

# ‘Seeing the wood for the TREES’

*Phil Turner<sup>§</sup>, Alex R. Rogers<sup>‡</sup>, Susan Turner<sup>§</sup> and Jeremy Ellman<sup>‡</sup>*

<sup>§</sup>The Department of Computing, University of Northumbria at Newcastle

Newcastle-upon-Tyne, NE1 8ST, England

Tel +44 191 227 47 50

email {phil, susan}.turner@unn.ac.uk

<http://www.unn.ac.uk>

<sup>‡</sup>The MARI Group, Wansbeck Business Park, Ashington, Northumberland, England.

Tel +44 191 402 0191

email alex.rogers@mari.co.uk, jeremy.ellman@mari.co.uk

<http://www.mari.co.uk>

**Abstract.** *This paper describes the work in progress of the TREE project. TREE is a European-funded language engineering project addressing the issue of advertising and accessing employment opportunities across Europe. Advertisements for jobs will be stored in a database and made available in the language of choice of the end-user via a WWW interface. An account of the early design and evaluation of the user interface to the TREE system is here described.*

## 1. The TREE Project

Free movement of labour across national boundaries is an important aim of the European Union. One of the prerequisites for this is accessibility of information about employment opportunities, both from the point of view of people seeking work, and of their potential employers. However, many EU citizens are denied full access to employment opportunities because information may not be readily available, and even where it is, it may not be available in the relevant language. The TREE project aims to address this problem by providing a system on the Internet where employers can place job adverts, and which users can browse, each in their own language. Access to this service will be either through the user's own Internet provider or at dedicated terminals located in employment centres. Although there are many Internet sites where jobs are advertised, TREE offers significant added-value in that intelligent search and summarisation are provided which are independent of the original language of the job advert.

The TREE system facilitates inter-lingual search by storing advert information in a schematic form which associates a set of semantically equivalent terms or phrases expressed in different languages with a language independent code. In addition, a hierarchical semantic structure (intended to reflect ‘common-sense’ classification) permits users to specify a general area of employment interest and retrieve specific jobs that fall within this area. For example, users might specify ‘engineer’ as a search term and be able to retrieve job adverts with job titles in the various different engineering domains (e.g. ‘chemical engineer’, ‘civil engineer’). These features of the search process and the TREE system in general have posed a particular problem for the user interface design in that search terms specified by users must be able to be associated with a language independent code in order to be valid and useful search terms. Additionally, although the database is intended to provide complete coverage of the range of terms associated with the employment domain this can not be guaranteed. Thus providing users with the freedom to enter free text may prove counter-productive as they may fail to retrieve jobs for what they consider to be perfectly reasonable search terms. To some extent

this problem can be limited by sophisticated terminology look-up methods whereby the root form of a search-term is extracted before look-up, but this does not avoid the case where a term simply does not exist in the terminology database.

Two solutions present themselves: the first is to implement client-side validation of search terms entered as free-text by means of a Java applet. Whilst this may present users with (relatively) immediate feed-back regarding the validity of the terms they have entered it will not avoid the possibility that users become frustrated by the systems inability to accept perfectly reasonable search terms. In addition the amount of data makes for excessive download times. The second alternative is to restrict the user to the set of valid terms by using select-lists, radio-buttons and so forth. This second solution requires the use of Java applets to overcome the limitations of HTML. In practice TREE will provide both solutions. The first, because user assessment of an initial prototype indicated that users preferred to enter search terms via a free-text box. The second because the project is aware of the incompleteness and fallibility of the terminology database.

### **This paper**

This paper confines itself to the issue of system usability and specifically to the user interface design. The design of such an interface poses significant challenges as it must provide access to complex search facilities while remaining usable by a heterogeneous user population.

## **2. The user interface**

User interface (UI) development is a complex, non-linear process. Consequently this document cannot hope to embody a final statement of the final TREE UI. Instead it is a *snapshot* of work in progress.

### **2.1 The purpose of the UI**

The user interface to the TREE system must satisfy the following primary goals:

- to provide access to the TREE database for the employment centre staff (who are expert users);
- to provide access to the TREE database for the WWW community;
- to support search parameters phrased in a variety of languages;

### **2.2 The technical specification of the UI**

Shneiderman (1997) has recently suggested that it will take a decade until sufficient experience, experimentation, and hypothesis testing clarify WWW design issues. However as a starting point he does recommend an approach described by Lynch (Lynch, 1995 - Yale C/AIM Web Style Guide):

*Proper World Wide Web site design is largely a matter of balancing the structure and relationship of menu or "home" pages and individual content pages or other linked graphics and documents. The goal is to build a hierarchy of menus and pages that feels natural and well-structured to the user, and doesn't interfere with their use of the Web site or mislead them.*

So given that there are no clear and agreed WWW-specific guidelines for designing a UI to a WWW-based service, other than to make it feel 'natural' and well-structured to the user, we have adopted conventional UI design heuristics (*c.f.* Nielsen, 1993, Hartson and Hix, 1993) in the development of TREE. The development process itself started with user profiling and

informal task analysis and follows an iterative prototyping and evaluation cycle, similar to that described by Hix and Hartson as the STAR methodology.

### User characteristics

Conventionally users are divided, somewhat arbitrarily, into 'expert', 'occasional' and 'naïve' user groups (see Dillon *et al*, 1997 for a critique), where the expertise / naïveté dichotomy is actually a computer usage/skills continuum. Potential **expert users** may be characterised as:

- **skilled computer users**. That is, they have keyboard and mouse skills. From this it may be safely assumed that TREE may be accessed by way of a standard Internet capable PC requiring no specialist input or output devices.
- **competent**, thus needing less 'hand-holding' than other user groups. This has a number of implications: firstly, they will expect the TREE system to look and feel (behave) like a standard GUI or a typical WWW page. Secondly, it may be assumed that access to the functionality of the TREE system should be as transparent as possible. Thus devices designed to guide the inexperienced user such as 'wizards' or 'assistants' will be unnecessary and undesirable.
- Finally, expert users will expect **flexibility** in accessing the TREE system. Users must be able to formulate database queries in an unconstrained manner.

As to the potentially **naïve users**, this discontinuous user group ranges from the completely computer illiterate to those with a basic appreciation of computer technology. It is unlikely that a user interface to the TREE system which relies on standard input devices and standard user interface widgets can be created to provide 'walk-up-and-use' access for the computer illiterate. Indeed, user trials with an early TREE user interface has confirmed this (see section 2.5). Given these constraints and empirical evidence, completely computer-naïve users are disfranchised from this version of TREE.

Turning to those users with some computer skills, **occasional users** may be characterised thus:

- occasional users are familiar with the operation of standard input and display devices, but may not be at home with the full gamut of GUI widgets
- are unlikely to be familiar with database query operations and Boolean logic, but nonetheless will wish to perform job searches which are just as complex as those required by expert users.

Here a balance must be struck between practical constraints and the need to support these less skilled computer users in formulating database queries. Devices designed to guide the inexperienced user such as 'wizards' will be required, while the more experienced of this group must be able to formulate database queries in an unconstrained manner.

### Frequency of WWW use

Kellogg and Richards (1995) have noted that an important dimension to understanding WWW user characteristics in addition to their knowledge of computers *per se* is the distinction between first-time, intermittent and frequent users of a website. Kellogg and Richards argue that different approaches to the design of a website must be adopted to cater for the these three user groups: first time users require a site overview with explicit command buttons, intermittent users should be supported by orderliness and reversibility, while frequent users need shortcuts, in-depth information and more extended services. The TREE user interface aims to take into account this additional dimension.

## 2.3 Existing user interfaces to employment databases

As a precursor to prototyping the initial interface to TREE, a survey of existing WWW UIs to employment databases was

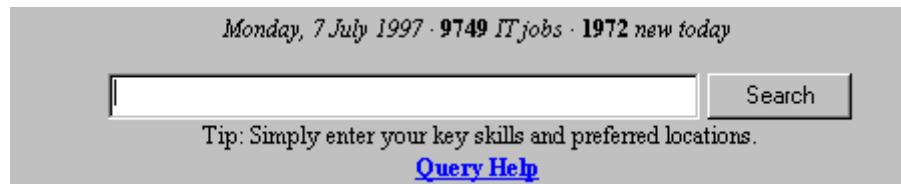


Figure 1: The Jobserve interface

undertaken. Most were found to have a very simple, text entry box interface. Figure 1 illustrates one such interface (taken from JobServe -<http://www.jobserve.com>).

## 2.4 Early prototyping

On the basis of the analysis of the expected end-user population, two distinct UIs were prototyped. The first, designed for expert users, consists of a text entry box. Figure 2 is a screen-shot of the expert user interface following the style of job search interfaces found on the Web.

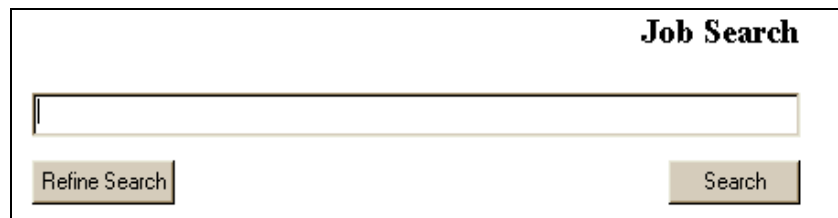


Figure 2: The TREE expert user interface

The <Search> button submits the query to the database, while the <Refine Search> is intended to bring up a more detailed job search dialogue. Figure 3 is a screen-shot from the interface designed to support occasional users. In contrast to the expert user interface this is a combination of *wizard*

and a tabbed dialogue. Such a design was intended to support user navigation in a step-by-step fashion but at the same time allow a complex set of job specification possibilities. A **wizard**

is a style of interaction commonly found in Microsoft™

applications which is designed to guide a user through a complex dialogue. Pages of information and dialogues are presented in an orderly manner allowing the user to

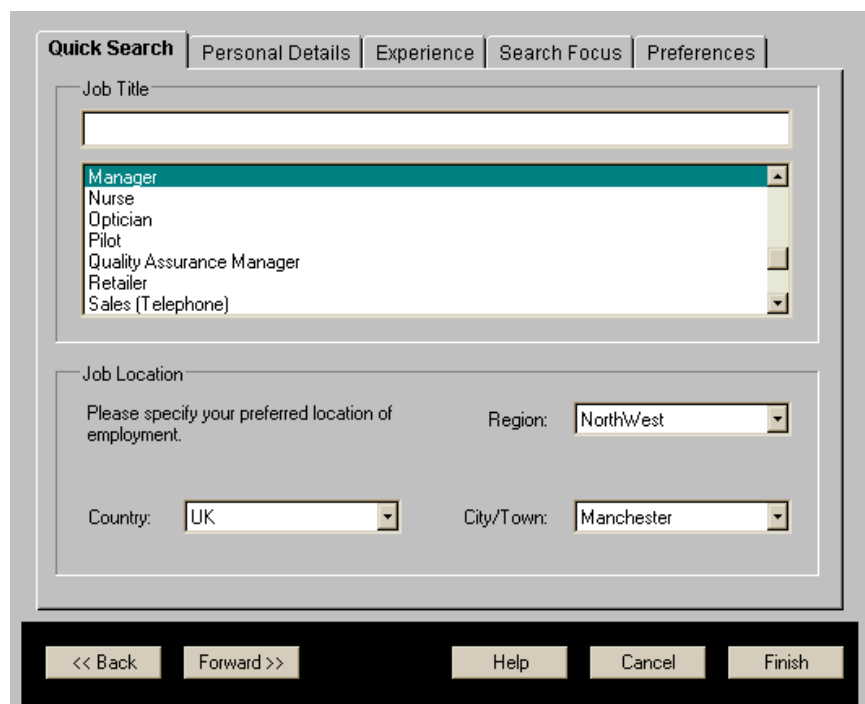


Figure 3: The TREE naive user interface

manoeuvre forwards and backwards through this process.

The **tabbed dialogue** of this UI consists of five pages designed to capture different parts of the job specification. The first page (as above) consists of a job title selector in the style of the Windows 95 help system. Job titles may be entered in the text entry box by:

1. initial letter (e.g. entering the letter *w* would bring up all jobs beginning with a *w* in the scrolling menu below);
2. in full;
3. being picked from the scrolling menu.

The remainder of this page consists of three pull-down menus allowing the user to select country, region, and city/town. At the very bottom of the page are the wizard buttons guiding the user forwards and backwards through the formulation of the database query. The remaining pages of the tabbed dialogue allow the user to specify salary range, work experience and so forth.

The remaining tabs elicited further information. The **Personal Details** tab requested information about where the job applicant was prepared to work and at what rate of pay. The **Experience** tab asked about the applicant's work history, the **Search Focus** tab was designed to deal with requests for generic jobs such as 'manager' by asking the applicant to identify the business sectors in which they were prepared to work. Finally, the **Preferences** tab allowed the user to manipulate the appearance of the naive user interface.

## 2.5 The initial evaluation of the interfaces

These prototype UIs were subject to a series of early evaluation exercises. The first of these is reported here. The initial evaluation of the above interfaces had three aspects:

1. **User profiling.** Details of participants computer usage and WWW experience were obtained.
2. **Using the expert user interface to searching for their own job.** The participants were asked to compose a search for their own profession or job. In doing so they were also asked what they expected the <Search> and <Refine Search> buttons to do. Finally, it was ascertained whether they understood the difference between the (Boolean) use of 'and' and 'or'.
3. **Using the occasional user interface to search for a well specified job.** The participants were asked to enter the details of a job from a short description it. They were also asked about their expectation of the behaviour of the interface.

### The users

Eleven users of differing computer skills, experience and occupations agreed to participate in this exercise. Each participant was asked about their frequency of PC usage, their GUI familiarity, and their WWW browser familiarity.

- Of these only 2 were expert in all three categories; one had never used a computer.
- The remaining 8 people had all used PCs (5 had used Windows / 3 DOS only) but none had used a browser.
- Everyone understood the use of the search and refine search buttons while 4 of the 11 failed to understand the correct usage of and/or in formulating a search.

### The expert user interface

The search parameters entered by the participants when asked to look for a new job for themselves were largely uniform. These included 'personnel', 'receptionist', 'admin',

'computer sales' and so forth, with only two of the 11 participants specifying more than a simple job title. On being asked why they had only entered their job title, the general reply was that they would narrow or expand the search afterwards (note, 'narrow' and 'expand' were used interchangeably).

### **The occasional user interface**

This interface was uniformly disliked. Only 1 of the 11 participants preferred it to the expert user interface but with the caveat that it would have to provide sufficient options. With those users who had MS Windows experience the interface proved to be most disliked and confusing. The mixture of the tabbed interface and the wizard being the source of the confusion. The <Finish> button (standard wizard terminology) was taken to mean 'quit' by 6 of the 11 users. The <Forward> and <Back> buttons taken to be used for within tab navigation rather than between tab and so on. On the basis of these findings this version of the user interface has been discarded.

## **2.6 Revised design**

The revised UI to the TREE system under development has a simple text input box for the search parameters with a greater emphasis on supporting refinement of the output from the search. Two prototypes are (at the time of writing) being developed to:

- refine the search using a *wizard*-like approach, and;
- to order the output using a *wizard*-like approach

The interfaces will then be evaluated and modified in the light of that exercise.

## **2.7 Conclusions**

Attempts to support occasional users through supportive dialogues have been surprisingly unsuccessful in this rather complex search domain and it will be interesting to see whether the unexpected preference of the occasional users for an 'expert' user interface stands the test of time. A key question is whether when faced with a very large number of hits for the job of their choice the occasional users would not rather formulate a more specific query by way of a more 'supportive' interface. Only full user trials will answer this question and these are time-table to begin in November 1997.

## **3. References**

- [Dillon 96] Dillon, A. and Watts, C. (1996) User analysis in CHI - the historic lessons from individual differences research, *International Journal of Human-Computer Interaction*, **45**, 619-637.
- [Hix 93] Hix, D. and Hartson, H.R. *Designing the User Interface*, John Wiley, 1993.
- [Kellogg 95] Kellogg, W.A. and Richards, J.T. (1995) The human factors of information on the internet, In Nielsen, J. (Ed.), *Advances in Human-Computer Interaction: Volume 5*, Ablex Publishing, Norwood, NJ, 1-36.
- [Nielsen 93] Nielsen, J. *Usability Engineering*, Academic Press, 1993.
- [Shneiderman 97] Shneiderman, B. (1997) Designing information-abundant web sites: issues and recommendations. In S. Buckingham Shum and C. McKnight, Eds. "Web Usability" (special issue) *International Journal of Human-Computer Studies*, **46**.