or may require extra payment. It is unlikely that mechanisms can be implemented that directly enforce the terms of such contracts, but we are interested in extending WAIF with automated auditing and tracking mechanisms that may help in tracking down violations.

Although there is a similarity between WAIF and newsgroups (and, if you will, the publish/subscribe paradigm), what we are proposing is fundamentally different. In newsgroups, publishers post messages to particular groups, forcing the publisher to anticipate which communities will be most interested in the news item. These communities are explicit collections of users (even though the subscribers are anonymous) joined by simple notions of affinity. In WAIF, publishers do not publish to any explicit group of subscribers. In that sense, WAIF is closer to a content-based [15.2, 15.4] rather than to a topic-based publish/subscribe paradigm. Of course, newsgroups may be tied to WAIF, in that its messages may be published in WAIF.

15.3 Conclusion

We are currently refining the WAIF architecture and have started building some of its components. Ranking strategies will be key to the success of WAIF, so at present we are focusing on that question. It clearly will be important to create prototypes and actively use them, in order to drive this research. Other research issues we are tackling include: the scalability of routing and news items, the privacy of consumers, and a way for publishers to charge consumers for news items.

References

15.1 D.G. Andersen, H Balakrishnan, M.F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proc. of the Eighteenth ACM Symp. on Operating Systems Principles*, pages 131–145, Banff, Canada, October 2001.

15.2 G. Banavar, T.D. Chandra, B. Mukherjee, J. Nagarajarao, R.E. Strom, and D.C. Sturman. An efficient multicast protocol for content-based subscription systems. In *Proc. of the International Converence on Distributed Computing (ICDCS'99)*, Austin, TX, June 1999.

15.3 M. Claypool, D. Brown, Le P., and M. Waseda. Inferring user interest. *IEEE Internet Computing*, 5(6):32–39, Nov/Dec 2001.

15.4 A. Carzaniga, D.S. Rosenblum, and A.L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, August 2001.

15.5 F. Hattori, T. Ohguro, M Yokoo, Matsubara, and S. Yoshida. Socialware: Multiagent systems for supporting network communities. *CACM*, 42(3):55–61, March 1999.

15.6 H.D. Johansen and D. Johansen. Improving object search using hints, gossip, and supernodes. In *Proc. of the 21st IEEE Symposium on Reliable Distributed Systems (SRDS'02)*, Osaka, Japan, October 2002.

15.7 K.J. Lauvset, D. Johansen, and K. Marzullo. TOS: Kernel support for distributed systems management. In *Proc. of the Sixteenth ACM Symposium on Applied Computing*, Las Vegas, USA, March 2001.

15.8 M. Maybury. News on Demand. *CACM*, 43(2):33–34, February 2000.

15.9 M. Meng, C. Yu, and K.-L. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34(1):48–89, March 2002.

15.10 G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49, March 1992.