The Life and Death of Software: Examining the Selection Process

Thomas N. Robb
Kyoto Sangyo University

Bernard Susser
Doshisha Women’s Junior College

ABSTRACT

This article describes the results of an Internet-based survey asking how foreign language instructors chose courseware. We first review the literature on software selection, looking particularly at methods and tools. We then report the results of a questionnaire advertised on several Internet lists for foreign language teachers. The main factors on which teachers base their selection were reported to be testing the software and the features advertised. We examine the effectiveness of various selection methods and describe problems in the selection process. This study does not present statistically valid conclusions because of the small number of responses, but it does suggest some problems with present methods of software selection.

KEYWORDS

CALL, Software Selection, Software Evaluation, Checklists, Frameworks

INTRODUCTION

Software selection is one of the first tasks in establishing a CALL program. A considerable literature exists which offers advice to teachers on how to choose software that will be suitable for their students, syllabus, and curriculum. On the other hand, the actual dynamics of software acquisition, while mentioned occasionally in the literature, has not, to our knowledge, been studied in detail. This study attempts to determine, by means of a questionnaire, which of the many potential sources of informa-
tion concerning software have in fact been important for the selection of CALL courseware. We wanted to find the answers to questions such as the following:

What is the most influential source of information for software purchases?
Do most people try out software before they purchase it?
Does software purchased by people who try it first tend to be better suited and more frequently used by their students?
Does the use of a checklist assure the purchase of more usable software?

Our conclusions, while tentative, suggest that there is gap between the advice in the literature and what teachers actually do to select courseware.

BACKGROUND

Teachers regularly select the materials they use according to how well those materials match the syllabus, course aims, teaching methodology, etc. (see, e.g., Cunningsworth, 1984; Sheldon, 1988; and the bibliography in Rea-Dickins, 1994). However, courseware presents some special problems, as Johnston (1987) has pointed out: “the evaluation of micro-computer programs is unlike that of learning materials in other educational media in general, and of the book in particular, in that software is not readily accessible to inspection by skimming, scanning and the selection of random parts for detailed examination” (see also Hubbard, 1987). Specifically, courseware’s multimedia elements, interactive aspect, and hierarchical structure all demand specialized expertise and much more time than that required for evaluating textbooks. Further, the constant development of new technologies, such as courseware delivered over the Internet, puts further burdens on the teacher in search of suitable materials for students. In short, courseware selection is a “painful process” (Hopey, Rethemeyer, & Elmore, 1995).

A number of resources are available to ease this pain. The TESOL CALL Interest Section Software List publishes a large list every other year with much useful information about English as a Second/Foreign Language courseware (Healey & Johnson, 1999). For educational software in general, teachers can consult large scale projects such as TLT’s “Flashlight Program” (www.tltgroup.org/programs/flashlight.html) and the Educational Software Selector (TESS), a database produced by The Educational Products Information Exchange (EPIE) (www.epie.org/epie_tess.htm). The International Society for Technology in Education (ISTE) has a large online database called a “preview guide” (wfm.sd38.bc.ca:8005/ESP/

Several means of selecting courseware have been suggested in the literature. Most common may be intuitive suggestions based on long experience. The expert teachers in Healey’s (1994) survey pointed to qualities such as depth, interactivity, independent interface, compatibility, etc. (see also Healey & Johnson, 1997). The evaluating organizations that contribute to the Only the Best series look for technical matters such as ease of installation, teacher options that are easy to find and use, levels of interactivity, etc. On the educational side, they look for solid content, age appropriateness, and relative value (i.e., how much the program costs in comparison to what it can deliver to users). Higgins (1995) lists among his “tools” for software evaluation “sales figures” and “the amount of imitation or plagiarism” as indicators of a given program’s popularity. Decoo (1994) suggests the TPH unit—“time price per hour”—as a means of evaluating the “market value of educational software.” Bishop (1999) advises asking a few “policy-related questions” on the relationship between the courseware and the teacher’s educational philosophy and specific teaching objectives.

The most common tools for courseware selection in the literature are checklists and frameworks. Many published checklists appear literally as checklists with a series of questions or statements to be checked off “yes/no,” marked 1-5 on a Likert scale, or with blanks to be filled in, such as amount of RAM required. (For several examples, see Squires & McDougall, 1994.) Other formats include lengthy text explanations that are then summarized in a list or checklist-like guidelines written up in prose style.

The most important framework for CALL courseware is Hubbard’s (1987, 1988, 1992, 1996). He defines “framework” as “an integrated description of the components of something—in this case CALL materials—with respect to a particular goal—in this case evaluation” (Hubbard, 1988). He emphasizes that “rather than asking a specific set of questions, a framework provides the tool through which an evaluator can create his or her own questions or develop some other evaluation scheme” (1988; see also 1996). In his first article (1987), the framework appeared as a checklist, but in the later articles as a network of linked modules. P. Hubbard (personal communication, June 9, 1999) emphasized that “the difference between my framework and a checklist is meant to be the difference between telling someone what to look AT and what to look FOR.” In other words, Hubbard is concerned that checklists may lock teachers into a mechanical evaluation emphasizing the presence or absence of a
given feature (looking “for”) instead of considering (looking “at”) what role the presence or absence of that feature plays given the “overall conceptualization and objectives of the software” (P. Hubbard, personal communication, September 4, 1999).

Two studies support Hubbard’s position by showing the problems of establishing parameters for courseware evaluation. Borton and Rossett (1989) used a survey to show that courseware evaluators, courseware developers, and teachers do not agree on the relative importance of standard courseware attributes. More recently, Murphy (1995) identified from the literature six “basic essentials” of good software: good documentation, learner control, branching capabilities, portability, ease of use, and cost effectiveness. However, she then interviewed four “computer-knowledgeable” doctoral students majoring in instructional computing and found that only four of these “basic essentials” were even mentioned by all interviewees and that all interviewees mentioned five items not on her list: color, graphics, sound, usefulness, and applicability. One would hope that “usefulness” might be considered a “basic essential” of any learning material. Murphy’s study showed that establishing absolute evaluation criteria for courseware is not an easy task. The problem is compounded by the fact that any given feature, such as user control, “may or may not facilitate learning, depending on [the] context” (Bangert-Drowns & Kozma, 1989).

This brief survey has shown that while courseware selection is difficult and “painful,” numerous resources and a body of research on evaluation are available to assist teachers in their decision-making process. To what extent are these resources and research being used in practice? Our task in this article is to describe the actual dynamics of software acquisition.

METHOD

A questionnaire was created as a cgi-enabled web page and its existence was advertised on TESLCA-L, NETEACH-L, JALTCALL, and other listserves. Despite a number of messages with various degrees of pleading, we received a total of only 51 usable responses. The sample is thus too small to draw any concrete conclusions, and we have not attempted to run statistical tests of significance on the results rather opting to report them simply as percentages. Naturally, the small sample size, plus a bias resulting from only web-savvy respondents on the above mailing lists having responded, may well have skewed the results. These caveats aside, we still hope that the results will provide a stimulus for a future study with greater validity.
Frequency of Selection Criteria Use

A list was developed of the various sources of information that might affect decision making (see Table 1).

Table 1
Sources of Information Potentially Affecting Purchasing Decisions

Which of the following entered into your decision to purchase the software?

<table>
<thead>
<tr>
<th>Applicable (A)</th>
<th>Somewhat (S)</th>
<th>Not Applicable (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>b. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>c. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>d. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>e. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>f. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>g. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>h. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>i. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>j. A</td>
<td>S</td>
<td>N</td>
</tr>
<tr>
<td>k. A</td>
<td>S</td>
<td>N</td>
</tr>
</tbody>
</table>

I obtained a copy or a demo for trial use and ...

- evaluated it by myself or with colleagues
- had some students try it
- used it under actual class conditions
- checked its features/operations against an evaluation checklist

The results are tabulated and rank ordered in Table 2. For the purpose of this analysis, responses of “Applicable” and “Somewhat Applicable” were grouped together.
The Life and Death of Software

Table 2
Rank Order of Responses

<table>
<thead>
<tr>
<th>Method</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>h. Obtained a copy/demo and evaluated it (self or w/colleagues)</td>
<td>81%</td>
</tr>
<tr>
<td>g. Advertised features</td>
<td>66%</td>
</tr>
<tr>
<td>i. Obtained a copy/demo and had some students try it</td>
<td>56%</td>
</tr>
<tr>
<td>j. Obtained copy/demo and used it under class conditions</td>
<td>52%</td>
</tr>
<tr>
<td>d. Vendor’s presentation</td>
<td>37%</td>
</tr>
<tr>
<td>a. Recommendations of colleagues</td>
<td>37%</td>
</tr>
<tr>
<td>f. Favorable reviews in the literature</td>
<td>31%</td>
</tr>
<tr>
<td>k. Obtained a copy/demo and used a checklist</td>
<td>25%</td>
</tr>
<tr>
<td>e. Software guides</td>
<td>24%</td>
</tr>
<tr>
<td>c. Recommendations at an academic meeting/presentation</td>
<td>21%</td>
</tr>
<tr>
<td>b. Testimonials over the Internet</td>
<td>15%</td>
</tr>
</tbody>
</table>

The main two factors would appear to be testing the software (h, i, j) and the advertised features of the software (g). In all cases where the software was tested by students or under class conditions, the instructor (or colleagues) had first tried it. In our survey, 67% of the software was previewed either informally by some number of students or under actual class conditions. Despite the emphasis on checklists in the literature, only 25% of the software was reviewed against a checklist before purchase.

Effectiveness of Selection Methods

The next question of interest here is how effective these various methods of evaluation are. One measure of effectiveness is the degree to which the software evaluated by each method is still in regular use. This measure is not perfect because there may be external reasons for discontinued use such as changes in the curriculum which could adversely affect the usage of any given program. However, hardly any responses indicated such external pressures.

The questionnaire asked respondents to state whether the software was (a) still being used on a regular basis, (b) seeing less use than originally planned, or (c) no longer being used. Table 3 shows the selection criteria associated with software still being used, grouped with similar percentages (< 6%).
Table 3
Selection Criteria Related to Software “Still Used on a Regular Basis”

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Recommendations of colleagues</td>
<td>78%</td>
</tr>
<tr>
<td>2. Obtained a copy/demo and evaluated it (self or w/colleagues)</td>
<td>71%</td>
</tr>
<tr>
<td>3. Testimonials over the Internet</td>
<td>71%</td>
</tr>
<tr>
<td>4. Obtained a copy/demo and had some students try it</td>
<td>70%</td>
</tr>
<tr>
<td>5. Recommendations at an academic meeting/presentation</td>
<td>67%</td>
</tr>
<tr>
<td>6. Obtained a copy/demo and used it under class conditions</td>
<td>67%</td>
</tr>
<tr>
<td>7. Favorable reviews in the literature</td>
<td>67%</td>
</tr>
<tr>
<td>8. Vendor’s presentation</td>
<td>61%</td>
</tr>
<tr>
<td>9. Advertised features</td>
<td>61%</td>
</tr>
<tr>
<td>10. Obtained a copy/demo and used a checklist</td>
<td>50%</td>
</tr>
<tr>
<td>11. Software guides</td>
<td>45%</td>
</tr>
</tbody>
</table>

“Recommendations of colleagues” was reported as the most influential source of information. Recommendations from other impartial sources (e.g., academic meetings, Internet, reviews, etc.) and actual trialing of the software formed the second cluster of most effective evaluation strategies. Software guides and evaluating the software against a checklist were not as strong predictors of “successful” software as other sources of information.

One respondent outlined his department’s selection process and claimed that thanks to this selection process, all of the software is still in use. We consider this to be an excellent model of software evaluation.

When we started our lab in 1994, I took Healey & Johnson and reviewed it for possible software purchases. I condensed the relevant reviews for distribution to teachers and coordinators here. They indicated preferences and prioritized them. I ordered and reviewed preview copies. If I could figure out how to use it and thought it would be manageable for students, I did a users write-up for teachers. They then reviewed the software and decided if it would be suitable to use for their course. Software was rejected at both stages in this process. Of the items we purchased, all are still in use in our lab except for those that have been supplanted by newer items. I think this is because we were not just out to spend money, but we had teachers and coordinators choose software that they were interested in, and which they could integrate into their courses. The lab staff provides training for students and that relieves teachers of the responsibility to have to be intimately familiar with the software. This system has worked extremely well. In a nutshell, we in the lab have been guides and helpers, but teachers have made the decisions.
Respondents mentioned the following problems with their software:

1) We saw this software at TESOL last year and heard great things on the Internet. The demo didn’t show the real problems with it. When restarting the program you cannot pick up where you left off. You have to start again from the beginning.

2) The company’s tech support has been spotty.

3) The program itself is too big to fit on our hard drives.

4) Not enough copies.

5) PC version only and we are a MAC lab!

6) The only limit to the use of this software is that people keep pilfering the CD-ROMs.

7) It was supposed to be networkable, but has been very buggy.

8) Need for good typing skills and knowledge of possible problems (tying a space at the wrong place) made it difficult for NNS without keyboarding experience at this low level.

9) Install is painfully slow!

All of us can sympathize with the colleagues who reported these problems because we have experienced similar ones ourselves. These problems occurred even though the software in each instance went through a selection process and was evaluated in some cases by more than one of the methods described above. Even so, serious problems appeared after purchase. (It should be mentioned that the software is still being used to some extent in seven out of the above nine cases.) Is there some way that these problems could have been foreseen? In (1) above, three sources of information (demonstration at a conference, information on the Internet, and evaluation of a demo) still failed to prevent purchase of unsuitable courseware. The specific problem here was the annoying design feature that did not allow users to begin where they left off at their previous session. This problem is typical of those small, but important, points which become obvious once the software is used on a regular basis but which may not be mentioned in any source of information or even show up during trial use. This particular problem is specifically mentioned in some checklists: “a user can begin at a point appropriate to past use or experience” (Sloane et al., 1989). Unfortunately, many checklists, while listing several aspects of “interface navigation” or “learner control,” do not remind the teacher specifically to check for this feature. In (2) above, the problem was poor technical support. Checklists and rubrics often remind us about this problem (e.g., Vaille, 1998), but it is not something that is easy to check before purchase.
Responses (3), (5), (7), and (9) deal with technical and hardware issues. Many checklists remind us to check the operating system, memory requirements, network compatibility, etc. (e.g., Bradin, 1999; Sloane et al., 1989). Checklists are less helpful about (9), the time consumed in installing or loading programs. Some, like Vaille (1998), mention “access speed” but not installation or loading times. Squires and McDougall (1994) warn purchasers to look for delays caused by graphics or excessive disk access and stipulate a reasonable start-up time but do not mention installation time.

This brief survey of the problems mentioned by teachers in the questionnaire suggests that there is no perfect method of software selection. The vendor who explains everything, the colleague who remembers everything, the checklist that covers everything, and the framework that suggests everything do not exist and, if they did, would probably be impossible to use. Consequently, software selection is still very much an art honed by experience. We should mention finally that the problems listed in (4) and (6) above are evidence that those teachers chose good software.

**CONCLUSION**

While the number of responses to our survey were too few to be able to draw statistically valid conclusions, the results indicate that purchasers value recommendations from an impartial source over other methods of software selection. A high proportion of the respondents did try the software themselves or had their students try it as part of their decision process. It seems that only about 25% of the purchasers used any sort of a checklist to evaluate the software, even though this method would have been a useful way to discover inadequacies that might otherwise have not been evident. We hope that a larger scale study in the future will confirm our tentative findings and suggest more reliable methods of courseware selection.

**NOTE**

1 Strictly speaking, “selection” should be distinguished from “evaluation” (Squires & McDougall, 1994), but we follow the common practice in the literature of using the terms synonymously.
ACKNOWLEDGEMENTS

This article is based on a presentation given at CALLing Asia (Kyoto, Japan, May 21-24, 1999). We would like to thank those teachers who responded to the survey and those who attended our presentation and offered valuable suggestions. Our special thanks go to Professor Philip Hubbard of Stanford University for his insightful comments on an earlier draft of this paper.

REFERENCES


The Life and Death of Software

AUTHORS’ BIODATA

Thomas Robb is a professor in the Faculty of Foreign Languages, Kyoto Sangyo University, Japan. He manages two computer labs for the faculty and has published a number of articles on uses of the Internet for language learning. He is Chair-Elect of the TESOL CALL Interest Section.

Bernard Susser is a professor in the English Department of Doshisha Women’s Junior College, Kyoto, Japan. He has published on computers and writing, CALL, and cross-cultural communication.

AUTHORS’ ADDRESSES

Thomas Robb
Faculty of Foreign Languages
Kyoto Sangyo University
Kamigamo-Motoyama
Kita-ku, Kyoto 603-8555
Japan
Phone: +81 72-845-1874
Fax: +81 72-844-7303
E-mail: trobb@cc.kyoto-su.ac.jp

Bernard Susser
English Department
Doshisha Women’s Junior College
Kodo, Kyotanabe City, Kyoto Pref. 610-0395
Japan
Phone: +81-75-231-7267
Fax: +81-75-231-1738
E-mail: bsusser@dwc.doshisha.ac.jp