

## **ELF: The Electronic Learning Facilitator**

**R. Kirby, P. McAndrew, A. Kilgour, H. Taylor, J T. Mayes**

**ICBL, Heriot-Watt University, Edinburgh**

**rbk@icbl.hw.ac.uk**  
**<http://www.icbl.hw.ac.uk/>**

### **Abstract**

As the world-wide computer network becomes ubiquitous, new tools have been developed, such as the World Wide Web (WWW), for the delivery of multimedia hypertext-based documents. Similarly, there has been an explosion in the amount of email, bulletin boards, and Usenet News available. This has led to the major problem of information overload. We are slowly but surely being overwhelmed by the amount of information available to us.

To address this problem, the New Technology Initiative (NTI) of the Joint Information Systems Committee (JISC) is funding the ELF project over a period of two years to develop a tool, the Electronic Learning Facilitator. This tool is aimed at assisting learners to make effective use of the information available. The ELF project started in December of 1993, with the goal of producing a tool for installation in UK academic sites.

This paper will first discuss the new networked information sources that are appearing around the globe, and then analyse the requirements of learners for accessing this information. From the requirements, the paper will show the desirable features of an ELF, such as monitoring the user at work to gain hints as to what the user is doing, being guided by the user directly, and utilising information gathered from other ELFs, or from monitoring the various information sources for announcements of new services, updates of information, etc.

The main body of the paper will illustrate the current progress on the project, outlining the implementation of the ELF. ELF is written using various high-level languages such as the Tool Command Language (TCL) and Perl, as well as the more traditional C++. It consists of various agents, each agent having a specific task to perform, such as monitoring a WWW site for any new documents, or monitoring a Usenet Newsgroup for articles that might be of particular interest to the owner of the agent. Finally the paper will report on our plans to test users on a prototype.

### **Introduction**

Recent years has seen an unprecedented increase in the use of the Internet and other global networks of computers. We have seen estimates of between 100,000 and 1,000,000 new users each month joining the Internet. Predominantly these are information seekers, but these figures do also represent a significant increase in the number of information providers. For learners, we now have a significant set of information sources, albeit obscured by a lot of noise.

There are many different types of information sources now available, all with varying attributes. For many years the Internet has provided email. For learners, this has allowed direct contact with the authors of research papers at different (and sometimes the same!) institutions. The Usenet News system has enabled learners to reach a far wider audience, with less intrusion (since it is up to each user to decide to read the News, whereas email is generally always read). Again, this

provides a good resource for the learner to find out more by asking questions, and receiving a wide variety of responses, although often this requires wading through a variety of "metoo" responses, and 'flames' (inflammatory responses which tend to solicit ad hominem attacks). Electronic mailing lists are the email equivalent of a newsgroup although they are harder to find out about, but tend to have less noise. FTP sites are another, now familiar, information source. Here electronic copies of research papers can be stored at a FTP (File Transfer Protocol) site and others can then download (copy) them to their local machine for printing and reading. This is a rich source of information in the sense that there is rarely any noise stored at a FTP site, but it is hard to know where to look and what to look for, except from some other information source (such as a News article).

Recently, new types of information sources, such as the World Wide Web (WWW), Gopher, and the Wide Area Information Servers (WAIS) databases have proliferated. Gopher can be viewed as a distributed menu system, where users select items on a menu which may lead to another menu, or to some actual information (such as a picture or a research paper). The WWW, developed initially at CERN in Geneva, is a distributed hypertext system where links to objects are embedded within other objects (where the objects can be text, pictures, video, and other multimedia). Both Gopher and the WWW support the browsing model; the user has to actively follow links to find information, in contrast to email or News where the user passively receives information. WAIS databases are essentially free-text search engines which allow for keyword searching of collections of documents.

The increase in the desire to use these networks, and the increase in the amount of information being made available over these networks is causing problems for both experienced and novice users. This paper will introduce the problems, and outline a possible solution. The paper will end with an indication of current progress.

## **Problems**

Information sources such as mailing lists and Usenet, tend to overwhelm users with the sheer volume of information (often with a low ratio of signal to noise). Both of these can be thought of as unsolicited information, where the user must invest large amounts of time in reading to find the nuggets of useful information. By contrast, users of the WWW often have a different problem, that of navigation. By this we mean knowing where the desired information is to be found. With WAIS databases, the problem is similar - knowing which database to search. There is a directory-of-servers (a database consisting of short descriptions about a large set of registered databases) to help with this, but as with any centralised solution, it can be tied up with the multitude of requests from users around the world. It is also not complete and uptodate.

The current work is predicated on the belief that there is useful information in all of these information sources, but the trick is to know where to look. For example, one often finds out about new WWW sites from the WWW newsgroups. Another major source of pointers to useful sites is from colleagues. There are also index sites which build a database of pointers to other sites using

some keyword mechanism to act as the key into the database. These include the WAIS directory-of-servers, and there are several others on the WWW (such as the Jumpstation, and the WWW Worm).

## **A Solution - ELF**

ELF is an attempt to address these problems. It is essentially a collection of agents which cooperate together to aid a user in finding information of relevance to the user's interests or tasks, in a timely and efficient manner. One of the main aims of this project is for ELF to work without explicitly requiring direct control by the user. To do this, ELF monitors the user at work to determine (from files recently edited, for example), the areas of current interest. ELF attempts to categorise the current work by a few keywords which are then used to seed a search conducted by various agents. The results of this search are presented to the user at some appropriate point (to avoid unnecessary interruptions in the users' current task). ELF can also be controlled directly by the user. This means that the user controls what information sources are searched, and decides on the search query terms.

There are two major sides to the ELF. The first handles the user, and attempts to gain knowledge about the users' tasks. The second is a backend which consists of agents that "know" how to query and search the various information sources. We will discuss each of these parts of ELF in more detail below.

ELF allows for varying interaction with the user. The minimal level of interaction involves a discrete icon on the screen from which various control panels can be accessed. The only other ELF tool that needs to be seen is called PrISM (Personal Information Space Manger). This is a hierarchical structuring tool which can be used as a simple file/directory browser, but can allow for the grouping together of related tasks. PrISM allows access to the keywords that ELF calculates for a directory or file, and provides for the user to edit these. The user can also use PrISM to determine and control which agents are currently working on the users behalf. While it is indeed true that the hierarchical structuring model leaves a lot to be desired, it is easy to understand and matches naturally to the common computer filesystem.

ELF provides several different mechanisms for notifying search results. In the case where a search finishes and the user is currently logged on, ELF can notify by forcing Mosaic (or certain other WWW browsers) to popup on the screen with a page containing pointers to the results found (see Figure 1), using the fact that Mosaic can access most types of information sources, thus providing a simple click-to-access interface to the results. Another option is to indicate within PrISM the presence of results at the relevant part of the hierarchical structure. A third option is to provide an xbiff type program (xbiff is a small icon that notifies the arrival of new mail by changing colour) to flag the arrival of results. These clearly vary in intrusiveness to the user from the popping up Mosaic to the small discrete icon in the corner of the screen.

If the user is currently logged out when the results arrive, then ELF can either send a mail message containing the pointers to the results, or allow for PrISM to indicate the presence of results the

next time the user logs in.

More experienced users can control ELF directly, controlling search parameters, and requesting immediate searches (cf traditional database searching where the user makes a query and waits for the result). ELF also collects statistics about what keywords seem to work the best, and which information sources produced the best results (as selected by the user). These can then be used either by ELF or the user to tailor the searching to fit in more with the working practices of the user.

The backend of ELF consists of the agents that perform the actual searching of the various information sources. One agent (called the News Agent) is responsible for querying a database of news articles. One of its responsibilities is to find articles that announce new WWW sites. This is done in a simple way by looking for "ANNOUNCE" (or some derivative), followed by a Universal Resource Locator (URL) indicating where the WWW site is.

Another agent interfaces to the Jumpstation search engine on the WWW. It mimics a user request to Jumpstation and reformats the returned result (which is a WWW page of links to the results) before presenting to the user. Similarly agents can be provided for other WWW search engines, such as the WWW Worm, and ALIWEB.

Other agents can include WAIS searchers, Gopher searchers, and mailing list handlers. Agents can also be designed to access interactive services such as BIDS (one of the library catalogue systems). One of the drawbacks of providing agents that use the same interfaces as the user (rather than requiring database implementers to provide a special Application Programming Interface (API)), is that if the interfaces change, then the agents will no longer work. This could be as simple a change as a rearrangement of some menus. We will address this issue somewhat in the next section.

Since different information sources provide different search strategies (such as boolean exact matches, set operations, partial matching), some agents have to do more work than just sending a query on and retrieving the results. For example, an agent that accesses a search engine that only accepts a single keyword, might need to submit several searches for a multiple keyword query, and then combine the results, so that as far as the user or the rest of ELF is concerned it answered the query in one fell swoop.

## **The Departmental ELF**

As outlined above, the ELF, being a computer program, can generate large amounts of queries in a short space of time for long periods without rest (after all this is why we replace people with computers!). It was quickly realised that if everyone used an ELF, then the networks would be flooded, and the search engines accessed by the ELF agents would overload. To address this, we propose a departmental level ELF. This ELF is responsible for actually doing the work. The agents of a user's ELF will submit their queries to the departmental level ELF, which then has the opportunity to multiplex queries across users (in other words, perform duplicate queries only

once). Results can be cached and used to answer later queries (recognising that the results may be out of date). The departmental ELF could also help mimic the 'Word of Mouth' phenomenon by allowing for transferring of information between user ELFs. Another benefit of the departmental ELF is to provide centralised management of agents, which addresses the changing interface problem alluded to in the previous section, by requiring just one place to be changed.

## **Current State and Future Plans**

A prototype for a single ELF is near completion. This ELF includes a database of news articles (dynamically changing due to new articles coming in and old ones being expired, a News agent that operates over this database looking for announcements or new WWW sites, or just general keyword searching. Other agents include a Jumpstation agent and a Archie agent (Archie is an index of most FTP sites in the world). The prototype also includes PrISM, and a user monitoring agent which periodically searches the user's directory space for new or changed files, and attempts to categorise them with a few keywords (using a combination of full text analysis, typed text analysis, and clustering techniques). This ELF runs under the Unix environment running X, and is implemented in a mixture of C++, TCL(Tool Command Language) and Perl.

The development of ELF will involve iterative design, with user testing at each stage. By employing usability metrics we are aiming to test groups of students, lecturers and researchers using the initial prototype. The development of ELF as a tool to support the process of learning in a higher education context is being conducted within an overall framework provided by the concept of the Intensely Supportive Learning Environment (Sykes et al, 1993). This adopts a constructivist approach to learning (Kommers et al, 1992) in which the learner has available a range of tools for the performance of learning tasks. The process of learning through active seeking and structuring knowledge is emphasised at the expense of the traditional view of courseware as packaged and pre-digested exposition. An important issue raised by this is the identification of the point at which the work done by ELF might become counter-productive for learning.

Final aims of the project are to build a departmental ELF, to port the user interface to MS Windows and to release the system to the UK Academic Community.

## **References**

KOMMERS, P., JONASSEN, D. & MAYES, J.T. (Eds) Cognitive Tools for Learning, Springer-Verlag, 1992

SYKES, P., SCHAPER, J., MAYES, J.T., PALMIN, H. & MARCIPONT, V. ISLE: A collaborative project to build an Intensely Supportive Learning Environment. In H.Maurer (Ed.) Educational Multimedia and Hypermedia Annual: Proceedings of ED-MEDIA 93. AACE, Charlottesville, VA, 1993