ReCALL **16** (2): 448–461. © 2004 Cambridge University Press
DOI: 10.1017/S0958344004001326 Printed in the United Kingdom

Another look at tutorial CALL

PHILIP HUBBARD
Linguistics Department and Language Center, Stanford University, Stanford, CA 94305, USA
(*e-mail: phubbard@stanford.edu*)

CLAIRE BRADIN SISKIN
Robert Henderson Language Media Center, University of Pittsburgh, G 17 Cathedral of Learning, Pittsburgh, PA 15260, USA
(*e-mail: cbsiskin@pitt.edu*)

Abstract

According to the dichotomy popularized by Levy (1995, 1997), computers may be used in language learning in either tutor or tool roles. Recently there has been a waning of interest in tutorial software for language learning in favor of tool-oriented applications. While we do not dispute the value of the computer as a tool, we argue that it is in the best interest of CALL practitioners and language teachers in general to be better informed about the realities of tutorial software so that the field can continue to grow and mature along multiple paths. In support of this position, we present a case for placing tutorial CALL back into the mainstream of the field. After a discussion of several common myths about tutorial CALL, we offer six reasons as to why tutorial CALL has been marginalized. We follow with some evidence that, despite the presence of these myths and this marginalization, CALL practitioners around the world continue to find tutorial CALL appealing for research and development. Specifically, we review the presentations at four major CALL Conferences in 2002 and classify them as primarily focused on tutor or tool uses. Although tool-oriented applications dominate overall, the results show a strong continuing interest in tutorial applications as well. We conclude with proposals for (1) reconsidering the dichotomous nature of the tutor-tool distinction, (2) changing the defining characteristic of tutorial CALL from evaluation to teaching presence, and (3) suggesting several aspects of language learning that tutorial CALL is best suited to support, given the present state of technology.

1 Introduction

An important distinction for CALL theory and methodology has been that between the use of computers as tutors to teach language learners and their use as tools to support learners in an active learning process. This dichotomy was popularized in a framework by Levy (1997: 180–185), whose extended definition of these concepts can be summarized as follows. In the *tutor role*, the computer evaluates the learner, controls the learning process and temporarily substitutes for the human teacher. In the *tool* role, the

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

computer does not evaluate; rather, it increases the efficiency or effectiveness of actions related to learning and may involve the teacher and learner both. Vocabulary flashcard and grammar drill programs are examples of computers operating in tutor roles; word processors, web browsers, and email programs are examples of computers operating in tool roles. Levy justifies the need for this distinction in the opening to the chapter introducing his framework. "The tutor-tool framework is offered here as a mechanism for addressing a number of concerns in CALL, the most far-reaching being the belief that all CALL is of the tutorial type, characterized by one-to-one interactions where the computer evaluates the student input and then presents the new material accordingly" (*op.cit.*:178).

However, in the relatively few years since those words were published, it is our impression that the situation has been reversed and that the role of the computer as a tool has come to dominate. Indeed, in his review of Levy's 1997 book, Wolff (1999: 127) remarks, "On the basis of our present knowledge of language learning, I believe that the only defensible role for the computer in language learning is that of tool." He adds later, "In learner autonomy, the key concept in modern language learning and in all humanistically oriented pedagogy, the computer as tutor cannot have a place" (*op. cit.*: 128). Wolff is not alone in his pronouncements, and given the often weak link between commercial software and contemporary language teaching research and methodology, his position is probably more mainstream than the radical tone of his words suggests.

We do not, however, share that position. We believe that the tutor role for the computer is still eminently justifiable and that the field of CALL will be more robust if efforts in developing both tutor and tool-oriented applications and techniques for using them appropriately continue. Our focus here will therefore be on reassessing the role of "computer-as-tutor" for language learning, or more simply "tutorial CALL."

In this paper, we offer a case for moving tutorial CALL out of the margins of the field of language learning. We begin with a discussion of some pervasive myths about tutorial CALL that have helped lead to its perceived irrelevance in contemporary language teaching. We next offer some reasons as to why tutorial CALL has been marginalized at the professional level. We follow with a review of conference presentations in 2002 showing that despite the presence of these myths and this marginalization, CALL practitioners around the world continue to find tutorial CALL appealing for research and development. We conclude with a proposal for reconceptualizing tutorial CALL by (1) softening of the tutor-tool dichotomy, (2) redefining tutorial CALL in terms of *teaching presence* in the software and (3) identifying the specific areas of language learning in which tutorial CALL can most effectively be positioned.

2 Myths

Here we discuss six myths about tutorial CALL. As myths, they are not always expressed directly but as will be seen are implied in widely cited published work and pervade teacher attitudes both within CALL and within the language teaching field as a whole.

Tutorial CALL is behaviorist. The tag of "behaviorist" is often attached collectively to CALL software. In a popular description of the history of the field, "behavioristic" CALL is noted by Warschauer (1996) and Warschauer and Healey (1998) as the first

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

phase of CALL, followed later by communicative and then integrative CALL. Since little of tutorial CALL software can be considered by their criteria to be either communicative or integrative, the conclusion seems to be that what remains must fall into the behaviorist category. However, Warschauer (1996) clearly acknowledges that the communicative phase of the 1970s and 80s included programs for skill building involving significant interaction in which the student had choice and control and which were presumably not behaviorist.

In responding to this notion of tutorial CALL equaling behaviorist CALL, there are two points to be made. The first is that although behaviorism as an overall theory of learning has been discredited, some contemporary researchers concede it may have a place in certain areas such as "the acquisition of vocabulary and grammatical morphemes" (Lightbown & Spada, 1999: 26). It may also be useful in accounting for certain types of phonological development and the acquisition of some of the features of idioms and other fixed phrases. More significantly, much of the current tutorial software is arguably not behaviorist but adheres more closely to the cognitive/constructivist perspective as outlined in Warschauer and Kern (2000). In discussing audiolingualism (a behaviorist approach) within the structuralist perspective, they note the following:

> Strongly influenced by the work of behavioral psychologists such as John Watson and B.F. Skinner, structural methodologists conceived of language learning as habit formation and thus saturated students with dialogues and pattern drills designed to condition learners to produce automatic, correct responses to linguistic stimuli… Practice, not abstract knowledge, was the key (*op. cit.*: 3).

In discussing the cognitive perspective, on the other hand, they note that it involves providing "language input and analytical and inferential tasks" (*op. cit.*: 13). A look at many of the current crop of tutorial programs will show that (1) many involve conscious manipulation of language forms based on abstract rules, which is not within the behaviorist tradition, and (2) many are based on reading or listening comprehension and even allow, through the request for meaning aids such as glossed words or transcripts, a form of meaning negotiation (albeit a programmed one). Thus the argument that all or most CALL software is behaviorist is not tenable.

Tutorial software is all drill and practice. This idea is closely related to the behaviorist notion as evidenced by the popularity of the phrase "drill and kill," which interestingly, used as a synonym for drill and practice, dismisses the explicit concept of practice altogether. A language drill is an activity that is repetitive and aimed at automatizing some specific aspect of language knowledge or use. Minimal pair recognition exercises are drills, as are audiolingual classics such as substitution drills. Practice on the other hand is not so easily defined, but seems when used in this sense to mean the conscious, reflective manipulation of language rules and forms with an aim toward increasing knowledge, fluency, or accuracy. Seen this way, translation exercises or grammar exercises that involve the application of an abstract rule or rules, such as those governing the English article system, would constitute practice but not drill. There are other forms of software that do not obviously fit into this category. Any computer-based activity that involves reading or listening to novel material for the purpose of building comprehension skills does not readily fit the "drill and practice" label (unless one wants to extend

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

the term "practice" so that it covers almost every language learning activity). And while programs that use dialogues primarily as models for learners to memorize and parrot fit this category, those that allow choice and branching in an attempt to simulate a conversation or engage the learner more actively do not. In fact, any activity that encourages exploratory learning, such as Storyboard-type programs involving full text reconstruction, are outside this category.

Tutorial software isn't useful because it isn't communicative. Levy (1997: 29) refers to general concerns that CALL does not fit into the communicative approach to language teaching, and indeed tutorial CALL is often denounced for its failure to provide real communication. While computers themselves fall far short of offering the quality of communication provided by human interaction, tutorial software can easily be used in communicative ways. In fact, any tutorial program that can be used by individual students in isolation automatically has the potential to become communicative when it is used by pairs or groups of language learners. As examples, a classroom teacher may demonstrate a CALL exercise to the whole class and ask for their responses to the group at large. Smaller groups of students may work together to complete exercises. Egbert (1999: 28–29) asks, "When is a drill not a drill? The answer is, when it is an interactive activity". She suggests several ways to add interactivity among language learners as they use tutorial software. Various roles may be assigned to students in a group. One student may look at the computer monitor while others supply the necessary information on a handout. One student may look up information in reference text and transmit it to others, who perform tasks on-screen. The subject matter of a tutorial software package may itself provide the basis for communication. When the answers and explanations provided by tutorial CALL software are incomplete, an information gap is created, and students have further reason to communicate (Bradin, 1999: 161). Meskill (1999: 150) fits what she terms the 'workbook format' into her scheme of the computer as a socio-collaborative tool, since students can work together to develop questions and exercises to pose tasks for one another. Finally, while communicative tasks and activities undeniably play a central role in modern language teaching methodology, there have been developments in SLA theory showing the importance of activities promoting a focus on form in addition to meaning (see Chapelle (2001) and Egbert & Hanson-Smith (1999) for discussions of this dimension in CALL). Thus, appropriately designed non-communicative CALL software that aids development of knowledge and awareness of form can play a legitimate role in language learning.

In tutorial CALL, the teacher is absent or has no significant role. Levy (1995: 241) implies that in tutorial CALL, the teacher's role may be reduced or completely excluded: "The absence of a significant role for the language teacher to play and the notion of the machine catering for all contingencies of teaching and learning have been major components of the tradition that places the computer in the role of tutor". Certainly in many cases tutorial software is used by individuals without the presence of a teacher. This may be the case in self-access centers or in circumstances in which language students work through CALL materials outside of class time or even apart from any formal educational institution. However, even in instances where use occurs outside of the classroom, teachers may be involved to a high degree. They may have been involved in evaluating and selecting the software and in designing the syllabus in which the software was integrated. They may have also assigned the exercises and monitored

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

the student's progress. If the software has tracking capabilities and reports on the students' activities, teachers may incorporate this feedback into follow-up activities in the next class.

In many situations, classroom teachers arrange to be present while their students use tutorial CALL. When a tutorial CALL activity is the focal point of a given class, teachers perform a useful function in providing additional explanations of items not handled well by the computer and serving as an important resource by elaborating on various points. Robinson (1989: 132) asserts, "Ensuring that CALL is an integral rather than separate component of the total program of instruction and that the teachers are likewise an integral part of the CALL laboratory is an essential first step".

An even higher level of involvement with tutorial CALL exists when language teachers produce software for use by their own students. Hubbard (1992: 55) argues for software that is designed for specific language learners. He states, "At some stage in development, the considered input of a teacher experienced with the target learners is needed to ensure that the content of the software enhances rather than detracts from its effectiveness". A wide variety of CALL authoring software exists to simplify the process of creating computer-based language materials. Classroom teachers may create more CALL materials for their own students than is generally believed, as is attested by the fact that at the time of this writing there are more than 30,000 registered users of the popular *Hot Potatoes* suite of authoring tools. Teacher-made software may be seen as an extension of the teacher's personality and expertise. As Rézeau (1997) observes,

> In the designing process of the CALL instrument, the teacher has a capital role to play as mediator of knowledge. By putting a part of himself into the machine, by making his own knowledge and expertise accessible and visible to the students, he can provide them with a rich cognitive apprenticeship environment, made even richer when peer-teaching is allowed to take place.

In tutorial CALL, the learners have no control. The assumption that the computer is always in charge in tutorial CALL does not hold up to close scrutiny. The benefits of designing CALL materials so that learners have many choices have long been recognized (Stevens 1984, 1992). With current tutorial software packages, learners may in fact have a high degree of control. They may be able to choose the theme for an activity, and within that theme, they are often presented with various activity types. Learner control might involve the ability to exercise various options, such as using help screens, online dictionaries, hypertext glosses, search functions, and explanations of grammar. Some exercises permit learners to seek a 'hint' before or after they supply an answer, and they may or may not have access to the correct answer. They may be able to set in advance the length of time that information will remain on the screen (Hubbard, 1992: 61). Learners may also have the freedom to determine the sequence in which they will do certain activities and even the manner in which they will do them. For example, in some exercises they are able to choose between a testing mode, in which they will answer questions and receive a score at the end, and a tutorial or practice mode in which they will receive feedback after answering each item. In either mode, the final score may include a report of which items they missed, and at this point, they may have the opportunity to return directly to those items and re-do them, rather than having to repeat

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

the entire exercise. Alternatively, they may opt to go on to another exercise or conclude the session.

In any event, the issue of how much control learners should have in CALL has not been settled (Cobb & Stevens, 1996: 135). Hoven (1999: 161) suggests that more structured CALL materials be available to language learners at the beginning but that as they become more capable of dealing with choices, they may be accorded more autonomy and control. She believes that it may be necessary to provide various levels or modes depending on the level of learner control desired. Similarly, Boling and Soo (1999) offer evidence that novice learners may want less control than advanced learners, preferring software that has clearly laid out paths and learning objectives.

It seems that as learners gain more freedom in the options included in a tutorial CALL package and become more comfortable with these options, the software in fact becomes more like a tool to be used at their discretion. Many students routinely opt to make use of tutorial CALL software before a test. Perhaps to many of the current generation of learners, all uses of the computer are tool-like. The scenario becomes even more complicated when language learners use authoring systems or templates to create their own tutorial CALL software. Since in this case the authoring program is a tool while the exercises that the students create for themselves fulfill a tutor role, the distinction becomes blurred.

If it's on a disk or CD, then it's tutorial software; if it's on the Internet, then it's 'communicative.' Many uses of CALL before the Internet involved tutorials, and none of the early uses of the Internet were tutorial in nature. As a result, the perception arose that all Internet-based CALL is automatically more 'communicative' and all disk-based CALL is tutorial. In reality, there have been ample models for non-tutorial uses of computers in language learning for the last 20 years. Word processing programs, spreadsheets, and other tools have been exploited in communicative ways (Hardisty & Windeatt, 1989). Computer-based adventure games, simulations, and exploratory software such as concordance programs are not tutorial in nature and have long offered important options in *CALL*. Non-tutorial software existed before the appearance of the Internet, and continues to reside on disks and local area networks.

Even before the Internet existed as such, there were numerous attempts to implement synchronous or asynchronous communication via computer with language learners (Bradin, 1984), and with the advent of the Internet, the value of computer-mediated communication (CMC) in the form of e-mail, chats, and bulletin boards became widely recognized. As new capabilities of the web emerged, interaction between learners and web pages became possible (Mills, 1999). There now exist innumerable instructional websites designed specifically for language learning, and these are becoming increasingly sophisticated. However, it is worth noting that to date the great majority of tutorial software available on the web is less interactive and pedagogically rich than what was previously available on disk and CD-ROM.

3 Why has tutorial CALL beem marginalized?

Tutorial CALL has never been well-respected within the general ranks of language teachers, though tool-oriented CALL appears on its way to making it into the mainstream. Why is this the case? We believe there at least six reasons why tutorial CALL has been marginalized by the language teaching profession.

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

1. Teachers believe the myths about it. Once teachers are led to believe that CALL software is behaviorist (a pejorative term in language teaching), that it is mostly "drill and kill" (read "boring" and "useless"), that it isn't communicative, and so on, then it is easy to ignore it.
2. It hasn't lived up to its expectations. There were high expectations surrounding tutorial CALL in the early to mid 1980s with the spread of microcomputers and again in the early to mid 1990s with the advent of multimedia programs. Arguably, the material that came out of these periods did not change language learning practice significantly and much of it did not even represent an improvement. Teachers looking for a revolution went away disappointed.
3. It appears inconsistent with some contemporary learning theories. It is a stretch to say that there is much in the way of tutorial software that is consistent with what Warschauer and Kern (2000) call the sociocognitive perspective of language learning. However, if there is one thing that over two decades of SLA research have told us, it is that language learning is a very complex and idiosyncratic phenomenon, and no one theory or integrated combination of theories seems to be able to offer explanations that are of much predictive value. While theoretical justifications are always to be strived for, it does not mean that an inconsistency with one learning theory is an inconsistency with good teaching practice. However, those who are anxious to attach themselves to the latest theoretical bandwagon will find that the connections to tutorial CALL are not obvious.
4. It is dominated by commercial interests. Compared to textbooks, commercial CALL software is expensive and the content and pedagogy has often been of low quality. In many cases the companies producing the material are newcomers and have not earned the trust of teachers the way that print publishers have, and teachers may see them as outsiders primarily interested in making money. In other cases where print publishers have ventured into the field, the results have often been unsatisfactory.
5. It isn't new. There has always been a tendency in language learning, and in education in general, for teachers' attention to be drawn toward the latest fashion. The rapid spread of the Internet has engaged the interest of CALL practitioners in CMC and web-based activities. As evidence for this trend, in the past five years or so, a number of books have been published on the Internet and other tool-oriented uses such as concordancing, but not a single book dealing in a significant way with tutorial CALL has appeared. As we will see in the next section, although interest in tutorial CALL remains high in CALL specialist conferences, it is much lower in general language teaching conferences.
6. The teacher's role seems limited. Back in the early days of CALL, the field was driven by teacher-programmers. Over time, as computers have become more complicated, the number of teacher-programmers seems to have declined. Although in many ways it is easier to produce professional looking materials these days, the amount of effort required is still substantial. Thus, the majority of teachers who do write tutorial exercises rely on relatively fixed templates such as those provided by Hot Potatoes. Additionally, when teachers use textbooks, they are working in a familiar domain. They can supplement the text with handouts

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

they produce in the same medium, and there is a sense that they are capable of creating a textbook of their own if they wish. This is not the case with anything but the simplest tutorial software. When tool-oriented CALL is considered, however, teachers are working in a more familiar domain. They are quite likely to already be using email and word processors, and with very little assistance they can be taught to construct a web page using a dedicated web editor or even just Microsoft Word.

4 The state of tutorial CALL

It has been argued here that tutorial CALL has been marginalized within the general field of language learning, but what about within the field of CALL itself? Are CALL practitioners also primarily focusing on tool uses of computers for language learning? In exploring that question, we provide some evidence of trends based on an analysis of presentations made at four conferences in 2002: the CALICO Symposium in March, the CALL portion of the TESOL conference in April, the EUROCALL Conference in August, and the CALL Journal Conference in August. We reviewed the conference abstracts and on the basis of the content there classified each presentation into one of four categories:

- Presentations involving tool-oriented CALL only
- Presentations involving tutor-oriented CALL only
- Presentations involving both
- Presentations involving neither (teacher training, language testing, etc.)

Note that the study is limited to the information that appears in the published conference programs rather than what may actually have been presented. The data appear in Table 1.

The results from this review suggest that while tool-oriented CALL holds an edge overall, tutorial CALL still holds great interest among practitioners worldwide. In fact, it is only within TESOL that the presentations are so overwhelmingly skewed toward tool uses. TESOL differs from the other conferences listed in that it is not specifically for CALL, so at least within the English as a second/foreign language profession, it can be seen as more representative of general teaching practice and interest than the three CALL specialist conferences. Another possible explanation is that many TESOL presentations are made by masters-prepared teachers whose work is either unfunded or funded at a minimal level and who use computers in their teaching but probably do not consider

Table 1. *Tool versus tutor-oriented CALL presentations at four conferences in 2002. All but TESOL were dedicated CALL conferences*

	CALICO	TESOL	EUROCALL	CALL
Tool	57	42	37	11
Tutor	46	7	30	25
Both	7	11	26	11
Neither	17	19	14	6

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

themselves CALL specialists. As noted in the previous section, tool-oriented CALL seems much more accessible to such teachers.

Interestingly, this review shows another trend. At all four conferences and at the EUROCALL conference in particular, we see a number of presentations involving a combination of tutorial and tool-oriented uses. Some of these are descriptions of language lab projects while others present blended applications or language courses that draw on multiple applications, some tutorial, some tool in nature. These combinations suggest the labels of tutor and tool are not absolute and that the notion that these concepts represent a dichotomy may be in need of revision.

5 Revisiting the tutor-tool dichotomy

As mentioned previously, the tutor-tool distinction in CALL literature was popularized by Levy (1995, 1997), who traces the historical roots of the tutor/tool model through Skinner (1954), Taylor (1980), Kenning and Kenning (1983), and others. We have noted throughout this paper that the distinction is often treated as a dichotomy. As outlined in the introduction, the tutorial role is seen as one in which the computer assesses and exercises control over the learner and acts as a surrogate teacher. The tool role is viewed as one in which the computer does not evaluate but enhances the learning process. The teacher's role is generally considered to be absent in the tutorial role and one of great involvement in the tool role.

However, there is evidence that the tutor/tool distinction may no longer be clear-cut. We have already seen that learners may enjoy a fair amount of control in being able to exercise many options in tutorial CALL and may view computer drills as a tool to enhance their learning. Teachers may be actively involved in both the use and creation of tutorial software. Since today's learners are increasingly proficient with software tools such as email and word processing, the instructor's role may be less than it was previously when computers were used in 'non-tutorial' ways.

Viewing tutor and tool-oriented CALL as mutually exclusive categories does not accurately reflect the reality of today's uses of CALL. Decoo and Colpaert (1999), for instance, note that many current CALL software packages include both tutor and tool functions. As an example they offer *Verbapuces*, in which students have the option of looking up information but can also work through lessons and be tested on that specific material. Richmond (1999) points out the dual functions in another popular software package *Transparent Language*. Levy (1999: 89) recognizes that tutor and tool can work together in "hybrid systems". In fact, both roles are available to some extent in many CALL software packages. Given this convergence, it seems reasonable to consider that tutor and tool could be parallel rather than opposing categories. There will still be clear instances of applications where one or the other seems to be the appropriate label, and we do not propose at this time that the distinction be eliminated. However, it may be more fruitful to speak of the *degree* to which a particular software application embodies the characteristics of tutor, tool, or both and the *quality* of the software as a tutor, tool, or both rather than forcing a classification as one or the other. Most importantly, even when we can make a distinction, we should not automatically assume that a software application is pedagogically less advanced because it is tutorial rather than tool-oriented.

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

6 The teaching presence in CALL software

In Levy's initial conception, the key element defining a tutor was evaluation – tutorial software is software that evaluates in some way. With this definition, he concluded that spellcheckers played more of a tutor than a tool role because they evaluate (Levy, 1997: 181), a somewhat unsatisfying conclusion since arguably neither the designers nor the users of spellcheckers see their central function as being spelling teachers. However, he also conceived of the tutor role as one in which the computer is a temporary substitute for a live teacher. While we have argued above that the live teacher is not necessarily absent when learners are using tutorial software, the software can still be "teaching". In fact, one way to view tutorial software is to see it as an extension through time and space of the *teaching presence* of its designer.

Once we shift the tutorial concept away from applying exclusively to the evaluative dimension, we can include other activities relating to teaching that do not involve evaluation. For example, we can include exploratory programs or programs with meaning aids such as hypertext glossaries, captioning, grammatical explanations and other elements placed there specifically to support language learning. If we focus on this broader teaching role, we can expand the definition of tutorial CALL as follows.

> *Tutorial CALL* refers to the implementation of computer programs (disk, CD-ROM, web-based, etc.) that include an identifiable teaching presence specifically for improving some aspect of language proficiency.

Under the preceding definition, spellcheckers would not be tutorial, and software designed for other instructional purposes, such as to teach chemistry or history, would be tutorial but not tutorial CALL. A web page with instructions for going to another web page for some purpose related to language learning would be tutorial (to a degree), because there would be a manifest teaching presence. Simply put, if the intent in design is to support language learning, then it involves a measure of tutorial CALL; if not, then it doesn't. The word *involves* is important here because it is consistent with the idea in the preceding section that tool-oriented and tutor-oriented could be parallel dimensions in CALL applications rather than mutually exclusive categories.

It is worth pointing out that this line of reasoning brings us to the conclusion that print textbooks are tutorial in nature; that is, they have a teaching presence. Does that same textbook put on a computer as a pdf file suddenly become tutorial CALL? Presumably it does. So what does this proposal gain us? By defining tutorial CALL in terms of teaching presence, we can now go about attempting to determine through research, development, and practice what forms, degrees, and qualities of teaching presence lead to *effective* tutorial CALL for particular groups in particular settings. We can readily acknowledge that there are computer applications in which the teaching presence does not represent very good teaching. We can especially note that the software may not make good use of the one-to-one student-teacher relationship that the computer affords. A non-interactive textbook placed in the computer medium is clearly not optimal teaching in that environment any more than a live tutor simply lecturing to a single student would be optimal teaching in a live environment. However, it still represents information and ideas structured by a teacher for learning purposes, and though the teaching presence can be seen as weak or degraded relative to a live teacher, it may still be

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

somewhat effective, just as a print textbook can be.

Defining tutorial CALL in terms of teaching presence blurs the tutor-tool dichotomy further in an interesting way. Contemporary language classrooms focus on the learner and contemporary characterizations of language teachers often position them as guides or resources who support learners on the road to autonomy. Given Levy's original definition, the contemporary view casts teachers more in the role of tools than tutors. With the revised definition, tutorial CALL can be compared more readily with live human teaching and both synchronous and asynchronous online teaching.

7 The place of tutorial CALL

One of the long-term problems of tutorial CALL has been the impression that its objective was to *replace* the teacher. The commercial sector has all too often been guilty of perpetuating this idea by marketing programs that claimed to teach all skills well. In this section, we identify the areas where well-designed tutorial CALL applications are most likely to have a positive impact. These are precisely the areas where the teaching presence can provide direction and feedback that is able to be anticipated with some degree of confidence.

Let us begin, though, by looking at some areas in which tutorial CALL has not been successful. Despite numerous attempts over the past 20 years or so, the computer is still not very helpful in teaching conversation. It can assist with certain limited aspects of speaking, such as providing models through lines of dialogues that can be recorded by the user and compared to the original, or by offering pronunciation evaluation through speech recognition software. There have even been a few attempts to allow branching conversations through voice input. In all these cases, however, there is at best an artificial form of oral communication being practiced.

Tutorial software similarly is of limited value in the teaching of writing, though knowledge *about* writing can be taught tutorially. Again, in terms of communication the computer does not provide a credible audience for the writer, though it remains the production *tool* of choice for most communicative purposes these days.

We believe the most promising uses for tutorial CALL to be the following.

1. Helping the learner pick up conscious knowledge of the language, including the learning of language forms, patterns, and rules. Here, we have the "traditional" CALL activities, such as grammar and vocabulary development exercises. While the learning path is not always fully controlled by the software here, it is clearly guided by it. This allows a higher degree of predictability and makes it possible in principle to anticipate learner actions and develop responses to them. For a simple example, in a fill-in-the-blank type of grammar exercise, the range of expected answers is usually quite limited. The teacher determining the program's responses can come up with feedback targeted differentially to the anticipated student input and present a response analogous to what a live teacher would do in the same circumstance. This anticipatory interaction can be equivalent in these limited domains to a live interaction. Unfortunately, most present-day software makes little use of this potential.
2. Helping the learner improve proficiency in the receptive skills of reading and

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

listening. Dedicated reading software can be programmed to allow learners not only to read and respond to comprehension questions, but also to control reading pace and make use of various meaning aids, such as glossaries. Perhaps the most exciting use of CALL software is in the area of listening. The technology can allow instantaneous control of pause and repeat/rewind to aid comprehension, provide meaning aids including slowed versions, text and glossary support, and even translations, and add visual support through static or animated graphics, photos, or video. Although some of these features, such as links to online dictionaries and control of speed and pacing, can be integrated automatically as tools under full control of the learner, the production of dedicated CALL listening programs is particularly useful at lower levels and with students who have not been trained in how to use these features effectively for independent learning.

3. Helping the learner improve pronunciation. One of the most active areas in tutorial software development currently is pronunciation, particularly within ESL/EFL. These programs can provide multimedia models with audio, technical description and graphic visualization of sounds. They allow the user to record and compare, and they provide various forms of sound visualization including wave forms, pitch contours, and spectrograms. Some even use a speech recognition engine to grade the accuracy of the pronunciation relative to some native standard, though the results are often unsatisfactory. We can expect the technological and pedagogical sophistication of these to improve as research results and feedback from users make their way back to inform future versions.

The preceding examples are not meant to suggest that tutorial CALL software will not eventually be able to support production and social interaction effectively. We can expect advances in intelligent CALL in the future leading toward simulated conversation partners and meaningful feedback on writing, at least within specific, predictable types of discourse. At present, though, most commercial developers would probably do better to concentrate their efforts in the three areas mentioned and save production and social interaction for face-to-face or CMC environments.

8 Conclusion

In this paper, we have argued that tutorial CALL, however it is defined, is still a viable part of CALL and deserves serious attention rather than summary dismissal. We have also proposed that the tutor-tool relationship should be seen not as dichotomous, where they are mutually exclusive labels, but more as involving dimensions or matters of degree. Along those lines, we have suggested that the most useful characteristic for defining a tutorial CALL application is not evaluation but a broader concept of "teaching presence" that subsumes evaluation along with other teacher roles. Given that CALL as an identifiable field within language teaching has only been around a little over twenty years, and that those twenty or so years have seen enormous changes in the technological foundation of the field, it is not surprising that we are still in the process of resolving fundamental conceptual and terminological issues. What is important is that we do not shut the door on promising areas for reasons of dogma. Research and development of both tutor- and tool-oriented CALL should move forward, individually and in

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

combination, to realize the potential of this remarkable technology for improving human language learning.

References

Bradin, C. (1984) Telecommunications in the ESL classroom. *Colorado TESOL (CoTESOL) Conference*, Denver.

Bradin, C. (1999) CALL issues: Instructional aspects of software evaluation. In: Egbert, J. and Hanson-Smith, E. (eds.), *CALL Environments: Research, Practice, and Critical Issues*. Alexandria: TESOL, 159–175.

Boling, E. and Soo, K. (1999) CALL issues: Designing CALL software. In: Egbert, J. and Hanson-Smith, E. (eds.) *op. cit.*, 442–458.

Chapelle, C. (2001) *Computer Applications in Second Language Acquisition: Foundations for Teaching, Testing, and Research*. Cambridge, UK: Cambridge University Press.

Cobb, T. and Stevens, V. (1996) A principled consideration of computers and reading in second language learning. In Pennington, M. (ed.), *The Power of CALL*. Houston: Athelstan, 115–136.

Decoo, W. and Colpaert, J. (1999) User-driven development and content-driven research. In: Cameron, K. (ed.) *CALL: Media, Design, and Applications*. The Netherlands: Swets & Zeitlinger, 35–58.

Egbert, J. and Hanson-Smith, E. (eds.) (1999) *CALL environments: Research, Practice, and Critical Issues*. Alexandria, VA: Teachers of English to Speakers of Other Languages.

Egbert, J. (1999) Classroom practice: Creating interactive CALL activities. In: Egbert, J. and Hanson-Smith, E. (eds.) *op. cit.*, 27–40.

Hardisty, D. and Windeatt, S. (1989) *CALL*. Oxford: Oxford University Press.

Hot Potatoes. Authoring software produced by Half-Baked Software. Available at http://web.uvic.ca/hrd/hotpot/

Hoven, D. (1999) CALL-ing the learner into focus: towards a learner-centred model. In: Debski, R. and Levy, M. (eds.) *WorldCALL: Global Perspectives on Computer-Assisted Language Learning*. The Netherlands: Swets & Zeitlinger, 149–167.

Hubbard, P. (1992) A methodological framework for CALL courseware development. In: Pennington, M. and Stevens, V. (eds.), *Computers in Applied Linguistics: An International Perspective*. Clevedon, Avon: Multilingual Matters, 39–76.

Kenning, M. M. and Kenning, M. J. (1983) *An Introduction to Computer Assisted Language Teaching*. Oxford: Oxford University Press.

Levy, M. (1995) Integrating CALL: the tutor and the tool. In: Gimeno, A. (ed.), *Technology Enhanced Language Learning: Focus on Integration*, *Proceedings of EuroCALL '95*. Valencia: Universidad Politécnica de Valencia, 239–248.

Levy, M. (1997) *CALL: Context and conceptualization*. Oxford: Clarendon Press.

Levy, M. (1999) Design processes in CALL: integrating theory, research, and evaluation. In: Cameron, K. (ed.), *op. cit.*, 83–107.

Lightbown, P. and Spada, N. (1999) *How Languages are Learned*: Revised edition. Oxford: Oxford University Press.

Meskill, C. (1999) Computers as tools for sociocollaborative learning. In: Cameron, K. (ed.), *op. cit.*, 139–162.

Mills, D. (1999) Interactive Web-based language learning: the state of the art. In: Debski, R. and Levy, M. (eds.), *op. cit.*, 117–131.

Rézeau, J. (1997) The learner, the teacher, and the machine: golden triangle or Bermuda triangle? Presentation at EUROCALL 1997. http://www.uhb.fr/campus/joseph.rezeau/articles/dublin/Dublin97.htm

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press

Richmond, I. (1999) Is your CALL connected? Dedicated software vs. integrated CALL. In: Cameron, K. (ed.), *CALL: Media, Design and Applications*. The Netherlands: Swets & Zeitlinger, 296–314.

Robinson, G. (1989) The CLCCS CALL study: methods, error feedback, attitudes, and achievement. In Flint Smith, W. (ed.), *Modern Technology in Foreign Language Education: Applications and Projects*. Lincolnwood, IL: National Textbook Co., 119–134.

Skinner, B. F. (1954) The science of learning and the art of teaching. *Harvard Educational Review,* **24**: 86–97.

Stevens, V. (1984) Implications of research and theory concerning the influence of control on the effectiveness of CALL. *CALICO Journal,* **2** (1): 28–33.

Stevens, V. (1992) Humanism and CALL: A coming of age. In: Pennington, M. and Stevens, V. (eds.), *op. cit.*, 11–38.

Taylor, R. (1980) *The Computer in the School: Tutor, tool, tutee*. Teacher's College, Columbia University, New York: Teacher's College Press.

Warschauer, M. (1996) Computer-assisted language learning: an introduction. In: Fotos, S. (ed.), *Multimedia Language Teaching*. Tokyo: Logos International, 3–20.

Warschauer, M. and Healey, D. (1998) Computers and language learning: an overview. *Language Teaching*, **31**, 57–71.

Warschauer, M. and Kern, R. (eds.) (2000) *Network-Based Language Teaching: Concepts and Practice*. New York: Cambridge University Press.

Wolff, D. (1999). Review of "Levy, M., CALL: Context and Conceptualization." *System*, **27**: 119–132.

https://doi.org/10.1017/S0958344004001326 Published online by Cambridge University Press