Tools for Second Language Support

George RS Weir

Dept. of Computer & Information
Sciences
University of Strathclyde
Glasgow, G1 1XH, Scotland, UK
george.weir@cis.strath.ac.uk

Giorgos Lepouras

Dept. of Informatics &
Telecommunications
University of Athens
Athens 151784, Greece
giorgos.lepouras@di.uoa.gr

Abstract

The second language problem is the context in which non-native English speakers are required to interact with English-based computer systems. In other papers, we have characterized this setting and proposed methods of supporting such users. The present paper details several tools that we have developed to assist in our work with second language support. A prime consideration in such tool development is to facilitate easy management of alternative language resources. The need for criteria to direct second language support and the role of such tools in helping to evaluate such criteria is detailed here.

1 Introduction

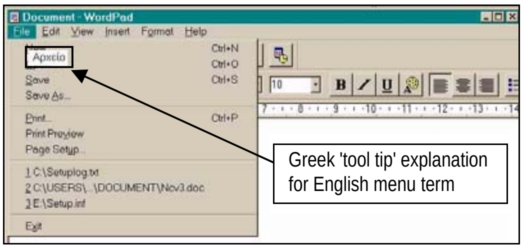
There are extensive populations of computer users for whom English is not a native language. Such users do not interact solely with systems designed for their own mother tongue. More commonly, these users have recourse to some systems designed for a native English speaking audience. Indeed, there is often a preference for such systems over fully localized alternatives (Weir et al, 1996). Indications are that such 'non-native language interaction' has a greater likelihood of problems in usability. In addition to the inherent interactive demands of the computer system, second language comprehension imposes additional overheads and is likely to lead to foibles in interaction that would not affect native English users. Furthermore, attempts to accommodate local languages often create problems that were previously non-existent (Weir & Lepouras, 2001a). We have coined the term 'second language problem' to describe this context and the challenge of supporting non-native English speakers when interacting with English-based computer systems (Lepouras & Weir, 1999).

Based upon our earlier studies, this paper describes several tools that seek to reduce difficulties that arise in the second language context. Two settings are addressed. Firstly, we describe a facility that assists in providing native language support as a supplement to existing interactive features. This complementary support takes the form of 'annotations' and is being prototyped in a Web setting. This facility for dynamically annotating English language information and adding selective native language support is described and we suggest that there is extensive scope for applying such dynamic annotation as a means of interactive support and as a basis for enhanced user comprehension (Weir & Lepouras, 2001b; Lepouras et al, 2002).

Our second toolset addresses a related but distinct situation. A teacher wishes to prepare on-line multiple choice tests for a mixed language audience, based on existing English material. A software tool allows the designer to select features as a basis for multiple choice tests with the added facility that native language (e.g., Arabic or Malay) information can be accommodated. A related tool provides a Web-based look-up facility for checking definitions and word meanings as support for building such on-line second language assistance.

2 Dynamic Annotation Tool

Our dynamic annotation tool is a technique that deploys targeted use of second language information, within the context of normal application use, i.e. adjacent to the English original. This development goes beyond earlier predefined local language supplements within Help files (Weir, Lepouras & Sakelleridis, 1996) to the dynamic generation of second language support for Windows-based applications. A dynamically accessed database of command terminology has been developed for use in a PC environment that allows supplementary Greek terms to be displayed adjacent to their English originals in the context met by the user (Figure 1).



Greek 'tool tip' explanation for English menu term

Figure 1: Dynamic Greek support for menus

Our work touches on a variety of languages that differ significantly from English, including: Greek, Arabic, Chinese and Malay. Three of these require font sets that complicate their use in conjunction with English language displays. The Greek case is simpler than Chinese or Arabic, since fonts are available that combine Greek with the Roman characters used in English. Such technical requirements for establishing native language assistance are prefaced by significant decisions on how to deploy such support. Full translation (e.g., to Chinese) may be given as accompaniment to English Web pages, but this entails significant translation overheads and requires advance selection and preparation of content. Targeted dynamic native language support seems preferable, especially if implemented in an Web context that supports free browsing by the end-user.

We have explored two approaches to Web-based native language support. The earlier variety used concurrent frames to supplement English information with predefined local language support. This affords some flexibility in local language support but is limited by the fact that all of the native language 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 moved to a 'look up' system that dynamically checks word content of the main English text and derives local language definitions and explanations.

In our preferred approach, the user's Web browser is configured to use the proxy server when accessing external Web pages. The proxy requests an external page on the user's behalf but does not despatch the received data directly to the user's browser. Instead, the proxy server invokes an external program to scan Web page content for specified key words or phrases. Based upon the successful matches, an annotated version of the Web page is created 'on the fly' and sent to the local user. The proxy server may also cache the annotated Web pages or the original documents.

This technique of native language annotation promises significant benefits for individual users, with customisation available across two dimensions: level of language comprehension and modes of annotation. In the first case, the annotation system determines 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 English language ability, and based upon weightings attached to items within the annotation database. In the second case, a user may control the manner in which support is attached to the annotated Web pages. According to user wishes, this may use frame-based supplements in the local language, pop-up second language definitions and explanations, or added Web links to more detailed native language support, including dictionary and thesaurus. Such 'enhancements' to information delivery are managed via the proxy-based system that accommodates individual preferences or preset options appropriate for user comprehension level. This architecture is illustrated in Figure 2.

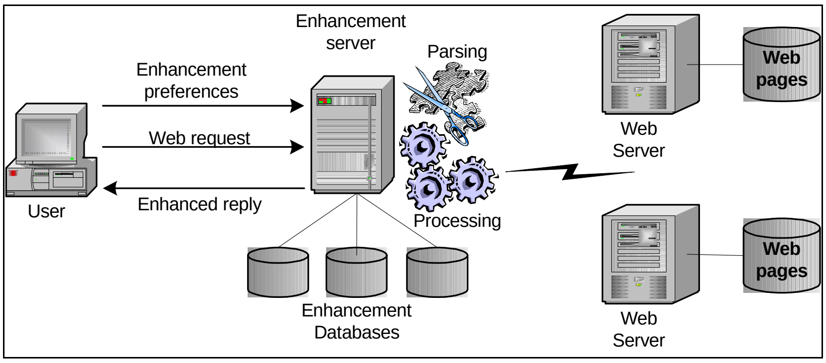


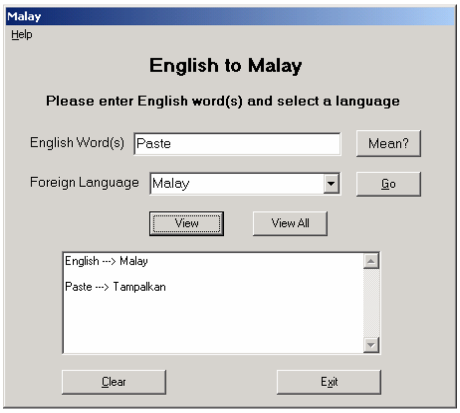
Figure 2: 'Enhancement server' architecture

3 Tools for Managing Mixed Languages[1]

Several software tools that we have developed for use in second language contexts address the need of the developer in managing native language support. The first of these tools provides an easy interface for recording and checking native language terminology (as used in our dynamic 'tool tip' support). A major overhead in creating such support is the need to generate a usable database of corresponding native language terminology. In this instance, we have two software components. The first allows the designer to enter individual English words or phrases, as might be found in Menu items or equivalent English tool tips. A target native language is selected (e.g., Malay) and the user composes or checks equivalent terminology (Figure 3).

An adjunct to this tool enables the terminology annotator to seek explanations of the English terms. This is accomplished via the 'Mean' button. When invoked, a search is activated that queries Web dictionary resources for insight on the original term or expression. For example, Figure 4 shows the result of checking the meaning of 'Paste'.

[1] We are grateful to Salim Al-Rawas and Hj Md Khairul Amilin Hj Idris, both of Strathclyde University, for their work on these program prototypes.



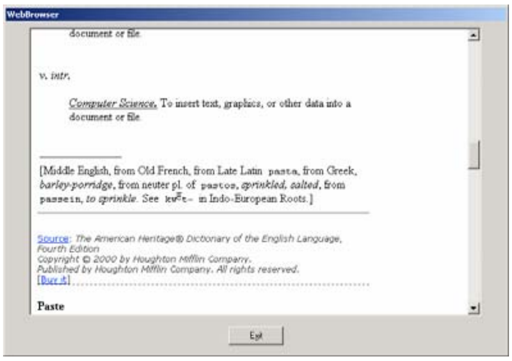
English Word(s): Paste

Foreign Language: Malay

```
English ---> Malay

Paste ---> Tampalkan
```

Figure 3: Checking corresponding English and Malay terminology



document or file

v. *intr.*

Computer Science. To insert text, graphics, or other data into a document or file

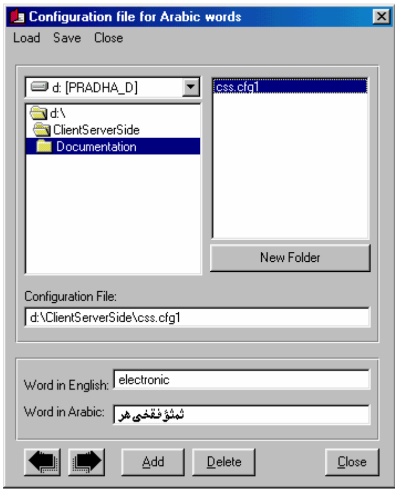
[Middle English, from Old French, from Late Latin pasta, from Greek, *barley-porridge*, from neuter pl. of pastos, *sprinkled, salted*, from passein, *to sprinkle*. See kwēt- in Indo-European Roots.]

Source: *The American Heritage® Dictionary of the English Language, Fourth Edition*
Copyright © 2000 by Houghton Mifflin Company.
Published by Houghton Mifflin Company. All rights reserved.
[Buy it]

Paste

Figure 4: Web dictionary lookup of 'Paste'

Our second tool implements a similar approach to the creation of Arabic tool tips in the context of English Web pages. Here, the program goes beyond merely recording terminological equivalents and directly generates native Arabic comments that can accompany the original English texts.



d: [PRADHA_D]

d:\
ClientServerSide
Documentation

css.cfg1

Configuration File: d:\ClientServerSide\css.cfg1

Word in English: electronic

Word in Arabic: ثمثؤفقخى هر

Figure 5: Defining Arabic equivalents for Web tool tips

For this application, a configuration file of keywords can be created off-line. In conjunction with a target set of English Web pages, the annotator runs this Arabic language support program. As a result, extra HTML and JavaScript code is added automatically (or interactively) to the source file to generate tool tips in Arabic (Figure 5).

4 Deriving criteria for second language support

The development and application of these tools is directed toward the second language problem. Their application raises further key issues that strike at the heart of second language support. In particular, we note the importance of providing a variety of different forms of language support. One dimension is the manner of presentation, e.g., selecting whether to provide native language supplements as tool tips, popup windows, status messages, or whatever). This addresses the question of how to convey information. A second, and more fundamental, dimension is the selection of message to convey in a specific presentational form. This selection process determines whether a simple translation, a summary, an example, a tutorial or some other purpose is addressed in the message delivery. This addresses the question **what** to convey to the user. Ultimately, the related issue of **where** (or **when**) to provide support rises as the most difficult question. This is the current thrust of our research in this area.

5 References

Lepouras G., & Weir G.R.S., 1999. It's not Greek to me: Terminology and the Second Language Problem. ACM SigCHI Bulletin 31, 2, Apr., ACM Press, New York, 17-24.

Lepouras, G., Vassilakis, C and Weir, G.R.S., 2002. Serving Enhanced Hypermedia Information, in Advances in Information Retrieval, 24th BCS-IRSG European Colloquium on IR Research, Crestani, F., Girolami, M. & vanRijsbergen, C.J. (Eds), Springer, Berlin, 2002, 86-92.

Weir, G.R.S. & Lepouras, G.,2001a. Localisation and linguistic anomalies, Proceeding of HCI International 2001, New Orleans, USA, 2001, 535-539.

Weir, G.R.S. and Lepouras, G., 2001b. English Assistant: A Support Strategy for On-Line Second Language Learning, ICALT 2001, IEEE International Conference on Advanced Learning Technologies, Madison, USA, 2001, 125-126.

Weir, G.R.S., Lepouras, G. & Sakelleridis, L, 1996. Second language help for Windows applications. In People and Computers XI, Proceedings of HCI'96, SASSE, M. A., Cunningham, R. J. & Winder, R. L. Eds., Springer, London, 129-138.