Decoding Programming Concepts in EAL and EFL Students: Is there a difference?

Christine Prasad
Unitec New Zealand
Auckland, New Zealand
cprasad@unitec.ac.nz

Kathryn E. Sanders
Rhode Island College,
Providence, RI, USA

John A. McTaggart
Drake University, Des Moines, IA, USA

This paper reports on an exploratory study carried out with a subset of the data collected from the NSF-funded Bootstrapping Research in Computer Science Education project (http://depts.washington.edu/ bootstrp. The primary focus of this study was to investigate whether there are differences in the internal representation of computer programming concepts between first-competency students for whom English is the first language (EFL), and those for whom English is an additional language (EAL). Due to the exploratory nature of this study and the small sample size, the results are not conclusive. They do, however, suggest that the two groups (EFL and EAL) represent concepts in surprisingly similar ways.

Keywords

English, Additional Language, First Language, Programming Concepts.

1. INTRODUCTION

Computer Science (CS) is taught in the context of an increasingly multi-cultural and multi-lingual society today. A typical CS classroom now has many students for whom English is an additional language (EAL), with the language of instruction being English. This has crucial educational implications for the EAL learner. The constant translation of information between languages can affect the student at the surface level of basic comprehension of lectures, assessment and course materials, and at the deeper level of actually acquiring concepts from the course (Collison, 1974 cited in Baker (1998), page 51).

Computer Science adds another level of complexity. Students who are already dealing with the complications of translating information between multiple languages have to cope with the "language of CS". Chmura (1998) asked what students found most difficult about programming and got the responses: "new terms/vocabulary – new meaning of words I already know in English" and "reading and understanding the text."

There have been few studies within CS education that investigate the relationship of EAL/EFL status and success in introductory programming. There are the obvious difficulties inherent in taking courses presented in a foreign language. Giridharan and Enriquez (2002) reported on the "challenges facing lecturers in the offshore classroom scene, in both the teaching of English and IT skills," and noted that EAL students have to deal with a "foreign curriculum" and "the computer which undeniably 'speaks' largely in English." Kurtz (1980) observed that native speakers, or students for whom English is the first language (EFL), performed at higher levels than EAL students on a written test of formal reasoning. He attributed the difference to linguistic difficulties with the test, since the two groups' programming scores were the same.

Better English language skills might be positively correlated with success for both EAL and EFL students. Leeper and Silver (1982) reported a strong correlation between verbal SAT scores and success in a first programming course. Within the specific area of learning computer programming, Master (1983) reported briefly on the relationship between language skills and computer programming skills. Wills (1982), in response to a study by Mazlack (1980), suggests the possibility of analysing programming skill with relation to linguistic capability.

On the other hand, the ability to learn a foreign language (which EAL students have already demonstrated) might be positively correlated with the ability to learn a programming language. Evans and Simkin (1989) considered thirty-four variables that might be used to predict programming success, and conjectured that "the ability to speak a second language might be helpful in learning a programming language". Byrne and Lyons (2001) considered success in introductory programming in comparison with grades in previous foreign-language courses and found no major correlation; they did not report the EFL/EAL status of their participants.

This paper reports on an exploratory study that investigates the way in which EAL and EFL first-competency students of computer programming mentally represent computer programming concepts. In this study, first-competency students are those who have completed at least two courses in programming prior to taking part in this research. Our investigation is an outgrowth of the NSF-funded Bootstrapping Research in Computer Science Education Project (http://depts.washington.edu/bootstrp), (Petre, Fincher, and Tenenberg et. al., 2003), which explored the conceptual structures that first-competency programmers create for programming constructs. Given that "different populations exhibit different concept organization structures" (Petre, Fincher, and Tenenberg *et. al.*, 2004), we are specifically enquiring whether EAL and EFL students form similar mental representations of programming concepts.

2. METHOD

The study design is reported in detail in Petre, Fincher and Tenenberg *et al.* (2003, 2004). The primary method of data collection was a repeated, single-criterion card sort (Rugg and McGeorge, 1997) to elicit students' internal representations of programming concepts. This consisted of providing 26 cards to participants, each with different a programming concept, and asking them to sort the cards according to some criterion. They were given concrete examples of sorting like cereal boxes and chocolates that can be sorted according to the criteria: colour, taste, preference, price etc. Cardsort data was recorded verbatim. Additional background information for participants was gathered including age, gender, and experience with various programming languages.

In addition to the background data gathered for participants in the Bootstrapping Project, we collected native language information from twenty-nine participants, representing three institutions, out of a total of 276 participants. Fifteen participants from this subpopulation reported English as a First Language (EFL), and fourteen participants reported English as an Additional Language (EAL). There were twenty-four males and five females in our subpopulation. The EFL and EAL participants were evenly distributed across the institutions: one institution had six of each, one had five of each, and one had four EAL participants and three EFL participants.

We used the data analysis mechanisms provided by the Bootstrapping Project toolkit (Petre, Fincher, and Tenenberg *et. al.*, 2004). In particular, we used four techniques:

- Comparative analysis: We compared the quantities of sorts and categorizations within the different groups.
- Verbatim analysis: We looked at individual sort criteria, categories within each sort, and the specific cards placed in each category.
- Gist analysis on sorts: We looked for similar criteria for sorting, although the names for individual criteria could have differed.
- Dendrograms: We generated dendrograms to provide collective views of our subpopulations, and then used a clustering method (Aldenderfer, 1984) to focus on similarities and differences in card groupings. Separate dendrograms were generated for the EFL and EAL subject groups, EFL and EAL subject groups by institution, and each institution as a whole.

3. ANALYSIS

We carried out a preliminary, exploratory examination of the data using a variety of techniques in order to identify potential questions for further analysis.

3.1 Comparative Analysis

We examined the data concerning EAL and EFL participants from several points of view. First, we considered the average number of sorts and the average number of categories per sort. As noted in Petre, Fincher and Tenenberg et al. (2003, 2004), these figures seem to increase with the performance level of the subject: in the overall sample, better participants tended to have higher numbers.

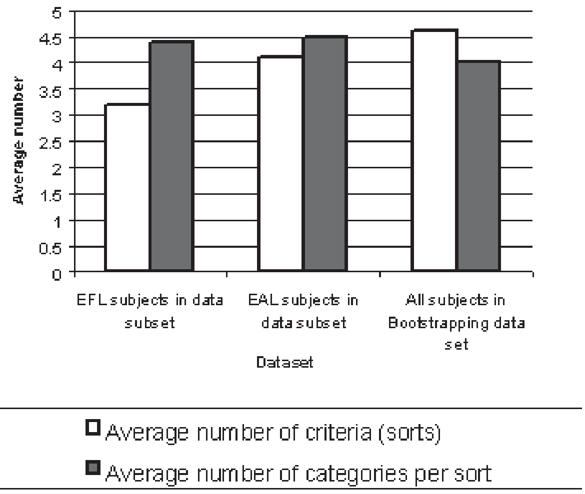

Figure 1. Average number of criteria and categories per student data set.

EAL participants, on average, achieved more sorts than EFL participants (4.1 EAL vs. 3.2 EFL), but the number of categories per sort is essentially identical for both EAL and ESL participants (4.5 EAL vs. 4.4 EFL). The average number of sorts is lower than the overall average for the entire Bootstrapping data set (4.6), but the average number of categories per sort is higher than the overall average (4.0).

3.2 Verbatim Analysis

3.2.1 Category and Criterion Names

For both EFL and EAL participants, it was generally easier to sort the cards than to articulate a name for the categories or for the criterion they were using. Both groups of participants occasionally used a name an expert would recognize, but unfamiliar names were much more common in both groups. We did not find a major difference between EAL and EFL participants in this respect.

For example, recognizable criteria included "object-oriented and non-object-oriented" (EFL) and "lower level versus higher level" (EAL). Unfamiliar criteria included "things and a way to view it," "what goes into the program," and "things that are likely to be used together" (EFL), and "used one with another," "how you manipulate," and "what objects do and everything else" (EAL).

The same hold true for categories. Recognizable categories included "things I see in OO programming" (EFL) and "control structure" (EAL). More frequently, however, the participants did not seem to have the vocabulary to name the categories they identified. For example, EFL participants mentioned "things that you do," "quality that applies to something," and "technique that enables you to apply previous group." EAL participants used "in the program stuff," "how everything is related," and "don't know name, but they go together."

3.2.2 Ragbag Categories

"Ragbag" categories are those that participants classify as a group without a known or certain label. According to Rugg and McGeorge (1997), ragbag categories indicate uncertainty. Examples of ragbag categories in our data are: "don't know," "don't apply," "don't belong," "don't fit anywhere," "don't know as well," "don't know what it is," "don't know what they are," "everything else," "miscellaneous," "not applicable," "no name," and "no category."

In our data, analysis of ragbag categories revealed further similarities between the EAL and EFL subpopulations. EAL and EFL categories were almost equally likely to be ragbag categories. Seventy-three of 473 categories (15.4%) were ragbag categories. Forty-two of 270 EAL categories (15.6%), and 31 of 203 EFL categories (15.3%) were ragbag categories. All the participants in our subpopulation had at least one sort with a ragbag category, with the exception of one (EAL) subject.

EAL and EFL sorts were equally likely to contain a ragbag category. Seventy of the 105 sorts (66.7%) contained at least one ragbag category. Of these, one sort contained three ragbag categories; one contained two; and the other 68 all contained one ragbag category each. Forty of the 60 EAL sorts (66.7%) and 30 of the 45 EFL sorts (66.7%) contained at least one ragbag category. Of the sorts that contained two or more ragbag categories, an EFL subject did one and an EAL subject did one. All the stimuli used in the sorts were found in at least one ragbag category.

Table 1 shows the most common stimuli occurring in ragbag categories, and the frequency with which they occur in each subpopulation. The numbers in the table are the percentage of ragbag categories in which a particular term occurred (rounded to zero decimal places). For example, "abstraction" was included in 43% of the EAL ragbag categories and 26% of the EFL ragbag categories. The table includes all the terms that were included in at

Table 1. Frequency of most common stimuli in ragbag categories.

Stimuli	EAL (% of ragbag categories)	EFL (% of ragbag categories)
Abstraction	43	26
Choice	36	32
Decomposition	50	61
Dependency	48	42
Encapsulation	43	16
State	43	39
Thread	57	65
Tree	36	32

least 30% of the EAL ragbag categories and/or at least 30% of the EFL ragbag categories.

All the stimuli shown in Table 1 are representative of abstract programming concepts as described in Petre, Fincher and Tenenberg et al. (2004). This suggests that both EAL and EFL participants are almost equally uncertain about abstract concepts and have a similar degree of uncertainty about these concepts.

3.3 Gist Analysis

An examination of the criteria presented by the different participants revealed the following main gist.

3.3.1 Program Related

These criteria suggest that a subject is relating concepts to programs only. Examples of such criteria are "composing a basic program" (EAL), "how it is used in the programs" (EAL), "Writing a program/function" (EAL), "the way things work in programs" (EFL), "what goes into a program" (EFL), "what programs are made of" (EFL), "What you expect to find in areas of code" (EFL). Twenty two percent of EAL participants sorted the stimuli according to program-related criteria, while 36% of EFL participants chose program-related criteria. This suggests that more EFL participants relate programming concepts to the higher level of writing programs.

3.3.2 Relationship-oriented

These criteria were those that suggested a connection between concepts. Examples are "How they are related" (EAL), "more closely-related objects" (EAL), "Their connection" (EAL), "Relationships that jump out at me" (EFL), and "Hierarchical" (EFL). Eight percent of EAL participants and 9% of EFL participants used relationship-oriented criteria. No meaningful differences were seen.

3.3.3 Strategy-oriented

These criteria indicate that participants were applying a strategy to the way they thought about concepts. Such criteria were often named with questions rather than descriptions. Examples are:

- "Does this help in calling a function?" (EAL);
- "Does this hold a variable?" (EAL);
- "Necessity to make an applet/GUI" (EAL);
- "Things that are likely to be used together" (EFL).

A larger proportion of EAL participants had strategy-oriented criteria: 10% EAL and 2% EFL. These results suggest that EAL participants tend to create strategies to make sense of programming concepts.

The observations within program-related and strategy-oriented criteria suggest differences in the two groups. However, it is premature to draw any conclusions from these without clarifying the implied meanings of the terms used to label the criteria. An in-depth study of the laddering data Petre, Fincher and Tenenberg et al. (2004) could provide greater insight into the implied meanings of the terms used by these participants and can form a basis for further research in this area.

3.4 Dendrograms

We computed dendrograms for both groups using the EZSort tool (http://www-3.ibm.com /ibm/easy/eou_ext.nsf/Publish/410), as shown in Figures 2 and 3 below. We acknowledge that dendrograms provide an average representation of the data. For an exploratory study like ours, they provide a high level indication of the responses of the two subpopulations under study.

Analysis of the dendrograms reveals additional similarities between our two subpopulations. Each dendrogram is divided into two subtrees of 10 and 16 concepts, where the first subtree is primarily abstractions and the second, larger subtree includes primarily more concrete code-related terms. Eight of the ten concepts in the first subtree are the same in both diagrams, and 14 of 16 concepts in the second subtree. Both dendrograms have a depth of seven (counting the number of splits before you reach a concept) and an average depth of 5.0 (rounded to one decimal place).

In both cases, the second "concrete" subtree is broken down into three main subtrees, which we might label process, data, and flow of control. In both dendrograms:

- The "process" subtree contains the concepts "function," "method," and "procedure."
- The "data" subtree contains the concepts "Boolean," "type," "variable," and "constant."
- The "control" subtree contains the concepts "loop," "if-then-else," "iteration," and "recursion."

In other words, of the 14 concepts that are in both concrete subtrees, 11 are placed in the same location within the concrete subtree.

The three exceptions are "object," which is associated with process by the EAL participants and with data by the EFL participants; "parameter," which is associated with data by the EAL participants and with process by the EFL participants; and "array," which is associated with control structures by the EAL participants and with data by the EFL participants. In each case, both groups seem to be making defensible choices.

Most interestingly, if we look at the closest associations, in both subpopulations, 11 of the 26 concepts are associated in the same way. These groups include "variable" and "constant"; "method," "function," and "procedure"; "if-then-else" and "loop"; "iteration" and "recursion"; "choice" and "expression."

The participants have a better understanding of the concrete concepts than the abstractions. If we look at the data, abstractions are often correlated because they are all in the "don't know" category. This is reflected in the fact that the abstractions are generally in a separate subtree, and that subtree itself is less deep (a maximum depth of 5 for abstract concepts vs. 7 for concrete ones, in both subpopulations).

It appears that participants' understanding, as expressed in their groupings, is better than one would think from reading through their responses. They are better at constructing groupings than they are at labelling them, and this too seems to be language-background-independent.

4. SUMMARY/ CONCLUSION

In this preliminary, exploratory analysis, the results for EAL and EFL participants were surprisingly similar. EAL participants achieved more sorts than EFL participants (on average), but the average number of sorts per category was nearly identical.

For both EAL and EFL participants, it was easier to sort the cards than to articulate a name for the categories or for the criterion they were using. On average, the ways in which the two groups sorted the cards (i.e., the cards that were generally associated with each other) were also similar, as shown in the dendrograms computed from the responses of the two subpopulations. In addition, the two groups were approximately equally likely to use ragbag, or "don't know," categories. The most common stimuli in the ragbag categories, in both cases, were the more abstract stimuli, suggesting that both groups struggle with the more abstract aspects of programming. It was only when we looked at the gist of the criteria used by these two different groups of participants that we found some possible differences.

Because of the small size of our sample, it is not possible to draw definite conclusions from this study. It does suggest, however, that the way in which students organize their knowledge of programming concepts may not be language-dependent, perhaps because programming involves techniques of formal reasoning, critical thinking, and problem solving that reach beyond the language of instruction.

Interesting directions for future work include repeating the same experiment with a larger sample size, specifically focusing on the question of whether there are differences between EAL and EFL students, and examining the laddering data to gain greater insight into the gist of the criteria used by the participants.

ACKNOWLEDGEMENTS

The Bootstrapping project (http://depts.washington.edu/bootstrp) was supported by the National Science Foundation Grant No. DUE-0122560. We are grateful to the organizers of the project for providing us with the guidance and opportunity to undertake an experiment of this nature.

REFERENCES

Aldenderfer, M. S., and Blashfield, R. K. (1984). Cluster Analysis (Sage University Paper series on Quantitative Applications in the Social Sciences, No. 07-044). Beverly Hills, CA: Sage.

Baker, D. E. (1998). A study of the effect of culture on the learning of science in non-Western countries. PhD Thesis, Science and Mathematics Education Centre, Curtin University, Perth, Australia. Retrieved 20 May 2003 from Curtin University http://adt.curtin.edu.au/theses/available/adt-WCU20021209.144123/

Byrne, P. and Lyons, G. (2001). The Effect of Student Attributes on Success in Programming. ACE SIGCSE Bulletin, Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education, Vol. 33. No. 3, pp 49-52.

Chmura, G. A. (1998). What Abilities are Necessary for Success in Computer Science? SIGCSE Bulletin, Vol. 30. No. 4, pp 55a – 58a.

Evans, G. E. and Simkin, M. G. (1989). What Best Predicts Computer Proficiency? Communications of the ACM 32 (11), pp1322-1327.

Giridharan, B. and Enriquez, J. G. (2002). English language and IT learning challenges in Australian offshore campuses. In Focusing on the Student. Proceedings of the 11th Annual Teaching Learning Forum, 5-6, Perth: Edith Cowan University. Retrieved 11 March 2004 from http://lsn.curtin.edu.au/tlf/tlf2002/giridharan.html

Kurtz, B. L. (1980). Investigating the Relationship Between the Development of Abstract Reasoning and Performance in an Introductory Programming Course. SIGCSE Bulletin, pp110-117.

Leeper, R. R. and Silver, J. L. (1982). Predicting Success in a First Programming Course. SIGCSE Bulletin 14(1), pp147-150.

Master, P. (1983). Computer programming and ESP. CATESOL News 14(5), pp 5 and 14.

Mazlack, L. J. (1980). Identifying Potential to Acquire Programming Skill. Communications of the ACM 23(1), pp14-17.

Petre, M., Fincher, S., Tenenberg, J. et. al. (2003). "My Criterion is: Is it a Boolean?": A card-sort elicitation of students' knowledge of programming constructs. Technical Report 6-03, Computing Laboratory, University of Kent, Canterbury, Kent, UK. Accessed 11 March 2004 from http://www.cs.kent.ac.uk/pubs/2003/1682

Petre, M., Fincher, S., Tenenberg, J. et. al. (2004). A large-scale elicitation of students' knowledge of programming constructs. In print.

Rugg G., and McGeorge, P. (1997). The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. Expert Systems 14(2), pp 80-93.

Wills, J. A. (1982). Are Programming and Natural Language Skills Related? Communications of the ACM 25 (3), p 221.