CALL and a Framework for Software Evaluation*

Yukie Aihara

CALL (Computer-Assisted Language Learning) is a part of CAI (Computer-Assisted Instruction), which is the term for the use of a computer as an instructional tool. The most important element in CALL is software. The urgent need to establish a framework for software evaluation and to exchange information is claimed both by the Ministry of Education and by in-service language teachers.[1] This article discusses some features of CALL software which differentiates it from other instructional devices and eventually forms a framework for a CALL software evaluation checklist.

Introduction

A recent trend which has caught the imagination of language educationalists and administrators is Computer-Assisted Language Learning (CALL). Some language teachers have already plunged into this computer revolution and have been enthusiastically exploiting this high-tech equipment.[2] Some might be pessimistic and say with a sigh that

* I would like to express my gratitude to Dr. Michael Carr for his careful review of the manuscript.

1) 日本視聴覚教材センター (Nihon Shichoukaku Kyouzai senta) 1985:168, Mizumachi 1984:17.

2) Mizumachi (1985) reported the achievements of the TELP (Tokai English Language Program)－CAI project, which developed a stand-alone microcomputer system with abundant courseware. The latest system called TELP-CAI system III is reported to have hint messages and feedback from error correction as well as to accept no less than three variant answers.

CAI/CALL would be left in the corner of a classroom and would be gathering dust as the language laboratory did in their 60s (Kita 1985). Some could not accept CAI/CALL because they see it as a threat to their job security or because they believe language teaching should not be done by such an inhuman device as a computer. Concerning this kind of rejection, Gerhold (1980: 17) indicates "in most cases it is a general resistance to educational innovation." There is another group of language teachers who are increasingly interested in the use of computers as an aid, but are not quite certain about CALL and its usefulness. The object of this article is to provide an overall view on using CALL to aid in language learning: what a computer can do, what kinds of software are available now, and what kinds of software are needed. Finally an evaluation checklist is submitted for language teachers' use to exchange information on prevailing CALL software and become knowledgeable enough to make a decision on whether or how to use a computer in their profession.

I. Where we are now

Computers[3] are now swiftly surging throughout Japan and there is no doubt that these revolutionary high-tech devices are soon to be everywhere in daily life. The Ministry of Education Survey in 1983 on the use of computers in Japanese schools and governmental facilities in education shows that 56.4% of high schools, 3.1% of junior high schools, and 0.6% of elementary schools possessed more than one computer respectively. The survey also reports that high schools have an average of 4.2 computers, and junior high schools 1.4. The number of computers

3) 'Computers' in this article indicates 'microcomputers' or 'personal computers which seem to spread widely for education near future.

at the time of the survey was small, but it must have doubled or tripled by now. In 1985 the Ministry of Education budgeted two and a half billion yen for bringing computers into classrooms (Kita 1985, Saeki 1986). In May, 1985, the Ministry of Education submitted a report on the use of computers in education in order to cope with this new trend of technology.The report gave comprehensive guidance on the use of computers in school education as well as in social education in general. It also provided overall guidelines to bring computers into education.[4] Saeki (1986) describes the year 1985 as the opening of the compter age in Japanese education.

As far as software is concerned, the Octorber 7, 1984 *Asahi-Shimbun* reported that approximately 15,000 kinds of computer software were available in the market. Although the amount of educational software was increasing, the problem was that most educational software was developed by programmers without any consultation with educationalists and/or specialists. As a result, bad software has overwhelmed the market. The article also reported that in September, 1984, the Society of CAI in Japan[5] published a users' checklist with about one hundred twenty items. In December, 1985, the Ministry of Education also publicized a guideline for developing educational software. All these guidelines seem to discuss such general items that little detailed information can be obtained about our specific interest, CALL.

In accordance with Miller's (1984) five evolutionary stages of computer use, at present Japanese schools seem to be in the stage two, where the emphasis is on purchasing equipment. Soon those schools will shift to the stage three: searching for software. Language teachers have to

4) 日本視聴覚教材センター (Nihon Shichokaku Kyouzai senta) 1985:149-176.
5) ＣＡＩ学会，神奈川県平塚市北金目1117，東海大学電子計算センター内。

cope with this phenomenon. They have to consider how CALL fits in with classroom teaching and what kinds of software they need.

II. Terminology on CALL

In case the reader is a novice at CAI/CALL, it might be helpful to explain some computer related terms.

CAI: Computer-Assisted (-Aided) Instruction

CBI: Computer-Based Instruction

CAL: Computer-Assisted Learning

The terminology for the use of the computer as an instructional tool seems to need more time to be standerdized. However, the fundamental idea of these terms is almost the same. In Japan, CAI is most frequently used. There are two terms for language teaching:

CALI: Computer-Assisted Language Instruction

CALL: Computer-Assisted Language Learning

Underwood (1984:38) expresses his preference for using CALL because at present most of the CAI/CALL systems are used for drill and practice or review, and actual instruction seldom occurs. So, the term CALL, implying 'learning,' is more suitable at present. These additional terms might be helpful, too.

Hardware: physical components of a computer, such as the computer itself, a display terminal, a key board and a printer.

Software: the programed commands that are understood by the computer. Basically the computer is blank. It needs commands in order to work for its user's need and usually those commands are saved in a floppy diskette.

Courseware: a set of instructional lessons that can operate on the computer.

III. What computers can do for language learning

When a new technological device is introduced, the first thing to do is to exploit its potential use. If we use a new device in the same way as we use old ones, our investment results in only a superficial change of classroom teaching, or just satisfying administrators' or teachers' pride that they are newly equipped.

What can a computer do for language teaching development? What are the inherent features of a computer which differ itself from other devices, such as a textbook, a workbook, flashcards, an OHP, and a language lab? We have to grasp those features in order to most effectively utilize a computer in our teaching.

A large memory, record keeping and calculation functions are useful features of a computer as a learning tool. In addition, in considering CALL software which actually directs a computer to contribute to language classes, these four features are manifested: feedback, pacing, branching, and novelty or creativity.

Feedback is characterized by the CALL software's capabilites of analyzing what a learner does, characterizing the learner's errors and giving appropriate advice. The learner needs explanations on why his[6] answer is wrong and help messages to figure out how to get the correct answer. Even if the answer is right, he needs to know that it is right and see that answer again for reinforcement. No other divices have this function, not even the language lab where the learner repeats after a tape, records his voice and compares his own response with the correct answer. In the lab the learner must correct by himself. There

6) Since most of my students are male, allow me to use male gender for the pronoun of 'learner.'

is no encouragement for right answers as well as no explanations for wrong ones.

Primitive feedback is neither informative nor supportive. It just tells the learner, "Your answer is wrong. Try again." There are no hints as to what the learner's problem is. Some advanced CALL programs have so-called "error anticipation," a list of possible wrong answers, and tries to use this diagnosis to provide hints as to the nature of the problem (Underwood 1984:47). The advantage of this feedback strategy is that the computer corrects the learner's answers without criticizing. The CALL program can read the learner's response and respond accordingly with a comment or an explanation, or by merely pointing out. The learner feels neither embarrassment nor anxiety by making mistakes (Stevents 1983:295). Such encouraging remarks as "Hats off," "You almost got it," and "There are signs of genius," are also helpful. They motivate the learner to continue the lesson and learn through trial and error.

In some respects, feedback is considered as an interaction between a computer and a learner. Through interaction with the computer, learners are motivated and gradually try to think by themselves and to learn on their own. The learner becomes active, rather than passive, at the computer. Underwood (1984:95) describes another kind of interaction. He reports that many educationalists have noticed dialogs occurring in front of the screen among the learners themselves rather than on it. There is a tendency that learners "band together to try to 'beat the machine'." Interaction among the learners elicits the exclusive use of the target language, and peer learning[7] arises.

---

7) A kind of learning in which a learner learns from another learner who has already learned something.

The second feature is pacing. Individual learners have their own pace in learning. Each learner should be allowed to progress at his own pace. In an ordinary class consisting of a teacher, learners, and textbooks, it is difficult for learners to maintain their own pace. They are sometimes forced to go fast to keep up with quick learners, and sometimes they have to slow down to wait for slow learners. In the lauguage lab, all the stimuli are usually prerecorded for the learners to respond. Thus, there is no chance for them to set their own pace except they stop the tape. On the other hand, learners in front of the computers can control the program and determine their own pace. In Imperial International Learning Corp's *Riddle? Me This*[8] the display is never changed unless the learner presses the return key. It is the learners, not the computer, that set the learning pace. Learners feel at ease when they notice that they control the computer, not the other way around. Better programs provide various directions for pacing: directions to stop the program, and those to allow learners to move back and forth in the program, skipping some part and coming back to it later.

The third feature is branching. The term 'branching' is used to describe the ability of a program to respond in different ways depending on the learner's input. A branching program allows learners to learn a language in their own ways. When an error is made, the computer reads and analyses it, then leads the learner to the most appropriate frame to solve the problem. In this way, the path of each learner through the same lesson is varied according to the learner's learning style.

Taken together, pacing and branching add up to individualization

8) The program was produced in 1983 and in 1984 it was selected as the Learning Computer Software Awards for Excellence in educational software for microcomputers by *Classroom Computing Learning*, in U. S. A.

which allows a learner to set his own pace and learning pattern. In an individualized CALL program, the learner can take as much time as he likes for thinking and keep his own learning rate.

The fourth feature is novelty and creativity. The CALL program can vary its response each time so that the learner feels novelty. The CALL program can produce responses or drills that it has not previously encountered. It can endlessly vary its outputs. In a quiz, the program shuffles the problems each time and puts them in a different order. In the testing part of *Gambarikun*,[9] for example, definitions of Japanese *Jukugo* (Chinese compound words) are asked in ten to one hundred questions with random selection.

The CALL program also can be creative when it adapts to the learner's input by itself. Consequently learners are provided with different responses from the computer each time. The author has seen a program for sentence making. When a learner types "a boy," a boy pops on the screen. Then, the learner types "jumps over a dog." A dog appears and the boy jumps over it. Though the number of words the learner was able to use was limited, the program enabled the learner to create quite a few sentences, and to see the objects he created jumping and running around on the screen. This program is motivating because it is fun to see the computer's immediate response to the learner's commands. In a sense, the computer takes a role of tutee who learns from the learner, not a tutor who drills the learner. This reversed situation increases the learner's motivation to learn through using the language.

So far, we have seen the four major inherent features of CALL programs. The most important thing to remember is that these potenti-

9) Authored by H. Eastwick-Field, 57-13 Kusakawa-cho, Nanzenji, Sakyo-ku, Kyoto 606 JAPAN.

alities of the computer should be fully utilized. If the computer is used just as an "electronic workbook" (Harrison 1983) or "page-turner" (Barrutia 1985), there is no reason to spend a large amount of money for CALL software. Harrison (1983:30), who published a software evaluation list with over 100 CALL programs in the U.S., regretted that the industry is "still in the state of infancy" and that the software available did not exploit the potential for CAI. [10] Language teachers should understand to potentialities and deficiencies of computers and involve themselves in the development of better software which utilizes all the advantages of computer capabilities. Software which does not exploit the computer's potential is more commonly available than better software. We should not make an early decision that CALL is inefficient only on the basis of using inadequate software.

IV. Activity types of CALL in relation to language teaching methodologies

How should all the above-mentioned potential CALL features be used? What kind of activities are generated and available? The activities in most CALL software are tutorial drills and practices, games, tests and simulations. CALL software is "methodologically neutral" (Fox 1985:96). It changes according to its author's imagination and intention for methodological application. In contrast, the language lab reflects audiolingualism based on stimulus-response, and is a device for practicing pattern drills. Because of its neutrality,

10) It may be caused by the limited capacity of hardware or the limited creativity of the program author. It is said more than one hundred hours is necessary to program and test a simple lesson (Stevens 1980, Cassidy 1983, *Asahi Shimbun* Jan. 22, 1986).

CALL software can also be produced easily with unimaginative and meaningless drills. Some of them are methodologically dubious. Among the software available in the market, two types are noticeable: Tutorial (or practice) type and simulation type. The former tends to get more criticism than the latter, and in turn the latter gets more admiration than the former.

Tutorial programs which offer knowledge by asking for certain responses, accepting right answers, and reacting to wrong answers, are usually in a drill or practice format. Such programs reflect behavioristic learning theory which prevailed during the audiolingual period. Emphasis is put on Skinnerian stimulus-response strategies and accounts for the early misuse and eventual demise of the language lab. This program emphasizes habits. Exercises are mechanical and usually focused on form, not on meaning. PLATO[11] lessons are an example of this tutorial drill which may well be effective for learning rather than language acquisition. PLATO lessons are useful with helpful feedback as supplementary to classroom instruction for those students with special problems or needing extra help. In the U. S. over 50% of the foreign language CAI software available in 1984 was of this tutorial type (Kossouth 1985). Much criticism is made on this type of program. Dalgish (1985a) claims that tutorial type program was "re-inventing the 60s" with language lab type materials and calls it an "electronic workbook." Once the novelty wears off, drill and practice actually prove to have negative effects like the language lab (Underwood 1984:94).

11) Developed by Intensive English Institute, Division of ESL, University of Illinois, Urbana, Illionis, and marketed by Control Data Corporation, P. O. Box 0, 8100, 34th Ave., South, Minneapolis, Minn. 55440 U. S. A.

If the software author is influenced by humanistic psychology,[12] his/her program will be simulation type which employs such computer's features as branching, enjoyable feedback, and creativity or novelty. Simulation offers the learner a real life experience on the computer in which he must constantly solve language problems in order to achieve some language goal. The most elaborate example is the system used to teach pilots how to fly. In *PHOTOFIT* (Higgins 1983), the learner's role is that of witness to a crime and his task is to help the police artist draw the face of the criminal which the learner has seen at the beginning of the program. The learner gives commands like "nose," "bigger," etc., and then compares when the drawing is finished. Higgins describes learners who have used modals, comparative adjectives and descriptive language without realizing how much English they have practiced and how many times they have voluntarily repeated those words. In the simulation program, the computer provides a "new and exciting way of increasing the learner's exposure to meaningful language" (Higgins 1983: 6). The program is aimed at meaningful communicative activities rather than mechanical drills.

Another frequently introduced example of simulation is so-called the "Adventure Game" (Underwood 1984:55-56, Kossouth 1985, Higgins 1983). Kossouth introduces a German language Adventure *Munich.*[13] In this program the learners are tourists visiting Munich and encouraged to make decisions about how to reach the place of interest, and when to

12) After reconsidering the cognitive or intellectual part of learning, humanistic psychology added a new light to the classroom. It is concerned with educating the whole person— the intellectual and the emotional dimensions. It puts stress on human interaction and self-actualization (Moskowits 1978:8-13).

13) Claremont Academic Software, 433 W. Harrison Ave., Claremont, CA 91711 U. S. A.

eat or sleep or change money. The learner types in single word commands, and it takes a learner on the average of ten tries before winning the first time. The combination of reading, interpreting, discussing and writing provides a rich integrated activity through the medium of the computer. Kossouth (1985) discusses the enjoyable context and friendly remarks which use the learner's name for praise or correction help lower the "Affective Filter", a mental block that prevents learners from fully utilizing the comprehensible input (Krashen 1985). She also points out that with the use of such Adventure Authoring system as *Adventure Writer*[14] and *Adventure Master*[15] with which teachers can easily produce a program which can provide the learner with ideal input, "$i+1$".[16]

Recent literature tends to criticize the tutorial practice and drill type of program and praise the simulation type. When the purpose of learning is stressed on learning forms, tutorial practice type is also useful with encouraging feedback, pacing, and branching. There are no perfect approaches, learning devices, or sofware at present that can be utilized for every stage of learners or for all teaching objectives. Language teachers should be knowledgeable about all types of CALL software and try to effectively use them to improve their teaching environment.

---

14) Codewriter Corp., 7848 N. Caldwell Ave., Niles, IL 60648 U. S. A.

15) CBS Software, 1 Fawcett Pl., Greenwich, CT 06836 U. S. A.

16) 'i' indicates our current level and '$i+1$' is the next level. The Input Hypothesis claims that humans acquire language in only one way—by receiving 'comprehensible input.' We progress by understanding input that contains structures at our next 'stage'—structures that are a bit beyond our current level of competence. We move from 'i' to '$i+1$' (Krashen 1985).

V. What we should do now

The effectiveness of CALL programs for classroom teaching depends upon their usage. In considering the classroom use, language teachers have to know what kind of programs are currently available. To study CALL software requires time and money. Usually software is not observed before it is purchased and it takes much time to evaluate a lesson. In order to save time and money, the exchange of information on CALL software sounds beneficial. For this information exchange, the author has developed a CALL software evaluation checklist to help language teachers decide whether they should use a program or not. The check list takes into consideration the three constituents in a CALL classroom: the students, the teacher and the software itself. Since the most important parts have already been discussed in the previous sections, only a few additional comments are made for the list items. The item letters and numbers below correspond with those in the evaluation checklist.

B. Software Information: This information provides an overall view on the software.

7. Exercise type: Mechanical type is just like pattern practice. Learners can go through the exercise even if they don't know the meaning of the sentences they are using. Meaningful exercise requires learners to understand the language. Communicative type doesn't have any prefabricated answer. The exercises are open-ended.

8. Sound educational value: This is one of the most important elements of software. Accurate and clear information is indispensable in a good teaching material. The appropriateness of correct answers should be checked. For the classroom use, we also check whether the instruction

is suitable for learners' understanding. The instruction of a good program is always challenging. It requires students to use their brains, to integrate information, to make dicisions, and to let them challenge the computer.

9. Ease of use: Clear directions prevent learners from getting lost in a program. Those directions, telling learners what to do next and what the computer is doing now, keep their attention and concentration on the screen instead of making them bored. Directions should be clear, simple, complete, easy to read and exclusively using the target language.

When learners don't know how to respond, they need easy access for help. Help messages prevent learners from being discouraged and frustrated while learners continue the program. Help messages also guard the program against panic learners, who don't know what to do, 'bombing' the program by pressing keys at random. Help messages also let them exit and re-enter the program at any time.

Documentation usually contains a table of contents, glossary, index and a summary of key commands. They are important resources for the language teacher to give a quick over view to the program. A well written manual has frequent illustrative examples and sample screens. It avoids technical terms or defines them for novices.

11. Student involvement: The degree of learner's input varies by letter, word, and sentence. In multiple-choice learner's input is the correct letter or number alone. The word itself must be typed. Too little input is the loss of an opportunity to practice production, but on the other hand too much input causes mistyping and dampens enthusiam.

12. Flexibility: Considering the high price of software, it is not effective to purchase a program that has one or two activities only for

limited learners. A good program provides teachers with an editing section for adding and deleting items in the program so that they can extend the program to their learner's need. The program should not only anticipate all possible misspellings and close answers but also accept them in case these factors are not so important for learning. Otherwise, learners' enthusiasm will decrease.

13. Creative use of graphics: Clear color and animation is a strength of the computer program. They are evaluated not only for their part in increasing learners' motivation, but also for its contribution to learners' concentration.

CALL SOFTWARE EVALUATION CHECKLIST

A. Titel & Ordering Information

Title:
Author(s):
Copyright Date:
Publisher:

Cost:

B. Software Information

1. Language: ___English ___French ___Spanish
   ___German ___Russian ___Chinese
   ___Japanese ___Others
2. Level: ___beginner ___intermediate ___advanced
3. School year: ___elementary ___junior-high
   ___senior-high ___university
4. Objectives: ___grammer ___translation
   ___vocabulary ___idioms

____culture ____conversation
____writing skill ____reading comprehension

Other comments: _____

_____

5. Activity type: ____tutorial drill and practice
____game
____quiz/test
____simulation
6. Focus on: ____meaning ____form
7. Exercise type: ____mechanical ____meaningful
____communicative
8. Sound educational value:
accurate instruction: ____Y ____N
clear instruction: ____Y ____N
challenging instruction: ____Y ____N
9. Ease of use:
clear direction: ____Y ____N
target language in directions: ____Y ____N
help for getting stuck: ____Y ____N
individualized pace: ____Y ____N
user controlled pace: ____Y ____N
branching: ____Y ____N
return to start: ____Y ____N
exit anytime: ____Y ____N
documentation available: ____Y ____N
clear manual: ____Y ____N
worksheets: ____Y ____N
10. Feedback:
user friendly responses: ____Y ____N
more than one response available: ____Y ____N
encouraging remarks: ____Y ____N
hints provided: ____Y ____N
11. Student involvement:
interactive program: ____Y ____N

learner's attitude: ____active ____passive
degree of learner's input: ____letter (multiple choce)
____word
____sentence

12. Flexibility:
modifiable content: ____Y ____N
reusable content: ____Y ____N
misspelling accepted: ____Y ____N
'close' answer accepted: ____Y ____N
13. Creative use of graphics: ____color ____sound
____graphics ____animation
14. Time:
How long would it take an average learner to complete a lesson?
____min.

C. Hardware information

required hardware:
memory:
diskette: ____5"/2HD ____5"/2DD
____8"/2 D

Comments: _____

_____

VI. Conclusion

Seeing is believing. If you want to know about CALL, it is suggested that you try out one or two CALL programs by yourself. Then, you will get a clear idea what it is like: how it works in language learning; how the computer's potential features of feedback, pacing, branching, and novelty or creativity, reinforce the program; how it is exciting. Some of you might be stimulated to use CALL in your classroom after your first trial with the computer.

Reflecting on the language lab failure, it has been said that software should take the priority in using CALL. A recent trend fosters combining CALL with communicative teaching "concentrating on genuine exchange of information, on games and other self-rewarding activities, and on lively simulations of real and interesting encounters." [17] Simulation type software has obtained more attention than tutorial practice or drill type. The combination of videodisc and computer promises a bright future for language teaching. We still have to wait for a few years to see artificial inteligence make its debut in our field.

In this technological age, there is one thing we have to keep in mind. Language teachers must not be overwhelmed by this new technology, but must become knowledgeable enough to integrate their ideas and experience for CALL and to keep stimulating software programmers. As Underwood (1984:xv) says the language teachers "must take the lead" in developing CALL software. Computers will not go away. We should positively involve ourselves in CALL since "there is clearly an opportunity, a chance to improve language teaching, raise the proficiency level of our students" (Underwood 1984:96).

17) From 'Forward' by Earl W. Stevick for *Linguistics Computers and the Language Teacher: A Communicative Approach.* John Underwood. Rowley, Mass.: Newbury House Publishers, Inc.

REFERENCE

Ahmad, Khurshid, Greville Corbett, and Margaret Rogers. 1985. Using Computers with Advanced Language Learners: An Example. *The Language Teacher* 9 (3):4-6.

Barrutia, Richard. 1985. Communicative CALL with Artificial Intelligence: Some Desiderate. *CALICO Journal* 3 (1):37-42.

Canale, Michael, and Graham Barker. 1986. How Creative Language Teachers Are Using Microcomputer. *TESOL Newsletter* 20 (1) Supplement No. 3:1-3.

Cassidy, Joan E. 1983. Using Computers to Teach Foreign Languages to Adults. *CALICO Journal* 1 (2):14-16.

Cook, V. J. 1985. Bridging the Gap between Computers and Language Teaching. In *Computers in English Language Teaching*, Christopher Brumfit et al. (Eds.), 13-24. Oxford: Pergamon Press.

Cunningsworth, Alan. 1984. *Evaluating and Selecting EFL Teaching Materials*. London: Heinemann Educational Books.

Dalgish, Gerard M. 1985 a. Current ESL Software. *TESOL Newsletter* 19 (1):14.

————————. 1985 b. Current ESL Software: Branching and Error Correction. *TESOL Newsletter* 19 (2):23.

————————. 1985 c. Current ESL Software: Student Control/ Teacher Contorol. *TESOL Newsletter* 19 (3):9.

Dutra, Irene. 1986. About Supplement. *TESOL Newsletter* 20 (1) Supplement No. 3:1.

Fox, Jeremy. 1985. Humanistic CALL: Teacher's Needs and Learners' Need in the Design and Use of Computer-Assisted Language Learning Systems. In *Computer in English Language Teaching*, Christopher Brumfit et al. (Eds.), 91-97. Oxford: Pergamon Press.

「学習ソフト一部に粗悪品」.『朝日新聞』(Asahi Shimbun) Octorber 7, 1984:1.

Gerhold, George. 1980. Teacher-Produced CAI. In *Computer Assisted Learning Scope, Progress and Limits*. R. Lewis, and E. D. Tagg. (Eds.), 15-25. Amsterdam: North-Holland Publishing Company.

Hamerstrom, Helen, Gladys Lipton, and Sherry Suter. 1985. Computers in the Foreign Language Classroom: No Longer a Question. *CALICO Journal* 3 (1):19-21.

Harrison, John. 1983. Foreign Language Computer Software: What? Where? How Good? *NEC Newslettr* Feb:26-30.

Higgins, John. 1983. Can Computers Teach? *CALICO Journal* 1 (2):4-6.

Kita, Hiroshi (北 弘志). 1985. 「英語教育とコンピューター：2度目のＣＡＩ時代を迎えて」, *LLA Computer Section News* 1:6.

「コンピューターと子供たち」. broadcasted in "ETV 8" by NHK on December 11-12, 1985.

Kossouth, Karen C. 1985. Using the Adventure Formats for CALL. *CALICO Journal* 3 (2):13-18.

Krashen, Stephen. 1985. *The Input Hypothesis: Issues and Implications.* London: Longman.

「教育用ソフトウエア開発指針まとまる」.『朝日新聞』 December 12, 1985: 3.

Last, Rex. 1984. *Language Teaching and the Microcomputer.* Oxford: Basil Blackwell Limited.

Marty, Fernand. 1984. Computer Aided Instruction: Language Teachers and the Man of the Year. In *Initiatives in Communicative Language Teaching*, Sandra J. Savignon, and Margic S. Berns (Eds.), 155-165. Reading, Massachusetts: Addison-Wesley Publishing Company.

Miller, Inabeth. 1984. How Schools Become Computer Literate. In *Guide to Computers in Education* Special Issue: 22-28.

Mizumachi, Isao (水町伊佐男). 1984.「ＣＡＩに関する米国二団体の活動の紹介」.『名古屋学院大学外国語教育紀要』 11:15-17.

————————. 1985. One Approach to CAI-EFL Problems in Japan. In *CALICO Journal* 3 (1):7-10.

Moskowitz, Gertrude. 1978. *Caring and Sharing in the Foreign Language Class.* Rowley, Massachusetts: Newbury House Publishers, Inc.

Myers, Erick V. 1985. The Development of Computer Interactive Simulations in Language Teaching. *The Language Teacher* 9 (3):11-12.

Nihon Nouritsu kyoukai Sougou Kenkyusho (日本能率協会総合研究所). 1985.『最新ＣＡＩ事情』. 東京：日本能率協会。

Nihon Shichoukaku Kyouzai senta (日本視聴覚教材センター). 1985.『教育におけるコンピューター利用』. 東京：日本視聴覚センター。

Otsubo, Kazuo (大坪一夫). 1984.「なぜＣＡＩか」.『名古屋学院大学外国語紀要』 11:3-5.

「パソコンと学校教育」.『読売新聞』 October 17, 1985:12.

「パソコン使った中学校教育の実情」.『朝日新聞』 January 22, 1986:14.

Philips, Martin. 1985. Logical Possibilities and Classroom Scenarios for the Development of CALL. In *Computers in English Language Teaching*, Christopher Brumfit et al (Eds.), 25-46. Oxford: Pergamon Press.

Rubin, Bella. 1986. Two Approaches to CALL: Curriculum-Based Courseware and Languages Exploration Activities. *TESOL Newsletter* 20 (1) Supplement No. 3:13-14.

Saeki, Yutaka (佐伯 胖). 1986.『コンピュータと教育』. 岩波書店 (黄版) 332. 東京:岩波書店。

Savignon, Sandra J. 1983. *Communicative Competence Theory and Classroom Practice*. Reading, Massachusetts: Addison-Wesley Publishing Company.

Stevens, V. 1983. English Lesson on PLATO, Review. *TESOL Quarterly* 17 (2):293-300.

Strei, Gerry. 1983. Format for the Evaluation of Courseware Used in Computer-Assisted Language Instruction (CALI). *CALICO Journal* 1 (2):43-47.

Takefuta, Yukio (竹蓋幸生). 1984.「英語教育とコンピュータ」.『名古屋学院大学外国語教育紀要』 11:67-74.

Underwood, John H. 1984. *Linguistics Computers and the Language Teacher: A Communicative Approach*. Rowley, Massachusetts: Newbury House Publishers, Inc.

Wyatte, David H. 1983. Three Major Approaches to Developing Computer-Assisted Language Learning Materials for Microcomputers. *CALICO Journal* 1 (2):34-38.