Evaluation Criteria for ESL/EFL Software

Eiji Hashimoto
University of British Columbia

Thousands of software programs exist on the ESL/EFL language learning and teaching market. However, systematic evaluation criteria have not been developed to assist teachers in making independent evaluations. In fact, the software evaluations that have been conducted have created confusion and have often mislead language educators. This paper reviews the literature related to ESL/EFL computer software evaluation and suggests avenues for the development of reliable criteria.

外国語あるいは第二言語としての英語教育のためのソフトウェアは数限りなくある。しかしながら、教師が独自にそれらを評価するための体系的な評価基準は、まだ開発されていない。実際、これまでに行われた評価は言語教師を混乱させ、誤った方向に導いてきた。本稿では、外国語あるいは第二言語としての英語教育のためのソフトウェアの評価を扱った文献を紹介し、信頼性の高い基準を開発するための提案を行う。

Many nations look to English as the language of science, technology, and world commerce and their schools and teachers have consequently looked to CALL (Computer Assisted Language Learning) to help meet pedagogical needs. Consequently, the development of CALL software has gained increasing attention, from both language educators and publishers. Stolurow and Cubillos (1983) counted more than 500 ESL/EFL software packages available. However, the original promise of CALL has diminished and results from the application of CALL in ESL/EFL programs have been disappointing (Hubbard, 1987). In many cases, the failure of CALL can be attributed to a lack of software evaluation criteria. Educators have no generally accepted standards by which to judge CALL software.

Background English instruction has involved many different methods, procedures and approaches during the past 100 years. The Audiolingual Method, for instance, had an impact on English instruction across North America in the 1960's (Larsen-Freeman, 1986). Most school administrators appeared to believe that modern technology, along with audiolingualism, would revolutionize language teaching and learning and rushed to install language labs, emphasizing memorization, repetition and mimicking in their schools. Although the impact was enormous, language labs did not significantly improve language learning (Davis, 1982). The 1980's saw the revolutionary development of a new technology, the computer. Again it was thought that this new technology would have positive effects on the teaching and learning of English. Instead of focusing on language laboratories, ESL educators began to see CALL as a new generation of technology superior to the "old fashioned" technology of language labs (Pederson, 1987), one which would result in improved language teaching and learning.

During its early development, the content, format, and compatibility of CALL programs were diverse. This diversity created confusion, especially among teachers and administrators who wondered what software was best for their students and how they could use what was then new educational technology. To make the situation more complex, publishers competed to dominate the software market. This was reflected in an indifferent quality of software. Johnson (1985) claimed that software developers produced low quality software packages to attract consumers irrespective of whether they were good educational programs. One early and consistent problem was that the quality of software had not been systematically examined, nor was there any agreement on evaluation criteria among those involved. One major problem was that teachers had no way of knowing the potential of the software package until they had actually tested it on students. Even in the 1990's, there is no agreed-upon set of evaluation criteria that teachers can use to assess CALL programs. In addition, several problems have been identified in the literature that makes it difficult to evaluate CALL software.

Hubbard (1987) claimed that five factors make the evaluation of CALL programs difficult and challenging. First, software evaluators are generally newly computer literate; their knowledge, therefore, is shallow and limited. In addition, they come from a variety of backgrounds, including language teaching, publishing, distributing, and computer programming. Publishers and distributors tend to overlook the students' perspective of language learning not only because they are not language specialists but also because they seem to have a difficult time obtaining reliable feedback from students, teachers, or administrators. Teachers and administrators understand the needs and abilities of their students, but have limited experience and knowledge about computer software. An essential question is, therefore, who should evaluate CALL software.

A second problem is that there is no way to skim software, which makes it difficult to evaluate programs. Program evaluation takes a great amount of time. Unlike the evaluators of textbooks and other printed materials, program evaluators must use relatively inflexible, virtually lockstep procedures to check the pedagogical value of the software. This restriction makes it difficult and complex to examine the format and content of the software in the time possible with textbooks.

The third problem is related to which software to use, and where and how to use it during a lesson. Pusak (1987) notes that one can use different software according to students' learning styles, teaching content, and learning targets. Another option is to use software related to the textbook being used. In other words, every software program has different pedagogical objectives, formats, and content. How, then, can evaluators establish common criteria to evaluate different types of software?

Fourth, there is a problem in the visual and auditory dimensions of computer software; the question is whether graphics, fancy colors, buzzes, beeps, or electronic melodies actually enhance or detract from the lesson. Many software authors and publishers believe these visual and auditory factors motivate learners and are the key to increased salability. However, Garrett and Hart claim that "graphics can be a powerful lesson feature, but in many cases they are merely added on and do not improve the content of the lesson or the student's interaction with it" (1985, p. 60). In addition, the visual and auditory dimensions of computer software are the most difficult to evaluate because learner preferences and learning styles are involved (Hubbard, 1992; Pederson, 1987). For example, graphics type "A" might help student "A" but not student "B". Does this mean that graphics "A" is bad? How can evaluation criteria be set?

Finally, the interactional aspect in software needs to be carefully reviewed. In other words, to what extent do students control the computer lesson and vice-versa? How does the software respond after making an evaluation? How intelligently does the software program evaluate students? In most cases, students deal with multiple-choice questions or simply press buttons when learning from computers. It is very difficult to assess whether they are really learning or not. How does the software evaluator determine how much interaction and what kind of interaction is necessary to improve the student's language proficiency (Garrett & Hart, 1985)? To address these concerns there have been some efforts to evaluate CALL programs.

Assessment Checklists

There are a number of published software evaluation lists. The CALICO Journal, *CALL-IS Newsletter*, and *CALL Digest* all include reviews of ESL/EFL software. The International Council for Computers in Education designed the MicroSIFT form to evaluate software from any field of study for use with CAI (Computer-Assisted Instruction) (Johnson, 1985). There are also published software evaluation lists developed by individuals. Although the formats and questions differ from one source to the next, there appear to be four characteristics among the published evaluation lists.

First, the evaluation sheet consists of a check list where the evaluators are asked to circle the number or item which best reflects the their judgment on a particular question. This is usually followed by a summary sheet for personal comments and overall assessments of the software. Second, questions are divided into 5 to 10 categories, including:

1) Content, Support material, Presentation, Stimulation of students' interest, Computer techniques (Cornick, 1984);
2) Instructional purposes and techniques, Instructional characteristics, Content characteristics, Technical characteristics, Program quality summary (Milley, 1985);
3) Interactivity intelligence, Human factoring, Documentation (Garrett & Hart, 1985);
4) Content, Approach, Design, Delivery (Pusak, 1987); and
5) Product information, Instructional design, Content, Summative evaluation (Yuen, 1989).

Many different terms are used under each of the categories, with considerable overlap. From a technical point of view, it seems esential to reach some consensus on what should be evaluated for published software. It is suggested here that questions concerning the technical aspect of problems can be classified into five categories.

First, evaluation sheets should include questions about teaching objectives and the skill area(s) that the software aims to cover. Second, target students should be defined and their language proficiency levels listed. Third, evaluations should have questions about hardware requirements. Fourth, the content of the software should be stated on the evaluation sheet. Finally, there should be one overall quality rating involving how all of these factors fit together to make a final product. The technical aspects of evaluation criteria seem very straightforward and questions can be rather easily structured. However, the pedagogical aspects of evaluation are more complicated and need clear definitions of the concepts of theories, methods, and approaches involved.

Another characteristic of published evaluation forms is that they lack methodological evaluation criteria. Miller and Burnett argue that "software evaluation criteria focus mainly on technical rather than on learning and educational issues" (1986, p. 159). This is a very important concept because once a particular language teaching theory or approach is set as a criterion for evaluation, "it automatically establishes criteria of the overall lesson structure and the role that graphics, sound, screen layout, etc. will play" (Hubbard, 1987, p. 251). Consequently, a single piece of software might require more than one evaluation depending upon what theory, method, or approach is embedded in the criteria. An outstanding CALL program designed within an audiolingual approach, for instance, may be a poor communicative competence software program. Most currently used evaluation sheets are designed to evaluate software without establishing the fundamental pedagogical criterion.

However, the idea of adopting a pedagogical perspective as a basic evaluation criterion has some shortcomings. For example, a number of theories and approaches exist in the field of TESL. Concepts and findings from recent research continue reshaping present theories and approaches. Therefore, relationships between theories and approaches are often unclear and their definitions are still arguable (Hubbard, 1987).

For example, Miller and Burnett claim that "from a holistic viewpoint, even the separation of reading from writing in instruction creates an artificial dichotomy" (1986, p.116). Does this mean that from a holistic teaching point of view, the software will be marked low if it does not combine writing exercises with reading exercises? Miller and Burnett continue that "enlightened subskillists, who advocate the isolation of skills for instructional purposes, recognize the need to create realistic situations for their application" (1986, p. 161). Does this mean that skills-based software can never receive a good evaluation even if both teachers and students react positively? Miller and Burnett (1986) conclude that a two-level hierarchy for evaluating software will solve this problem: the first level provides a theoretical orientation, and the second level focuses on various technical issues. Under the theoretical umbrella, technical concerns are evaluated. This hierarchical model can avoid confusion and inconsistency in the assessment of software.

While Miller and Burnett's two-level hierarchy model is theory based, Hubbard (1987) suggests three categories of approach as fundamental pedagogical criteria: Behaviorist, Explicit learning, and Acquisition. He claims that "these three categories reflect useful distinctions for materials development and CALL software evaluation, since they reflect major components of specific theories and models of second-language acquisition" (p. 231).

However, this categorization has a shortcoming, that is, the nature of CALL. The major criticism of CALL is the over-use of the behaviorist stimulus-response language learning theory. Unfortunately, there is a great deal of similarity between the Audiolingual Method and CALL. The following is a list of principles of audiolingualism noted by Larsen-Freeman (1986):

1. presents vocabulary and structure appropriate to the learner's level;
2. maintains the learner's attention to task;
3. requires the learner to input the correct answer before proceeding;
4. provides the learner with positive feedback for correct answers;
5. provides sufficient material for mastery and overlearning to occur;
6. reinforces patterns and vocabulary presented in a lesson; and
7. presents grammar rules or patterns inductively with no attempt at teaching explicit formulations of them. (Hubbard, 1987, p. 231)

This list of principles is very similar to the principles in most current CALL software programs. Strictly speaking, it may be that the newer approaches to language teaching that involve learning strategies and acquisition can never be adequately programmed into CALL because the basic CALL format is behavioristic in nature. The basic program involves stimulus-response. Can language competence be written into such a program?

Even though providing a theoretical orientation with evaluation criteria appears the only way to solve the inconsistency of software evaluation, it calls forth two important questions: 1) What theoretical aspect should be adopted as a base? 2) Can the theories, approaches, and language learning and teaching principles derived from mainstream ESL/EFL research apply to CALL?

Finally, most of the evaluation criteria do not include a category of "learner strategy." Hubbard (1987) claims that the basic idea of employing learner strategy as one evaluation criterion is to judge the effectiveness of software. Learner strategy is different from teaching strategy. Here is a good example of learner strategy: "in teaching vocabulary with a learner-strategy orientation, the focus is not on learning individual lexical items; instead, the teacher introduces and provides meaningful practice in strategies for guessing the meaning of an unknown word" (Hubbard, 1987, p. 238). Learner strategy plays a role in taking an inside look at software: it measures whether the software is really effective for learners rather than the pedagogical significance the software has relative to a particular theory or approach. Although the advantage of the learner-strategy paradigm is that it "is not limited directly to any particular category of approaches" (Hubbard, 1987, p. 237), it seems questionable whether any present software can be evaluated for learner strategies until it is tested with them. Hubbard (1987) argued that as skill-oriented and strategy-oriented software featuring sophisticated technology appears, this new category will be necessary to determine the potential effectiveness of the software. However, another question of importance is whether we should consider other learner factors such as learning style, motivational factors, and learner ethnicity as future evaluation criteria (Hubbard, 1992; Pederson, 1987). Future development of CALL software needs to take these into account.

CALL is a relatively new area with software development apparently still in a growing stage and an evaluation system which has not been well shaped and organized. Johnson (1985) stated that:

> "... a similar phenomenon occurred when language proficiency tests first came into widespread use. Thick catalogues appeared which were useless in helping to make choices. Gradually research firms, states, and publishers all played a role in narrowing down the field and eliminating the worst tests." (p. 14)

CALL is in a similar state of confusion. It is difficult to make informed choices concerning computer software because there is so much and the information provided is complex and difficult to understand. Members of different communities, publishers, educational organizations, individuals, and journals such as CALICO Journal and the CALL-IS Newsletter must work together to develop evaluation criteria that are narrow and informative. Without such an effort CALL programs will continue to be difficult to assess and evaluate.

Summary and Conclusions

English educators around the world have looked to the computer as one answer to the problems they face in providing programs designed for students from diverse backgrounds and abilities. CALL has been viewed as one answer. However, a basic problem for teachers is to select software appropriate for their students and programs. This is no easy task since there are no agreed upon criteria to evaluate software. Based on the review of research related to software evaluation presented, it is suggested that further research on CALL address the following issues:

1) What criteria do users of CALL programs consider important in evaluations?
2) What aspects of CALL programs do users believe make them good?
3) What features of CALL programs do users find make them poor?
4) What features of CALL programs do students like best?
5) What special features of CALL programs should be included in an evaluation form?
6) What physical features, such as format and price, should be included in an evaluation?
7) Should a program's theoretical format be evaluated? If so, how?

Many educators have turned to the computer and to CALL to help students learn English, however, serious problems occur because there are no agreed-upon criteria to evaluate software. Educators currently make serious judgments about CALL software without a resource to guide them. The establishment of valid and reliable criteria is essential for the future use of CALL in ESL/EFL education.

Eiji Hashimoto, M.A. TESL, St. Michael's College, is a candidate in the Ph.D. program in the Curriculum and Instruction at the University of British Columbia. He is interested in CALL and classroom management.

References

Cornick, L. (1984). Evaluating foreign-language CAI: A checklist. *Unterrichtsprazis*, 17(2), 315-318.

Davis, N. (1982). Foreign/second language education and technology in the future. *NALLD Journal*, 16(3), 5-13.

Garrett, N., & Hart, S. (1985). Foreign language teaching and the computer. *Foreign Language Annals*, 18(1), 59-63.

Hubbard, P. (1987). Language teaching approaches, the evaluation of CALL software, and design implications. In W. Smith (Ed.), *Modern media in foreign language education: Theory and implementation* (pp. 227-252). Lincolnwood, NE: National Textbook Company.

Hubbard, P. (1992). A methodological framework for CALL courseware development. In M. Pennington & V. Stevens (Eds.), *Computers in applied linguistics: international perspective* (pp. 39-65). Clevedon: Multilingual Matters.

Johnson, D. M. (1985). *Using computers to promote the development of English as a second language.* New York: Carnegie Corp. (ERIC Document Reproduction Service No. ED 278 211).

Larsen-Freeman, D. (1986). *Techniques and principles in language teaching.* Oxford: Oxford University Press.

Miller, L., & Burnett, D. (1986). Theoretical considerations in selecting language arts software. *Computers and Education,* 10(1), 159-165.

Milley, R. (1985). *Software reviews for adult education.* Chelmsford, MA: Merrimack Education Center. (ERIC Document Reproduction Service No. ED 268 342).

Pederson, K. (1987). Research in CALL. In W. Smith (Ed.), *Modern media in foreign language education: Theory and implementation* (pp. 99-131). Lincolnwood, NE: National Textbook Company.

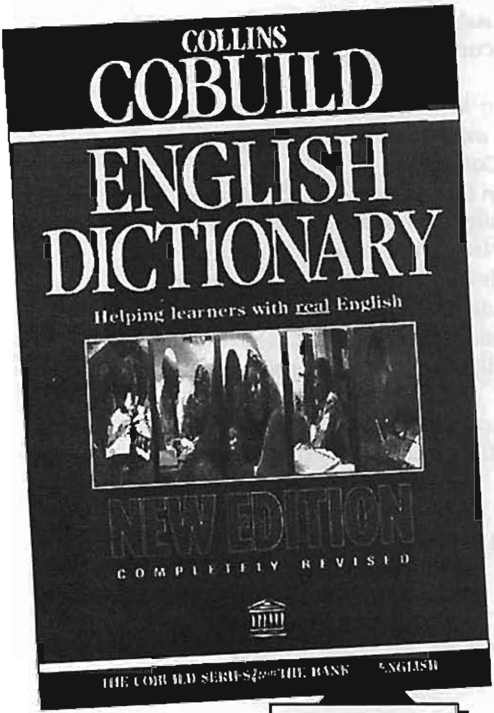
Pusak, J. (1987). Problems and prospects in foreign language computing. In W. Smith (Ed.), *Modern media in foreign language education: Theory and implementation* (pp. 13-39). Lincolnwood: National Textbook Company.

Stolurow, L., & Cublillos, E. (1983). *Needs and development opportunities for educational software for foreign language instruction in schools.* Iowa City, IA: Center for Education Experimentation, Development, and Evaluation, University of Southern Mississippi. (ERIC Document Reproduction Service No. ED 242 204).

Yuen, S. (1989). *Computer assisted instruction: A handbook for ESL teachers.* University of Southern Mississippi. (ERIC Document Reproduction Service No. ED 317 044).

(Received July 26, 1994; January 12, 1995)

ANNOUNCING...


COLLINS
COBUILD
ENGLISH
DICTIONARY
Helping learners with real English
NEW EDITION
COMPLETELY REVISED
THE COBUILD SERIES from THE BANK OF ENGLISH

NEW

COLLINS COBUILD ENGLISH DICTIONARY

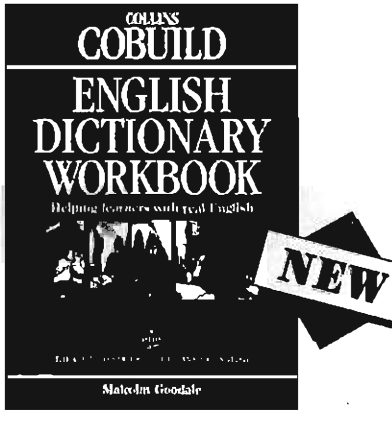
- Based on evidence of written and spoken English of the 1990's
- Over 100,000 NEW examples taken from the 200 million word Bank of English
- Special treatment of core vocabulary
- Streamlined grammar coverage in the unique extra column
- Revised and improved pronunciation guid
- Includes American and British English
- Full sentence explanations in natural English

Also coming

COLLINS COBUILD ENGLISH DICTIONARY WORKBOOK

by Malcolm Goodale

Fun and practical exercises to help learners get the most from their dictionary



For more information or samples contact:

David Gray
HarperCollins Japan
1-2-1 Sarugakucho, Chiyoda-ku, Tokyo 101
Tel: (81) 3-3291-6343 Fax: (81) 3-3294-8284

For orders, please contact our
exclusive importer and stockist:
KINOKUNIYA COMPANY LTD.
38-1, Sakuragaoka 5-chome
Setagaya-ku, Tokyo 156
Tel: (81) 3-3439-0161 Fax: (81) 3-3439-0839