Working Paper Series
ISSN 1170-487X

Language Use in Software

by Alvin Yeo and Robert Barbour

Working Paper 97/20
July 1997

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Language Use in Software

Alvin Yeo
Department of Computer Science
University of Waikato, Hamilton, New Zealand.
Email: awy@cs.waikato.ac.nz

Robert Barbour
Science, Mathematics and Technology Education Research Centre
University of Waikato, Hamilton, New Zealand
Email: r.barbour@waikato.ac.nz

Abstract

Many of the popular software we use today are in English. Very few software applications are available in minority languages. Besides economic goals, we justify why software should be made available to smaller cultures. Furthermore, there is evidence that people learn and progress faster in software in their mother tongue (Griffiths et al,1994)(Krock,1996). We hypothesise that experienced users of English spreadsheet can easily migrate to a spreadsheet in their native tongue i.e. Bahasa Melayu (Malaysia's national language). Observations made in the study suggest that the native speakers of Bahasa Melayu had difficulties with the Bahasa Melayu interface. The subjects' main difficulty was their unfamiliarity with computing terminology in Bahasa Melayu. We present possible strategies to increase the use of Bahasa Melayu in IT. These strategies may also be used to promote the use of other minority languages in IT.

1. Introduction

Information Technology influences many facets of our daily lives. Consequently, many countries have or are now introducing computing at the pre-tertiary level. A drawback of such policies is that most, if not all, the software packages used are in English. Many companies do not translate software to other languages because they have the attitude that "...they (people who want to use the software) can learn English." (Sprung,1990). Instead of software which helps to improve learning, new users in non-English speaking nations have to learn English before they can use the software. Software exporters are adding another obstacle to the adoption of information technology (IT), that is, having to learn a new language. Not all countries have the technology and expertise to create software. New firms are beginning to create software for other languages realising that they can profit if they export to other countries. This trend is evident in the report that the top 100 US companies earn more than half their revenue from outside the US (Kay,1994).

Besides the profit taking motive, there are other reasons for creating software for use by minorities. By creating software for the rest of the world, the companies may be preserving some elements of the cultures of those nations. These elements include perspectives that may be the catalyst for potential solutions to future problems. Furthermore, the American Management Association reports that workgroups consisting of different cultures create solutions that are more innovative and effective than those workgroups consisting of only one culture (HR Focus,1993). By focussing on smaller cultures, software suppliers may be helping speakers of those languages to progress by gaining confidence in IT. Significant contributions to IT may not be realised (Griffiths et al,1994) if these countries were not able to adopt IT. By considering the language needs of these smaller cultures, the rate of extinction of language may be slowed. (NST (1997) reports that 90% of the world's languages will cease to exist within the next 100 years.)

It has also been reported that children learn and progress faster when using software that is localised to the indigenous language. Griffiths et al (1994) report that people who can interact with computers in their own language, learn and progress faster. For example, Bulgarian children only painted and drew but did not write in software in English. However, when provided with software in their mother-tongue, the children attempted to write creatively and explore all potential of the software (Griffiths et al,1994).

Given the above observation was made in a country where little English is generally spoken, we explore whether the same observation can be made about Malaysia i.e. if given a software in the native language in a bilingual country, will the people be able to use software "more easily"? It seems intuitively correct to assume that a person provided with a software package in his or her native language will be able to perform better in these conditions. Consequently we would like to determine if subjects in Malaysia can easily migrate their understanding of software concepts in English to another language such as Bahasa Melayu (Malaysia's national language). All the subjects in the study were familiar with software in English and were asked to evaluate a spreadsheet in Bahasa Melayu.

In the next section, we will give some background about language use in Malaysia. We then describe the methods used in our experiments in Section 2. The initial findings and observations are detailed in Section 3 and the analysis is described in Section 4. We summarise our findings in the last section.

1.1 Language Use in Malaysia

Malaya gained her independence from Britain in August 1957 and was joined by Sarawak and Sabah to form Malaysia in September 1963. Malaysia has a population of about 21 million (Dept. of Statistics,1996). The Malays and other indigenous groups make up 59% of the population, the Chinese, 32%, and the Indians, 9%. Bahasa Melayu is the official and national language. Besides Bahasa Melayu and English, Chinese dialects, Tamil and other ethnic languages are also prevalent. As Malaysia was under British rule, the use of English is quite widespread. Bahasa Melayu is used in official and administrative matters whereas in the business sector, English is the *lingua franca.*

From 1957 to 1967, Bahasa Melayu and English were the official languages. From 1967 onwards, Bahasa Melayu became the sole medium of instruction in secondary schools (Pakir,1994). Six years of primary school level education is available in Bahasa Melayu, Chinese and Tamil. To prepare the Chinese and Tamil-educated students for the secondary schools, these students enter "transition" (Bahasa Melayu-intensive) classes. Bahasa Melayu and English are compulsory in both primary and secondary education.

In the next section, we will detail the method used to conduct the study.

2. Method

The method we adopted is a variant of behavioural research with an ethnographic flavour in which we observed learners' actions while they used the specially designed software. We also interviewed them about their views of the software.

2.1 Hypothesis

We hypothesised that Bahasa Melayu native speakers would be able to use a software package with a Bahasa Melayu interface "without too many problems or much difficulty". We also hypothesised that the software package would be more intuitive in the local language.

2.2 Tools Used

We have access to the source code of a spreadsheet that could be modified to accommodate Bahasa Melayu and other languages such as Maori and English. The spreadsheet was developed as a result of research into localising software (Barbour and Yeo,1996). This spreadsheet was developed in such a way that a non-programmer can modify the spreadsheet to accommodate languages based on the Extended ASCII characters. Languages currently available for the spreadsheet include English, Mäori, Bahasa Melayu and Iban (an ethnic group's language in Sarawak, Malaysia). The translation of the spreadsheet interface was based on the *Istilah Komputer,* a collection of computer terminology published by the custodian of Bahasa Melayu, *Dewan Bahasa dan Pustaka* (the Institute of Language and Literature, Malaysia).

2.3 Subjects

We chose subjects who have had experience using spreadsheets in the workplace. We believe that an experienced spreadsheet user would be able to adapt to using a different spreadsheet tool more quickly than a non-spreadsheet user. For example, an experienced Excel user would be able to use another spreadsheet package by transferring his or her knowledge to the new software package, e.g. Lotus 123.

Eighteen subjects were asked to try out the spreadsheet in Bahasa Melayu. The subjects were all staff members of a Malaysian university that had a high IT presence. All subjects have been using spreadsheets for their daily tasks. Their reported average spreadsheet use ranged from about an hour a week to 3-4 hours a day. Users from different levels of the organisation were selected in order to acquire a representative view from the different levels of the organisation. The subjects comprised managers, lecturers, tutors, administrative assistants and clerks. (Refer to Table 1 for a breakdown of the subjects' position in the organisation).

Fifteen of the subjects were Malays whose mother tongue is Bahasa Melayu. There were five Malay males, and eight Malay females. There were also three non-Malays in the study, two males and one female.

Position	Prefer Bahasa Melayu	Prefer English	No Preference	**Total**
Managers		5	2	**7**
Lecturers		2		**2**
Tutors	1			**1**
Assistant Accountant		1		**1**
Administrative Assistant		1		**1**
Clerks	2	2		**4**
Data Processing Operator		1		**1**
Typists	1			**1**
Total	**4**	**12**	**2**	**18**

Table 1: The breakdown of the subjects' preferences

2.4 The Experiment

In the experiment, all the subjects were assigned to complete the same tasks in a spreadsheet with a Bahasa Melayu interface. The subjects were required to "talk aloud", that is, describe what they were thinking while they completed tasks using the spreadsheet. These tasks included entering data and formula into the spreadsheet, increasing the width of a column and changing the cells to currency format. After the talk aloud session, the users were required to fill in a Systems Usability Scale questionnaire. This questionnaire is described as a "quick and dirty" method to gauge the usability of software (Holyer,1994). The subjects were then asked questions to elicit their opinions about the Bahasa Melayu spreadsheet. The type of questions asked and responses to the interview as well as "talk aloud" observations are described in the next section. The complete results of the "talk aloud sessions" and the System Usability Scale will be presented in a future paper.

3. Results

After the subjects had completed their tasks and the questionnaire, they were asked some questions. One of the questions was, "If you were given a choice between the spreadsheet in English and another in Bahasa Melayu, which language will you choose?"

From the initial analysis in Table 1, four of the eighteen subjects tested preferred the Bahasa Melayu interface whereas twelve others preferred an English interface.

Two respondents did not have any personal preference as they were comfortable in both languages. One of the respondents mentioned that he would use a Bahasa Melayu interface if his subordinates were not proficient in English. The reasons given by the subjects for preferring English to Bahasa Melayu are summarised into three main categories:

1. Widespread use of English,

The subjects indicated that the use of English is widespread and that most, if not all, of the computer references are in English. Thus, to learn any software, a computer user would have to use English. Likewise, software upgrades in English would be faster than upgrades in other languages as original software would probably be in English. One subject commented that one would "do double the work if you do the translation (to Bahasa Melayu)". The subjects generally accepted that English is required for technological transfer. The level of proliferation of English is exemplified in this observation: when the subjects (native speakers of Bahasa Melayu and whose English were not fluent) when asked to "*simpankan hamparan*"(to save the spreadsheet), the users would refer it as "save" and not "*simpankan*".

2. Jargon or terminology,

   Another reason the English interface was preferred was because the Bahasa Melayu terminology was new and "alien" to the subjects, both Malays and non-Malays. The general consensus was that the interface was not clear to them and that they needed time to figure out what the various commands and text meant. Two subjects suggested that it would be easier to learn English and that there would be no need for translation. The original terms in English were more familiar and clearer to them than in Bahasa Melayu. For example, some subjects had problems with simple words like *Lajur* (column) and *Baris* (row). One subject mentioned that the Bahasa Melayu was "funny" in the context of computing software.

3. Bahasa Melayu proficiency

   The last reason described was the level of the subjects' Bahasa Melayu proficiency. Two of the subjects (non-Malays) preferred English because they felt that their English was better than their grasp of Bahasa Melayu.

From the "talk aloud" sessions, all the subjects had difficulties with the Bahasa Melayu interface. The main problem observed was that the commands were not intuitive (similar to reason 2 above). On many instances, the subjects "misinterpreted" the commands. In some cases, they could not understand the meaning of the commands, and thus guessed the meanings of the commands.

In the following analysis section, we suggest a possible rationale as why migration to the Bahasa Melayu software was not smooth.

	No Preference	Prefer Bahasa Melayu	Prefer English	**Total**
Malays	2	4	9	**15**
Non-Malays			3	**3**
Total	**2**	**4**	**12**	**18**

Table 2: Language Preference of Malays and Non-Malays

4. Analysis

Excluding the subjects who had no preference, nine of the remaining thirteen native speakers, preferred an English interface as opposed to only four preferring the Bahasa Melayu interface.

The results (see Table 2) indicate that even though a software package is provided in the native language, the speakers of that language would still prefer to use English and that they would have problems with the software in the native language form. This result is especially true if the users have not been exposed to computer terminology in their native language -only one of the eighteen subjects have seen a software in Bahasa Melayu. Although the language is their native language, it would appear that the subjects would need to learn the new jargon as if it was their first time using the software.

Thus, it would be unwise to expect users to easily migrate from using an interface in English to other languages. It can be argued that languages (like Bahasa Melayu and English) and computer language are separate entities. This argument is supported by Barbour and Keegan (1996) who reported that when translating a computer science course manual from English to Mäori, three languages are actually involved in the translation; English, Mäori and the computer language. Thus, in order for the Bahasa Melayu speakers to use the terminology, the Bahasa Melayu equivalent of the computer terms must first exist or are known by the user. If the terminology does not exist, then a substitute would be used, in this case, the English terminology. This argument may describe why the Bahasa Melayu speakers said "save" and not "*simpankan*" since the association of *"simpankan"* to the function "save" has not been made. Thus, it can be reasoned that the Bahasa Melayu speakers are absorbing the computer language into their own language by acquiring the English words and meanings as part of Bahasa Melayu.

From a linguistic view, the English jargon is simply additional words added to and absorbed into the local language usage patterns. English has grown in just such a fashion as is shown in the origins of many English words borrowed from Greek and Latin. Given that the Bahasa Melayu users know the computer terminology in Bahasa Melayu and were adept in English software, these users would have fewer problems migrating to Bahasa Melayu software. Learning would also be faster if the users understood the computing concepts. The current problem of users knowing only English software is because hardly any Bahasa Melayu software exist and also computing was not taught in the primary or secondary schools in Malaysia. In all universities in Malaysia, most computer courses are taught in Bahasa Melayu but software and references are mainly in English.

In the next section, we will put forward some suggestions to address this problem.

5. Discussion

The problem described in Section 4 suggests the promotion of Bahasa Melayu use in IT. One strategy is to introduce computing into the education system. In order to facilitate learning, computers could be used in many aspects of their education. Given that medium of instruction in primary education in Malaysia can either be in Tamil, Mandarin, or Bahasa Melayu, it would be a good idea to introduce Bahasa Melayu language lessons using CAI in Bahasa Melayu. On a positive note, the

Malaysian education system will be offering computing as a subject in the final two years of secondary education. This positive step will promote the use of Bahasa Melayu in IT.

Another strategy is to ensure there is a proliferation of Bahasa Melayu software in the market. However, the software must be as good as, if not better than, the software available in English. People will still use English software if the Bahasa Melayu alternative pales in comparison with the English software.

More incentives should be given to local software developers to create Bahasa Melayu software. Another strategy to obtain state of the art software is to internationalise the software. This strategy means that the software has been designed and developed to accommodate other languages in the localised phase. Localisation of internationalised software would thus be faster and cheaper. Techniques on internationalisation and localisation can be found in Nielsen and del Galdo (1996), Kano (1995), O'Donnell (1994), Uren et al (1993), Apple (1992), Digital (1992), Madell et al (1992), Taylor (1992) and Nielsen (1990). Translation ideas can be obtained from articles such as Aston and Dolden's (1994) paper, *Logiciel Sans Frontiers* which details Europe's initiative to increase access of software to a variety of languages and cultures.

The above suggestions also apply other countries in which English is a first or second language, but where minority languages (which may be in danger of extinction) also exists e.g. the Mäori language in New Zealand.

6. Summary

As observed in the software evaluation and interviews with users, the users' migration from English to Bahasa Melayu was not smooth. The observations suggest the Bahasa Melayu interface is no more intuitive than a new software. A majority of the Bahasa Melayu native speakers chose Bahasa Melayu over English because of the widespread use of the English and the unfamiliarity of the Bahasa Melayu terminology. The non-native speakers preferred English because their proficiency in English was better than in Bahasa Melayu. Thus, in order for Bahasa Melayu to be accepted, users should be taught the Bahasa Melayu terminology as soon as possible in their educational experience. The Malaysian Ministry of Education has taken a positive step in introducing computing at the secondary level. In addition to this, further effort must be made to promote Bahasa Melayu in IT.

References

1. Apple Computer. (1992). *Guide to Macintosh Software Localization.* Addison Wesley.
2. Aston, M. and Dolden, B. (1990). Logiciel Sans Frontiers. *Computers in Education.* Vol 22. No 1/2.
3. Barbour, R. and Te Keegan, T. (1996) : Education in Technology for the LCTLs: Nga Tautono Rorohiko. *NFLRC Symposium* (University of Hawai'i, 8-12 July). http://www.lll.hawaii.edu/nflrc/NetWorks/NW2/morn.html

4. Barbour, R. and Yeo, A. (1996). Internationalising a Spreadsheet for Pacific Languages. *New Zealand Journal of Computing.* (Forthcoming)

5. del Galdo, E. and Nielsen, J (ed.). (1996). *International User Interfaces.* Academic Press.

6. Department of Statistics. (1996). *Yearbook of Statistics (Malaysia) 1996.* Department of Statistics, Malaysia.

7. Digital Equipment Corporation. 1992. *Digital Guide to Developing International Software.* Digital Press.

8. Kano, N. (1995). *Developing International Software for Windows 95 and Windows NT.* Microsoft Press.

9. Kay, R. (1994). Software Goes Global. BYTE. 19. 6. p90-91.

10. Griffiths, D., Heppell, S., Millwood, R. and Mladenova, G. (1994). Translating Software: What It Means and What It Costs for Small Cultures and Large Cultures. *Computers in Education.* Vol. 22. No. 1/2. p9-17.

11. Holyer, A. Methods for Evaluating User Interfaces. Cognitive Science Research Paper No. 301, School of Cognitive and Computing Sciences, University of Sussex, http://www.cogs.susx.ac.uk/cgi-bin/htmlcogsreps?csrp301, 1994

12. HR Focus. (1993). More Companies Are Drawing Strength from Diversity. *HR Focus.* Vol. 70. No 6. p2.

13. Krock, E. (1996). Cultural Learning Differences in Software User Training. In del Galdo, E. and Nielsen, J. (Ed.) *International User Interfaces.* John Wiley and Sons, Inc. USA.

14. Madell, T., Parsons, C. and Abegg, J. (1992) *Developing and Localizing International Software.* Hewlett Packard Professional Books.

15. Nielsen, J. (Ed.). (1990). Designing User Interfaces for International Use. Elsevier, New York.

16. NST. (1997). Expert: 90pc of the world's languages facing extinction. *New Straits Times (Malaysian Edition).* 22 April 1997.

17. OfDonnell, S. (1994). *Programming for the Whole World: A Guide to Internationalization.* Prentice Hall.

18. Pakir, A. (1994). Making Bilingualism Work: Development in Bilingual Education in Asean. In Khoo, R., Kreer, U. And Wong, R. (Ed.) *Towards global Multilingualism: European Models and Asian Realities.* Multilingual Matters Ltd., Clevedon, England. p13-27.

19. Sprung, R. (1990). Two faces of America: Polyglot and Tongue-tied. In Jakob Nielsen (Ed.) *Designing User interfaces for International Use.* Elsevier, New York. P71-102.

20. Taylor, D. (1992). *Global Software: Developing Applications for the International Market.* Springer-Verlag.

21. Uren, E, Howard, R. and Preinotti, T. (1993). *Software Internationalization and Localization: An Introduction.* Van Nostrand Reinhold.