Synchronizing language pedagogy and language software

Dr. Deborah Healey, University of Oregon
dhealey@uoregon.edu
http://www.deborahhealey.com/

This article will explore the relationship between software and pedagogy from the 1970s to the present. It will examine the many ways that computers have been used badly in language teaching and the ways that they can be used well. I will discuss expectations and results from the past, then explore expectations and probable results from the present and future. As a reminder, the Internet wasn't a factor until the mid- to late-1980s with email, and the Web didn't arise as a factor until the mid-1990s. In the beginning, all we had was software.

1970s-1980s: PLATO

The first PLATO (Programmed Logic and Automated Teaching Operations) system was created in 1960, but became useful as a teaching platform only with PLATO III in 1969. Students used terminals (See Figure 1) attached to a mainframe computer that was originally located at the University of Illinois. The courseware was teacher-developed, and foreign language learning was one of the focus areas. Much of the software was drill and practice. However, the exercises included sophisticated error-correction and record-keeping. It also allowed the teachers who were part of the development community to communicate with each other electronically, becoming the first bulletin board system. In the 1970s, a speech synthesizer added the ability to "speak" - though the speech was far from useful for accent training.

PLATO fit a certain kind of pedagogy, especially in the 1970s. While the exercises were primarily drills, learners could have some choice in the exercises they used, depending on how their teacher used the computer. Error-correction and feedback to the learners could be extensive, where it had been programmed in. Where it was available, appropriate feedback allowed learners to make better decisions about what they needed to know and what to learn next. Sound became an option in later versions of PLATO, as well.

Figure 1. PLATO terminal

As we will see again and again, teachers were the key to whether or not the learners were empowered to make choices or not, and how the exercises were used in the classroom. The high cost of the terminals and time on the mainframe prevented PLATO from being widely used, thus reducing its impact on teaching and learning.

1980s: Microcomputers and CALL

Computers did not play a broad role in language teaching until the BBC Micro (United Kingdom - Figure 2), Apple IIe (United States - Figure 3), and IBM PCs in the 1980s. Unlike the PLATO system, microcomputers were inexpensive enough that they could be used in classrooms and labs by a broad range of people. Teachers who used these early computers were very excited about them and looked for the advantages that technology offered in the language classroom. The ability to


Figure 2. BBC Micro

give “immediate feedback,” to repeat as many times as needed, and to customize instruction (to some extent) were all mentioned as major advantages of technology.

It was during this period that some different typologies of CALL were developed. Notable among them were John Higgins’ (1987) "magister" and "pedagogue." Software described as magister was one that directed student activity, generally drill and practice. Pedagogue-style software was envisioned as like the educated Greek slave who was there to answer questions posed by the young master – software that served as a helper, not a taskmaster. Programs like word-processors, dictionaries, simulations, and concordancers were pedagogues rather than magisters. This was an important distinction and encouraged the use of software that would hand over some control to the learner.

Figure 3. Apple IIe

Taylor and Perez (1989) expanded the typology to “knower of the right answer,” “workhorse,” and “stimulus.” The same typology was used by Healey in the 1995 sequel. The “knower” incorporated Higgins’ magister but expanded it to include games such as Hangman and cloze exercises. This category included any program where the learner was expected to find the answer that the computer expected. More sophisticated programs, such as GapMaster from Wida, allowed more than one correct answer and gave feedback – if the teacher who authored that exercise included these options.

Workhorse programs included word-processors, grammar checkers, sound recorders, and authoring systems. These were tools that could be used to teach, but they had a purpose in the real world outside of teaching. Word-processing was coming into its own during this time, seen by most people as easier than using a typewriter for writing where one might make mistakes. Colette Daiute's *Writing with Computers* (1985) solidified the benefits of using technology to improve students' writing skills. Word-processors made revision, an essential element of process writing, easier for students. As a result, students offered less resistance to the concept of repeated cycles of writing and revision. That workhorse programs also had real-world applications was a strong advantage. Teachers felt that they were preparing students with more than just English, and that these programs were therefore more authentic than ones with a purely language focus.

Stimulus programs included simulations, text reconstruction, concordancers, and multimedia dictionaries. As Taylor and Perez say, this is where “the computer and the program serve to evoke language use by the students” (p. 63). The goal is to solve a puzzle or a problem, create something, or discover information. This type of program was seen as the most interesting for students and potentially the most rewarding – but also the one that took the most time and effort to set up in such a way that language learning could occur. If learners worked by themselves on simulations or games, they tended to use very little of the target language other than what they had to read on the screen. If learners worked in groups but one person monopolized the keyboard, similarly little language learning benefit was likely. The best scenarios had learners working in groups, each person with a specific role to play, and with the expectation of an off-computer language outcome, such as a report. A lot of teacher preparation was required. As the title of Chris Jones’ article so aptly put it, “It’s not so much the program, more what you do with it.”

CALL proponents emphasized that any program, no matter how drill-like, could be used to encourage communication if learners worked in pairs on a task. Research on the type of conversation taking place at a computer pointed out that yes, it was possible to encourage communication in pairs, but the quality of communication depended greatly on the task set for the students. For example, I noted some interesting student interaction coming from a pair of students working on a grammar drill from Grammar Mastery (circa 1988). Students tried the exercise, looked at the help screen, and continued talking with each other about just why conditionals in English worked the way they did. A couple of other students came by and were included in the extended conversation over a very simple question on a simple screen. See Figures 4-7 for the screens they were looking at.

```
--------------------------------------------------------------------------
QUESTION #3                PERCENTAGE: 50%
--------------------------------------------------------------------------

I THINK I'LL WRITE THIS IN INK.
SOMEONE COULD ERASE IT IF I WRITE
IT ON THE BLACKBOARD.

       D)   IF I WRITE
       E)   IF I WROTE
       F)   IF I'D WRITE

==============================================

O.K.?  IF NOT, CHANGE IT.  THEN RETURN
```

Figure 4. Initial student response – option A.

```
I'M SORRY, BUT THE PRESENT TENSE ISN'T CORRECT
FOR THIS ANSWER

I'LL GIVE YOU ANOTHER CHANCE TO CHOOSE THE
CORRECT ANSWER
```

Figure 5. Computer response to the student's choice.

```
                HYPOTHETICAL
                ------------
                CONDITIONS
                ------------
      IF CLAUSE          MAIN CLAUSE
      ---------          ------------
IF.. (PAST TENSE    #
         VERB)      #       WOULD
                    #
IF...   COULD       #       COULD
                    #
IF...   WERE        #       MIGHT
                    #
                    #

                SPACE BAR
```

Figure 6. Help screen invoked

```
--------------------------------------------------------------------------------
QUESTION #3                PERCENTAGE: 50%
--------------------------------------------------------------------------------

I THINK I'LL WRITE THIS IN INK.
SOMEONE COULD ERASE IT IF I WROTE
IT ON THE BLACKBOARD.

       A)   IF I WRITE
       B)   IF I WROTE
       C)   IF I'D WRITE

=============================================

YOU'RE REALLY LEARNING, ALI.
```

Figure 7. Second try, with computer response

Again, not so much the program, more what you do with it.

Some of the earliest language software for microcomputers included simulations such as London Adventure and Lemonade Stand. London Adventure remains one of my favorite early CALL programs. In it, learners are told they have a few hours in London before their plane leaves and certain items that they are expected to buy to take home with them. The adventure comes in the users' interactions with various shopkeepers and other people they meet.

In Figure 8, Phillips (1986) describes this scene from London Adventure, where students want to buy a postcard from a kiosk. They are presented with the following linguistic choices, one at a time. The learner has to press a key to see each choice and can cycle through repeatedly.

1. Excuse me, where is Trafalgar Square?
2. I want this postcard.
3. Excuse me. I like this postcard.
4. I'd like this postcard, please.
5. I'd like these postcards, please.

Depending on their choice, one of the following responses is elicited [again, one choice at a time]:

1. "It's right outside the National Gallery.", but no other action is forthcoming.
2. The assistant ignores you and serves another customer.
3. "Yes. It is nice.", but no other action is forthcoming.
4. "That will be 25p, please.". followed by the sale of the postcard.
5. "I'm sorry. You can't have them all." No other action.

Figure 8. London Adventure (Phillips, 1986: 48).

As Phillips points out, this program could be used by an individual, but it would be better for group work because of the discussion possible. Presenting one option at a time also encourages learners to read and remember. Such focused engagement is very good for language learning.

CALL authors in the United Kingdom were generally more proactive than those in the US in terms of encouraging learner autonomy. Tim Johns was writing about "data-driven learning" from 1986 on, encouraging teachers to put language research tools into the hands of students. Wida's Authoring Suite was easy enough for students to use in creating their own exercises. Given the relative lack of enthusiasm for plain language drills in the UK, it should come as no surprise that the two simulations mentioned above, London Adventure and Lemonade Stand, both came from the British Council.

The major issue with software and pedagogy during this period was the frequent disconnect between what teachers wanted software to do and what they could do with it. There were two major factors involved. First, early software in general was written either by programmers who knew little or nothing about pedagogy, or by teachers who knew very little about programming. Second, lack of teacher training in CALL meant that most teachers used drills because they understood what drills were supposed to do. Sadly, not many teachers were as adept at creating learning spaces for students, using simulations, games, and text reconstruction programs and encouraging student autonomy.

Late 1980s to mid-1990s

As time went on, software added multimedia elements, and programmers and teachers got better at talking with each other. HyperCard let non-programmers create multimedia relatively easily. Since HyperCard also came for free with Macintosh computers, it was widely used in teaching. Of course, non-programmers were still not very good at creating sophisticated software, but they could do much more with HyperCard than with programming in BASIC. Software began using voice recognition so that students could sometimes speak their answers and have the computer respond, as with CPI's Traci Talk (which, while interesting at the time, did not work consistently and no longer exists today). Language teachers and learners started using simulations, games, word-processors, text reconstruction programs (e.g., cloze, whole-text deletion, jumbled sentences), and authoring tools more frequently. Multimedia that would appeal to different learning styles was a description for just about all software, even programs with nothing more than a few graphics.

Figure 9. New Dynamic English

Larger software packages, such as DynEd's Dynamic English (Figure 9), provided practice in a systematic way across a range of skill areas. These larger learning systems had a curriculum and often came with a scope and sequence of instruction. Localized DynEd products were sold with bilingual help files for the Spanish, Portuguese, Japanese, and other major EFL markets. Their markets were teachers and schools without an adequate background in English and/or language teaching. Such schools needed the comprehensive curriculum that

the large software product would give them.

The pedagogy had changed to be more learner-centered, and there were certainly more software options in that direction during this period. Teachers were frequently reminded that multimedia could appeal to learners with different learning styles by providing multiple forms of input, with text, audio, graphics, and video. Virtually all HyperCard stacks for language teaching included text and graphics, and many included audio as well. The comprehensive programs enabled teachers and schools to plug their language learners into the computer for a full range of language practice – for good or for ill. In the US, the Office of Technology Assessment (1987) found that far too many English language learners were being shunted off to computers, primarily to more basic drill and practice programs. Even though better programs existed, administrators with little background in language teaching and little knowledge of CALL were going back to the way they thought computers could be used in language teaching: for drills.

Mid-1990s to early 21st century

Email was available to many educators in the mid-1980s, primarily at universities. Command-line programs let users download software, participate in mailing lists and newsgroups with specific interests, and browse a limited set of online databases using Gopher. Email had become more robust by the mid-1990s, but it was not heavily used in teaching. One exception was Ruth Vilmi's Helsinki University of Technology (HUT) Email Writing Project, which connected teachers and learners globally in shared writing activities (see http://www.writeit.fi/ruth/publications/overview.html). The advent of Mosaic and the World Wide Web completely changed how teachers and learners used the Internet. Mosaic was a graphical browser (its successor was Netscape, and now Mozilla's Firefox). Even those who did not consider themselves computer experts could use the graphical interface to link to different websites. The content on those sites could finally include graphics, and later sound and video.

Teachers' interest in software faded as they saw free websites with interesting content. The web offered sites with texts and graphics to look at, plus links to other sites with text and graphics to look at. The first CALL sites catered to teachers and had useful links to suggestions for classroom activities and to content-rich sites. Dave's ESL Café (http://www.eslcafe.com) broke new ground with the Graffiti Wall, where students could submit comments and Dave would review and post them – one of the first truly interactive CALL sites. The ESL Café also boasted a variety of quizzes where learners could get immediate feedback. The quizzes were not organized by topic or level, and the feedback consisted of a checkmark for a correct answer and an X by an incorrect one. But they were free.

I liked the Graffiti Wall (see Figure 10) because it was a way for students to have a wide audience for their voices. The quizzes, on the other hand, seemed to be more basic than what was available in the mid-1980s. Even the earliest versions of Grammar Mastery asked the student to input his or her name, and then personalized the feedback. It also had a sequencing of exercises and help files – both important pedagogical features completely missing from web quizzes then. Even now, sequencing and help options are almost never seen in quizzes and exercises on the web.

Figure 10. Graffiti Wall (Internet Archive)

As the web grew, there was a blending of media and delivery systems. In the 1990s, textbooks occasionally came with an associated CD-ROM. Now, textbooks started coming with related websites that offered additional material for teaching, including lesson plans, extra readings, more discussion questions, and the like. The websites for textbooks were largely static collections of information. More whole courses started appearing online. These included both free sites subsidized by heavy advertising and other courses for a fee. Some software vendors began linking their software to the Internet. Rather than buying a CD-ROM outright, schools could purchase an annual subscription that provided frequent updates to the software, record-keeping online, web activities, and other add-ons. The steady flow of revenue wasn't bad for the publisher, either.

Another benefit of the web was that it could eliminate the platform wars between Macintosh and Windows users. Publishers who used the web as a delivery system only needed to create one version of their material. Sadly, that single online version often neglected sound and video as well as other dynamic content when it shifted to the web. Web developers started using more Java, Flash, and other technologies that allowed dynamic content, but they were limited by the bandwidth (Internet speed) that their users were likely to have. Charles and Lawrence Kelly used better web tools and started producing some varied web exercises, now found at Interesting Things for ESL/EFL Students (http://www.manythings.org/) and among the exercises at Activities for ESL Students (http://a4esl.org).

A solution to the issue of dynamic content and online speed came with software like Hot Potatoes (Figure 11), an easy authoring tool designed by English language teachers for language teachers (http://hotpot.uvic.ca). Having downloaded the software, teachers could use it to generate cloze, matching, multiple-choice, and scrambled sentence exercises and crossword puzzles. These could be uploaded to the web (with some difficulty initially), used offline, or printed out to use in class. Like other authoring programs, Hot Potatoes allowed teachers to add graphics and later sound and video, as well as to give feedback on learner choices and multiple correct answers in the cloze and multiple choice questions – if teachers took the time to enable these features.

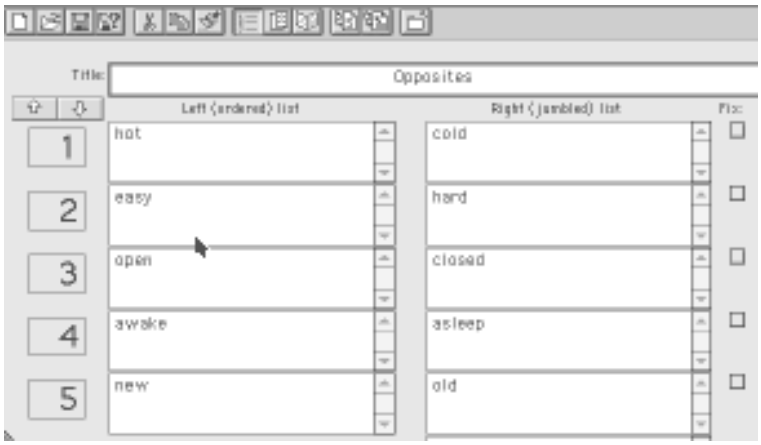

Figure 11. Hot Potatoes JMatch exercise

The shift to the Internet meant that technology provided both less and more than before. Online quizzes tended to be simplistic, hearkening back to "drill and kill" days, largely because they were written by teachers who were not programmers or programmers who were not teachers. Language teaching websites were collections of links in some order invisible to the user. Lack of structure and sequencing made these sites not very useful for independent study by learners – nor quick review and use by teachers.

The strength and richness of the Internet lay elsewhere: in its ability to link people to information and to each other, finally opening up far more opportunities for real, person-to-person, communicative learning options. These options included keypals (electronic penpals), either one-to-one or class-to-class; and tandem learning, where those wanting to learn one language interacted with a native speaker of that language who wanted to learn theirs. Instant messaging was also available, but tended to be less used for language teaching because of bandwidth and lack of structure. Teachers who wanted to use these communication possibilities in their classes needed to do a certain amount of work to set up pairings and create projects. The projects that resulted could be far-reaching, however, such as the global activities set up by iEARN (http://iearn.org/).

Today – Web 2.0 and ubiquitous computing

Computers now come with gigabytes of memory and hundreds of gigabytes of disk storage space. Internet speeds are increasing dramatically. While not everyone has high-speed capability, YouTube (http://www.youtube.com) has been a factor in wider access to video for people around the world. Individuals can set up audio and video chats using Skype (http://www.skype.com), Yahoo Messenger (http://messenger.yahoo.com/), and other free tools that connect people. In addition, software and the web are merging. For example, Google Docs (http://docs.google.com/) allows people to do word-processing and share it online so that multiple people can be editing the same document despite being half a world apart. Google Docs is just one instance of what is being called Web 2.0, or the social web. Other examples are podcasts, blogs, wikis, social networks such as Facebook and Ning, and virtual worlds such as Second Life. These are ways for people to interact with and change web content, not just look and click. The web has become much more like a classroom where learners can interact with information and with other people in real time or asynchronously.

Another trend with a major impact on teaching is ubiquitous computing. This means that technology is everywhere, part of everything. Cell phones act like computers, reading email and delivering language lessons with text and audio. A public display can respond to an individual's spoken request for information. Microtransmitters will send information about what specials are available in a restaurant as someone passes by. We will interact with our

world in different ways, and every place and every thing may be able to deliver information to us.

Recap and overview

The period of CALL use from the 1960s through the late 1970s is what Mark Warschauer (1996) refers to as the "behavioristic CALL" phase. He feels that the computer was primarily seen as a tutor and used in ways that mirrored behaviorist views of people and learning – repetition was the key. But even PLATO was a bit more nuanced than simple drill and practice, especially considering what most web quizzes still look like. PLATO's Wide Area Network linked its dispersed community of teacher-developers, presaging later Internet capabilities.

Warschauer (1996) feels that CALL entered a "communicative" phase with the advent of microcomputers. They offered more than drills, including word-processing, simulations, and other activities. Still, there was a focus on discrete skills with most software, as with most language teaching. Warschauer notes that both with and without computers, "a number of educators were seeking ways to teach in a more integrative manner, for example using task- or project-based approaches" (Communicative CALL subhead, ¶9.)

The current "integrative" phase began with the introduction of multimedia and then the Internet. This phase, according to Warschauer (1996), allows teachers to address integrated skills, offer more learner control, and create more authentic learning spaces. The need for this type of CALL is expressed even more strongly in Warschauer and Healey (1998): "The multimedia networked computer—with a range of informational, communicative, and publishing tools now potentially at the fingertips of every student—provides not only the possibilities for much more integrated uses of technology, but also the imperative for such use, as learning to read, write, and communicate via computer has become an essential feature of modern life in the developed world" (¶7).

Stephen Bax (2003) takes a somewhat different view of the phases, offering three categories: restricted, open, and integrated. Restricted CALL represents early uses of computers in language teaching, both on mainframes and early microcomputers. Bax points to the varied uses of CALL during this period and rejects the term behavioristic. Usage was restricted by limitations in hardware and software, but was more than simple repetitive drills. We are currently in the open CALL phase, according to Bax. Much more is possible than before, and software provides a range of activities and feedback.

Bax refers to the final stage, integrated CALL, also as "normalized" CALL. We are not there yet. In this stage, there will not be "CALL" because technology use will simply be taken for granted, part of the background in learning. The technology used in teaching and learning will be no more exciting or fear-invoking than shoes or writing.

The changes in technology have been dramatic over the past 40-plus years. The changes in language teaching have also been far-reaching, at least in many parts of the world. That learners should help direct their own learning is as great a change from teacher-centered practice as the Internet is from repetitive drilling. The changes have often complemented each other, especially in the hands of skilled teachers. In an age where learners can find answers on their own easily, it is possible for them to point out errors that their teachers may make. They can use an online dictionary, such as Merriam-Webster (http://www.merriam-

webster.com/) to get multiple meanings of an unknown word quickly and to hear the pronunciation as spoken by one or more native speakers. They can use the Cobuild Concordance Sampler (located at http://www.collins.co.uk/Corpus/CorpusSearch.aspx) to show that a word is not used in the way that the teacher has used it. A teacher who pretends to know everything is at risk of being shown to be ignorant in many areas. Teachers need to give up the illusion of having total control over their students' learning and instead move in the direction of creating effective learning spaces. Technology – software and the Internet – can open many doors to project- and task-based learning, where learners can acquire language in an information-rich environment.

We have come a long way from the early days of limited capabilities and excessive drilling. It has become a very interesting time to be a technology-using teacher. When every place and every thing gives us information, what will language teaching and learning look like? In a world overflowing with data it may be more important than ever to have a teacher who can sift through the information, organize it, and present it at a level and in a way that learners can understand. As ever, it's not so much the program/website/information delivery device, more what you do with it.

References

Bax, S. (2003). CALL - past, present and future, *System, 31* (1), 13-28.
Daiute, C. (1985) *Writing with Computers.* Reading, MA: Addison-Wesley.
Davies G., Walker R., Rendall H. & Hewer S. (2009) Introduction to Computer Assisted Language Learning (CALL). Module 1.4. In G. Davies (Ed.) Information and Communications Technology for Language Teachers (ICT4LT), Slough, Thames Valley University. Accessed April 1, 2009 at http://www.ict4lt.org/en/en_mod1-4.htm
Healey, D. (1995). *Something to do on Tuesday*. Houston, TX: Athelstan.
Higgins, J. (1987). *Language, learners, and computers: Human intelligence and artificial unintelligence*. London: Longman.
Johns, T. (1986). Microconcord: A language-learner's research tool. *System, 14* (2), 151-162.
Jones, C. (1986). It's not so much the program, more what you do with it: The importance of methodology in CALL. *System, 14* (2), 171-178.
Office of Technology Assessment, U.S. Congress. (1987). *Trends and status of computers in schools: Use in Chapter 1 programs and use with limited English proficient students*. Silver Spring, MD: National Clearinghouse for Bilingual Education.
Phillips, M. (1986). *Communicative language learning and the microcomputer.* London: British Council.
Taylor, M. & Perez, L. (1989). *Something to do on Monday*. Houston, TX: Athelstan.
Warschauer, M. (1996). Computer-assisted language learning: An introduction. In S. Fotos (Ed.), Multimedia language teaching (pp. 3-20). Tokyo: Logos International. Accessed April 1, 2009 at http://www.ict4lt.org/en/warschauer.htm
Warschauer, M., & Healey, D. (1998). Computers and language learning: An overview. *Language Teaching, 31*, 57-71. Accessed April 1, 2009 at http://www.gse.uci.edu/person/markw/overview.html

Sources for figures

Figure 1 (PLATO terminal) from http://en.wikipedia.org/wiki/PLATO_system
Figure 2 (BBC Micro) from http://en.wikipedia.org/wiki/BBC_Micro
Figure 3 (Apple IIe) from http://oldcomputers.net/pics/appleii-system.jpg
Figure 9 New Dynamic English from http://www.dyned.com/products/nde/

Figure 10 Graffiti Wall – archive – from Internet Archive at http://web.archive.org/web/19971211085601/http://www.pacificnet.net/~sperling/wall.html

Bio data

Deborah Healey (University of Oregon) has been teaching with technology for 25 years. She is a co-editor of *ORTESOL Journal*, former editor of *CAELL Journal*, and former Chair of TESOL's CALL Interest Section. She frequently consults, presents, and writes about technology in education. Her doctorate is in computers in education. Her website is http://www.deborahhealey.com and she can be reached at dhealey@uoregon.edu.