Dynamic Second Language Support for Web-based Information Systems

George R S Weir
Department of Computer Science
University of Strathclyde
Glasgow G1 1XH, UK

and

Giorgos Lepouras
Department of Informatics
University of Athens
Athens 157 71, Greece

ABSTRACT

Non-native speakers of English are faced with the 'second language problem' when required to interact with English-based information systems. This paper describes strategies for addressing the problem, which arises through difficulties in comprehension, in the context of Web-based information systems, by means of a dynamic annotation of English language Web pages.

Keywords: Dynamic annotation, Second language support, User support, Human-Computer Interaction.

INTRODUCTION

Elsewhere, we have characterised the second language problem as a specific issue in user-computer interaction that arises under the influence of natural [1]. Whenever non-native speakers of English have to use English-based software there are inevitable risks that the user's degree of English competence, or influence from their cultural background, will adversely affect their interaction with the computer system. Our on-going research seeks appropriate ways of addressing and reducing the negative impact of poor language comprehension.

In the eyes of many, this problem can be addressed through 'localisation' (cf. [2], [3], [4], [5]). Invariably, this amounts to little more than translation from English to the language of the local community. For example, Figure 1 illustrates menus from a word processor, one of several original English language applications that have been made available in localised versions.

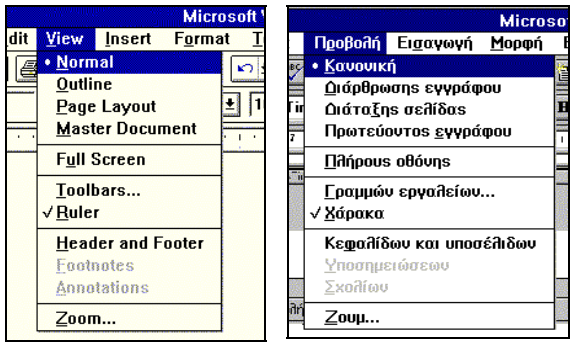

Figure 1: The 'View' menu from Word 6 (English and Greek versions)

Our ongoing research supports the view that such 'localised conversions' offer limited guarantees of successful user interaction. Localisation of existing software applications introduces a further level of potential terminological confusion. Even where common usage is established across an existing range of English language products, there remains scope for variation in the choice of local (translated) terms.

The localisation process is also constrained in its selection of terms by limitations introduced at the user interface. Translated terms may strive to be accurate, but may also be limited in length if required to fit the look and feel of an existing user interface. For this reason, using multiple words to achieve accurate translation of a single non-local term is unusual and constrained by available space (cf. Figure 1).

A recent survey of localised Greek Word processors suggests that the 'translation

approach' to the second language problem introduces new difficulties to the scenario through linguistic corruption and other selective rendering of terms from English to the local language. Our recommended strategy for dealing with the issue is an integrated approach that combines selective translation with other varieties of second language support, including explications, summaries and examples in the user's local language [1].

Elsewhere we have illustrated how this may be accomplished in the context of applications based on MS-Windows, whereby, an application's English on-line help information can be supplemented with native language support (op. cit.). This technique relies upon the possibility of targeted use of second language information within the context of normal application use, i.e. adjacent to the English original.

In addition to predefined local language supplements within Help files, some progress in *dynamic* second language support has been made for Windows-based applications. Thus, a dynamically accessed database of command terminology has been implemented for use in a Windows environment. This allows a range of supplementary Greek terms to be displayed adjacent to their English originals, within the context in which they are met by the user. This is illustrated for the Wordpad application, in Figure 2, below.

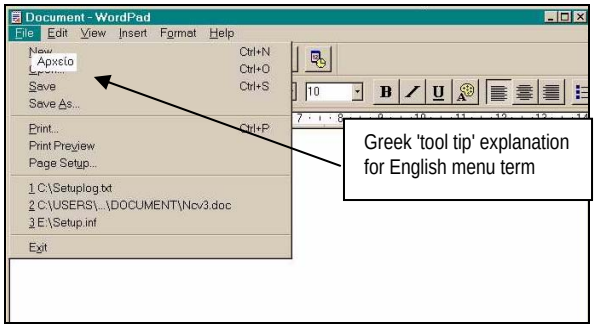

Figure 2 : Dynamic Greek support for menu commands

THE WEB CONTEXT

The present paper addresses a similar range of issues, related to the second language problem, in the context of Web-based information systems.
The World-Wide Web supports easy distribution of information across national boundaries via the Internet. Web-based information systems are often in English or some other local language. Occasionally, alternative language versions are provided to accommodate non-local users and the fact that many Web browsers also support a range of character encoding schemes extends the possibility for alternative language renderings. This is evident in the wide variety of natural languages now found across the Web.

At first sight, Web information systems seem an unlikely locus for the second language problem. Web pages undertake to deliver information, hence language comprehension will be an issue, but, *prima facie*, this raises no significant concerns for user interaction.

Importantly, Web users do more than read delivered information. Even in its simplest browsing role, the Web is an interactive system in which the user must negotiate a route through selected information resources. This means that the user's linguistic abilities (especially in comprehension) will determine their choice of successive resources.

Putting this another way: the user selects links based upon presumption of their relevance and interest value. Where hyperlinks rely upon textual description, the user's interpretation of the text plays a dominant role in determining interaction. We thereby conclude that language comprehension significantly affects the success of user interaction in the Web context. In consequence, user's whose first language differs from that used in the Web pages are more likely to experience degraded interaction. This is precisely the second language problem.

TECHNICAL REQUIREMENTS

Our work on second language support focuses on two languages that differ significantly from English, viz., Greek and Chinese. In both cases, these languages require font sets that complicate their use in conjunction with English language displays. This has technical implications for any system that seeks to combine local language support (e.g., Chinese text) with the original language (i.e., English).

The Greek case is simpler than Chinese since fonts are available that combine Greek with the Roman characters used in English (e.g., Times

New Roman Greek). Thereby, a single font set can display both languages (Greek and English).

In the Chinese case, additional software is required to support display of Chinese ideograms in conjunction with English (e.g., NJWin, available on the Web from www.njstar.com). When the facility to support characters both for English and the local language has been established, we have the technical means of *displaying* local language support with the primary language.

To this must be added a further requirement, which is the ability to create information in the local language. For Greek and Chinese this requires a facility through which the user can *input* Chinese or Greek text. In each case, we employ a native language word processor in conjunction with the required keyboard drivers. Naturally, local language users are essential for creating the native language support texts.

4. Web-based support

Beyond these technical requirements for establishing second language assistance lie significant decisions on how to deploy such support. Obviously, full translation (e.g., to Chinese) may be produced as an accompaniment to English-based Web pages. In our experience, targeted deployment of local language support can enhance user interaction by improving comprehension, with minimal additional risk of language-based dissonance.

To date, we have developed two approaches to Web-based second language support. The first and simpler variety, relies on the use of concurrent frames to supplement the primary English information with local language support. This affords considerable flexibility in the range of local language support that may be implemented. An example is shown in Figure 3.

This implementation of second language support facilitates a range of alternative techniques. Aside from complete translation, the Chinese Options menu can offer a Chinese summary of the main English text, details of key vocabulary with Chinese explanations, and a Chinese list of the link texts that appear in the main English window. Subtler methods for annotating the original English content are also available. For example, potentially difficult technical terms can be targeted with pop-up explanations (Figure 4).

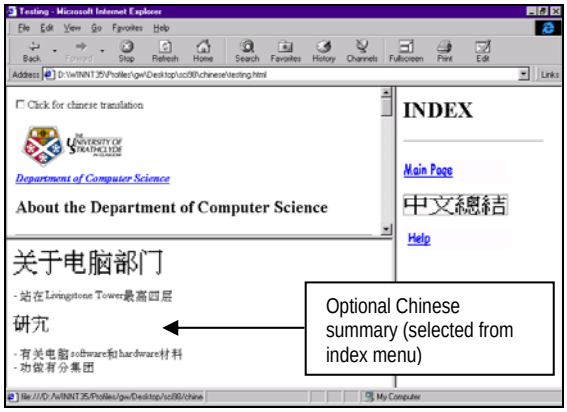

Figure 3 : Frame-based Chinese support

The principal limitation of this strategy is the static nature of the second language support. All supplementary support texts must be prepared off-line and added (as frame-based components to Web documents or as associated pop-up definitions) to the original English information. To enhance the application of this support we are implementing a 'look up' system that dynamically checks word contents of the main English text and provides access to local language definitions and explanations.

DYNAMIC SUPPORT

Our approach to dynamic support is based on an English-Greek on-line dictionary of informatics developed at the University of Athens. A variety of technical means are in development to provide a dynamic link between this facility and Web page content.

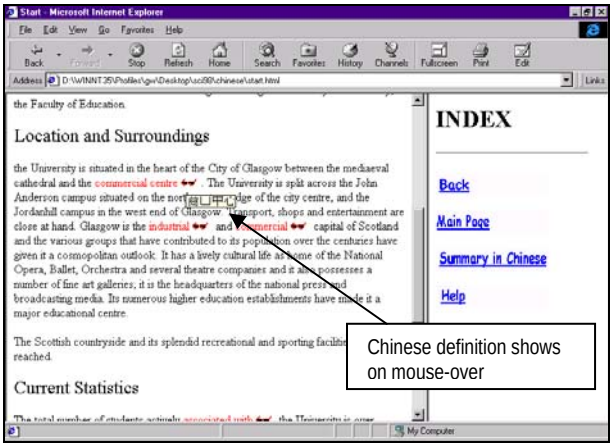

Figure 4 : Pop-up Chinese supplements

An early approach explored the use of a Java program to 'read' terms from the loaded English Web page and seek corresponding Greek explanations from the dictionary. An alternative considered implementation employs a server-side approach with a cgi-script that receives a Uniform Resource Locator from the user, retrieves this document behind the scenes, and performs annotation before sending the result to the user's Web browser. Latterly, we have moved toward the use of a server that acts as an intermediary between the end-user and the requested Web pages.

In this final approach, the user's Web browser is configured to use a local proxy server when accessing external Web pages. The proxy server requests the external Web page on the user's behalf but upon receipt of the requested page does not despatch the received data directly to the user's browser. Instead, the proxy server invokes an external program to scan the Web page contents for occurrences of specified key words or phrases.

Based upon the successful matches, an 'annotated' version of the Web page can be created 'on the fly' and sent to the local user in place of the original Web page data. In addition, the proxy server may be set to cache the annotated Web pages or the original documents. This general process is outlined in Figure 5, below.

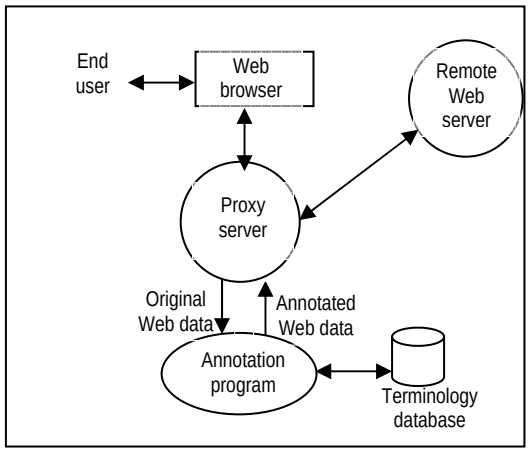

Figure 5 : Architecture for dynamic annotation of Web pages

Customisation

As a refinement of this approach to second language annotation, the proposed system promises additional benefits through customisation for individual users. Customisation will be provided across two dimensions:

1. Level of language and domain comprehension
2. Modes of annotation

In the first case, the annotation system will determine how much and what variety of second language support to add to the source Web pages. This level of support should be determined as a function of the user's language comprehension level and their level of domain comprehension. This will enable a varied response from the annotation system according to the anticipated degree of need in each end user and depends upon weightings being attached to terms and explanations within the annotation database.

In the second case, the user may control the manner in which supplementary information is attached to the annotated Web pages. According to user wishes, annotation may use frame-based supplements in the local language, pop-up second language definitions and explanations, or added Web links to more detailed second language support, including dictionary and thesaurus.

CURRENT STATUS

Presently, we are adapting an existing proxy server for use as a caching annotation server with links to a local Access database facility. This will incorporate an informatics terminology database, supplemented from our studies of the terminology employed across existing applications translated from English to Greek. The database weightings are an experimental issue that will depend upon on-going user trials and will be revisited in later publications.

BIBLIOGRAPHY

[1] G.R.S. Weir, G. Lepouras. and U. Sakellaridis, Second Language Help for Windows Applications, in M.A. Sasse, R.J. Cunningham and R.L. Winder (eds.), People and Computers XI, Proceedings of HCI'96,Springer, London, pp. 129–38. 1996.

[2] E. Uren, H. Robert and T. Perinotti, Software Internationalization and localization. An introduction. Van Nostrand Reinhold, New York, 1993.

[3] J. Karat and C. M. Karat, Perspectives on Design and Internationalization, SIGCHI Bulletin, vol. 28, No. 1 (Jan) 1996, p. 39.

[4] P. Russon and S. Boor, How fluent is your interface? Designing for international users. Proceedings of InterCHI '9, ACM Press, 342-347, 1993.

[5] A. Yeo and R. H. Barbour, Software for the rest of the world, Working Paper 96/2, Computer Science Department, University of Waikato, New Zealand, 1996.

[6] G. Lepouras and G.R.S. Weir, It’s not Greek to me: Terminology and the Second Language Problem, (submitted for publication), 1998.