Review of Developments in Computer Assisted Language Learning

Wayne E. Arnold

Abstract

This paper will provide an overview of the broad spectrum of information concerning Computer Assisted Language Learning (CALL) and focus attention on two related areas: historical expansion of computer language programs through innovational progress and the still-present programming needs of CALL applications for further improvement of digital language resources. The rapid expansion of the internet has shifted the direction of CALL research so that now new designs are taking to task the benefits of implementing websites, online wikis, and chat programs. I explain why the efforts of CALL programmers have shifted to construct online environments geared to help students overcome many of the initial inhibitions which occur during the SLA process. Finally, due to the multiplicity of platforms available, there is an increasing demand for standards across the different CALL applications.

Keywords: Second Language Acquisition (SLA), Computer Assisted Language Learning (CALL), Natural Language Processing (NLP), End-User-Development, computer language programs, digital language resources
Review of Developments in Computer Assisted Language Learning

Throughout the evolution of Computer-Assisted Language Learning (CALL), technology has remained the driving factor with software lagging behind equipment development, much to the frustration of the language teacher and learner. The task of both the teacher and programmer has been to incorporate CALL programs with linguistic functionality involving Second Language Acquisition (SLA). The disparity between learner requirements, technology, and the programming essential to joining the two has progressively decreased in recent years; nonetheless, there are still many theoretical questions that have yet to meet the needs of SLA. This paper will provide a summary of the broad spectrum of information concerning CALL and focus attention on two related areas: historical expansion of computer language programs through innovational progress and the still-present programming demands of CALL applications to advance the incorporation of improved digital resources. The greatest issue within these two areas is apparent inability to create a unified approach to construct language programs. While there is a consciousness and firm agreement for the continued integration of research to enhance the evolving program formats, to what extent and depth language teachers and end-users should be engaged in the behind-the-scenes modifications is still tending.

History

Computer-Assisted Language Learning has a rather extended background. Originally envisioned in the 1950s as a possible method for formalizing a standard language learning format, as technology improved, in the 1960s and 70s programmers began incorporating the language learning concepts of the time period, particularly the behaviorist theory (Warschauer, 1996). In
addition to the behaviorist methods, during this era there was also “rapid growth in language labs, bolstered by the then fashionable audiolingual approach to language teaching” (Davies, 1997). Both methods, however, fell out of favor with the continually developing linguistic approaches. With the invention of the microcomputer, the communicative phase of CALL was introduced, promoting actual patterns of communication as the “[p]roponents of this approach felt that the drill and practice programs of the previous decade did not allow enough authentic communication to be of much value” (Warschauer, 1996). More recently, language acquisition research has become one of the fundamentals behind the shift to communicative CALL, since this method “reflects a well-informed pedagogical practice due to its strong theoretical base” (Salaberry, 1996, p. 8), requiring linguists to apply SLA theory to CALL applications in order to be more successful. In addition to these developments, one more phase that would bring CALL up to its present status of acceptance across language departments (yet still causing confusion as to which programs are best) is the multimedia and Internet expansion taking place in the past fifteen years.

During the early 1990s, researchers set out to respond to the following three questions concerning overall effectiveness of CALL programs in classrooms: “Do students like it? Do students use it? Does it work?” (Chapelle & Jamieson, 1986, p. 27). Prior to this, there were attempts to arrive at a consensus regarding the functionality of in-classroom computers. The pros and cons were intertwined, causing difficulty in applying the research. For instance, there was some concern over student ability to explore and experiment with programs, allowing them to
initiate a personal educational experience. Research indicates that a negative result of this feature arises in evaluating student success and explaining student failure. One advantage is that students are able to store their results—for example, grammar grades or paragraph writing—and student errors can be gathered, evaluated, and then, if the program is advanced enough, particular lessons to strengthen student weaknesses can be assigned (Chapelle & Jamieson, 1986, p. 29). Overall, though answers to the researchers’ questions were often ambiguous, there was a general positive attitude towards the integration of computers into the realm of language education. What eventually became apparent was the need for functionality in CALL applications, encouraging student involvement.

A resource propelling CALL acceptance in the early stages was the language conference. In the early 1980s, Roger Kenner, head of Concordia University’s Learning Laboratories, and David Sanders, a professor of methodology at Concordia, approached the Teachers of English to Speakers of Other Language (TESOL) 1982 Conference with a proposal to query students’ perception of subjects involving computers and their input regarding the necessity for improvements. Their proposal incited much interest in the TESOL conference participants so that by the next year—at the TESOL Toronto Convention—Kenner relates, “[a]fter several years of very rapid, haphazard development, things in the field of CALL were rapidly coming together […]. The first books on the subject appeared. Microcomputer platforms standardized. The first acceptable commercial software began to appear. Learned journals began publishing” (2009). Additionally, at the Toronto conference, Kenner began to network further with professionals interested in the budding field of CALL and was invited to speak on the benefits of computers in the classroom.
As Kenner suggests, pivotal moments in early CALL support revolve around the years of 1982 and 1983. One of the first articles to utilize the term CALL was published in *Educational Computing*, in October 1982, written by Graham Davies and David Steel. Entitled “First steps in Computer-Assisted Language Learning at Ealing College of Higher Education,” Davies and Steel expressed concern that language learning with computers “has so far failed to make much of an impression on language teachers” (1981). Davies and Steel took efforts to lean BASIC programming on a Prime 300 minicomputer in order to originate some of the first SLA applications for the computer. From our current vantage point with modern computers, we can appreciate how far CALL applications have developed over the years, thanks to the efforts and research performed by those who were certainly cutting-edge language teachers of the late 70s and early 80s. The significance of the Davies and Steel article is that, with sufficient determination, teachers could be encouraged to consider—in a positive light—the advancing improvements in computers for supporting language learning. Detailed in their description of both the tools utilized (BASIC, GDTEST) and the hardware used (Prime 300, EC10), Davies and Steel explain some their own limitations in the new realm of computers and language programs: “[d]epending on the size of programs being used, 3-4 students can use the EC10 at once. Most programs require 5K to 8K of memory, but with only 24K of memory shared between all users there are obvious restrictions” (1981). I include this seemingly out-dated bit of information from the Davies and Steel article to draw attention to the early CALL articles and to highlight the close integration of microcomputer terminology of programming/technology with language applications. As the field would burgeon over the next thirty years, there is a significant shift in research articles between hardware/software and language applications and to what depth their
content delves into computer programming jargon—many articles began to merely discuss the theoretical applications of CALL rather than specific programs and code. Interestingly, fifteen years later, Davies would acknowledge that “[e]arly CALL programs also owed much to programmed learning. I confess to having been initially fascinated by the idea of the computer reacting to the learner’s input and branching accordingly, and I admit to having written many CALL routines that were unquestionably behaviouristic” (Davies, 1997). By behavioristic, Davies is referring to the limitations of the design within his own programs. These types of limitations would eventually lead researchers to realize that to be beneficial, CALL programs needed to go beyond behavioral theories to encompass the full spectrum of methods by which humans learn languages.

From the earliest of computing days, mainframes and microcomputers were applied to design language learning tasks, but not without complications. In his 1987 article, “Advanced Technology in Foreign Language Instruction and Translations,” John Fought explains that “instructional systems” were designed for mainframes such as PLATO, available for both mainframes and microcomputers and designed by the University of Illinois. However, Fought notes, “[t]he difficulty with these systems in their original form is their high cost per user, especially high in the case of Plato, with its requirement of specific hardware, including very expensive terminals” (1987, p. 36). Mainframe systems, in particular, designed in the 1960s and then gaining ground in the 1970s, limited diversity through the costly equipment, and even more so, demanding increasing man-hours to design the complex programming; yet, the ability to innovate language lessons proved valuable and the ease of the microcomputer eventually replaced any role the mainframe computer might serve.
programming; yet, the ability to innovate language lessons proved valuable and the ease of the microcomputer eventually replaced any role the mainframe computer might serve. In his article, Fought reviews previous technology (before microcomputers) in a section entitled “Lessons from Yesterday’s Advanced Technologies,” and it is worth noting that some (but not all) fundamental aspects of technology and language learning have not altered, merely modifying with the times. Fought summarizes the fundamental changes in language education technology:

- Audiovisual aids: record players, films and transparency projectors function in a “supplementary role.”
- Language laboratories: initiated in the 1950s, these progressed into lab areas where languages could be practiced in a controlled environment. Labs tend to be technology driven and publishers quickly followed the hardware advancements with disks, films, magnetic media, and textbooks all geared around the environment of the language laboratory.
- Teaching machines: now a defunct technology, these were originally machines that could be programmed at a low-level of complexity and implement looping of audio tracks that were programmable; however, Fought notes that “teaching machines apparently produced often violent aversion in teachers and learners alike. If language labs often made students feel like inept parrots, teaching machines made them feel even worse—like incompetent learning machines” (1987, p. 39).
- Computers and translation: early prospects (1940s) for computer aided translation looked bleak due to the cost of equipment and inefficiency of results. Yet, with technological
advancements there is a large push in research, funded by the United States, Europe, and Japan.

Considering these points from a 1987 perspective, it is clear that CALL applications were steadily gaining ground but were far from the now-expected computer capabilities. Other articles, such as “Interface Design for Computer-based Learning Environments” (Jones & Okey, 1995), reviewed research for early computer programs recommending that programmers provide such options as searchability, progress markers, and use location indicators, which, from a present day standpoint, seem to be obvious standards for programs. And even more recently, computer-based testing has been evaluated as weak in the area of listening testing, failing to incorporate methods which are typically performed with paper and pencil (Okey, 2009, p. 843).

A fundamental contribution to the field of computers and language learning came in 1991 from Cornell University. The Department of Modern Languages and Linguistics professor Nina Garrett published “Technology in the Service of Language Learning: Trends and Issues,” which quickly established itself as a landmark article. Garrett’s objective was to engage teachers who were not yet utilizing computers in their language classroom and also to advise teachers on new practices by detailing the available resources at their disposal. Authentic video, SCOLA (Satellite Communications for Learning Worldwide), and video recordings, for example, are discussed in great detail. Computers are essential, Garrett argues, but there arises confusion over which system to purchase. In addition, Garrett encouraged the integration of computers into the classroom while keeping an eye on developing research surrounding CALL: “the author would...the full benefits of CALL will not be realized until its use is fully integrated with classroom work on the basis of theoretically motivated research on the kinds of learning activities most enhanced by technology and those best undertaken without it.
argue that the full benefits of CALL will not be realized until its use is fully integrated with classroom work on the basis of theoretically motivated research on the kinds of learning activities most enhanced by technology and those best undertaken without it” (2009a, p. 702).

Essentially, Garrett’s article became the “go-to” source for both new and experienced teachers, giving a run-down for the “four skills” governing philosophy of language learning: speaking, reading comprehension, listening practice, and writing. For each skill, Garrett provides an overview of the way CALL programs assist in these areas as well as what universities are working on developing programs addressing these skill-areas.

To attest to the significance of Garrett’s article, the Winter 2009 supplemental issue of Modern Language Journal devoted the entire publication to “Technology in the Service of Language Learning: Update on Garrett (1991) Trends and Issues.” Garrett includes an updated survey intended to bring abreast the advancements gained in the field during the last 18 years since her initial article. A clear shift in her approach to CALL applications has occurred over the years: “My 1991 title stressed the primacy of pedagogy over technology; today, by contrast, I want to emphasize that none of the three major components of CALL—pedagogy, theory, or technology—should dominate the others” (Garrett, 2009b, p. 720). In addition to the equalizing of these three areas, Garrett adds infrastructure to her list of valuable considerations for language departments. For Garrett, infrastructure is composed of “physical/technological setup of our teaching and learning spaces, […] institutional professional development support structure for technology use, […] and a […] national structure of language education and the national support structure for it” (2009b, p. 720). Computer infrastructure has become essential because many universities are unaware that the technological needs of language labs differ significantly from
general computer labs (where students are expected to work silently). The dissimilarity between the language students and the typical student is that the student using the language software needs to be able to hear and speak their target language. Over the years, the shift of language labs into the hands of IT technicians moves the technological infrastructure away from those aware of how the advances—created with an eye to the linguistic research—aids students in learning languages.

As CALL applications have increased, research directions have changed. More recent investigation targets the presence of CALL in the classroom and its influence on students. These studies have expanded their focus in order to observe what Román-Odio and Hartlaub have referred to as cognitive-oriented questions, such as: “learners’ attributes and aptitudes,” “mental processes activated and cultivated as a result of CALL instruction,” “specific aspects of L2 competence in relation to CALL lesson type,” and “social effects of CALL (single-student tutorial, couple-group networking) on the quality and quantity of language production” (2003, p. 593). As these topics suggest, the attention to independent variables has been the result of shifts in the applications of CALL programs, and research links directly with the development of the programming aspects of emerging software. There is a growing awareness that “CALL should be integrated into the traditional classrooms where the instructor is also available for further assistance and questions so that students are not deprived of human contact” (Veena & Krishna Kumar, 2012). Román-Odio and Hartlaub have argued that researchers will most benefit CALL

Garrett believes that social networking venues such as Facebook and virtual realms like World of Warcraft should be evaluated as possible language activities: “To engage students in meaningful communicative practice that reflects what they do in the natural world, language professionals must leverage this interest in social computing and consider incorporating some of those activities into the L2 curriculum”
applications by extensively studying the following: lesson type, student characteristics, the effect of formal variables on student learning (including feedback), and educational outcomes of CALL (2003, pp. 593–594). It remains true that, concerning feedback, “CALL can assist the structure (grammar) lessons of the language classroom since it enables learners to get immediate feedback” (Veena & Krishna Kumar, 2012); such immediate feedback encourages the continuing integration of CALL applications in the classroom. With the increase of diverse environments in which CALL programs are making an appearance, various new aspects of research like the ones just listed are in need of further evaluation and would be beneficial projects for expanding applied CALL theory.

In the past fifteen years, a rapidly developing arena for CALL applications has been the Internet, or social computing. This virtual medium has assisted the additional acceptance of CALL in the classroom. It is not without certain limitations, however, as Davies notes, “[t]here are very few websites that offer the learner the possibility of recording and playing back his/her own voice, a popular key activity among both teachers and learners that has been possible since the widespread use of the tape recorder in the 1950s” (2005). Garrett believes that social networking venues such as Facebook and virtual realms like World of Warcraft should be evaluated as possible language activities: “To engage students in meaningful communicative practice that reflects what they do in the natural world, language professionals must leverage this interest in social computing and consider incorporating some of those activities into the L2 curriculum” (2009b, p. 732). These social media networks allow “students [to] engage in authentic discursive practices related to their disciplines in unprecedented ways” (Arnó-Macià, 2012, p. 93). When internet implementation is introduced, there is a significant impact on the
cultural and social aspect of second language learning. The integration of the internet into a language classroom, whether in a CALL application or a social media, assists teachers in develop[ing] intercultural competence through telecolaborative projects with people from other cultures, who can provide valuable feedback for the development of learners’ language (including pragmatics) as well as potential social relationships. In doing so, the pedagogy raises issues of intercultural tensions, negotiation of language and identity, as well as social and institutional contributions and constraints (Chapelle, 2009, p. 747).

Rather than complicating the language environment, such venues bring to the forefront one of the more fundamental aspirations of learning a foreign language: to communicate with people from different cultures. In one study examining how students create social identities on the internet, the author noted that “a prominent aspect of Internet-based communication is the use of textual and other semiotic tools to create communal affiliations and construct social roles and narrative representations of self” (Lam, 2000, p. 477). The rapid expansion of the internet has quickly shifted the direction of CALL research. Research is now taking to task the benefits of implementing websites, online wikis, and chat programs. As will be elaborated upon towards the end of the next section, online environments help students overcome many of the initial inhibitions which occur during the SLA process. Furthermore, numerous longstanding CALL applications, such as language translation and audiovisual aids, are now readily available.

In one study examining how students create social identities on the internet, the author noted that “a prominent aspect of Internet-based communication is the use of textual and other semiotic tools to create communal affiliations and construct social roles and narrative representations of self” (Lam, 2000, p. 477).
online. These rapid advancements bring to question how quickly researchers can evaluate the effectiveness of CALL applications, but also how integrated linguistic knowledge is correctly applied to the programming of these new online features.

**Programming**

With rapid forward movement in technology, programming has continued to be a source of frustration for implementers of CALL programs. Garrett draws attention to a significant disparity between hardware and software, with programming lagging far behind hardware advancements, as “[t]hose who need to decide on major hardware purchases are usually advised to consider first the software they want to run and buy the machines that run it. In general that is good advice, but in education (perhaps especially in language education) the existing software does not yet adequately represent the capabilities of the current hardware” (Garrett, 2009a, p. 701). At present, programming still lags behind hardware. Garrett’s new recommendations is to focus interest in CALL programming on “tutorial, engagement with authentic materials, and communication” (2009b, p. 722). This shift moves away from the issue of hardware vs. software advancement and toward the application of pedagogical and linguistic theory.

Complexity in computer programs builds resistance between teaching and learning. The crucial points in CALL pedagogy addressed by Garrett have also been voiced from other arenas as well: “Teacher educators need to design CALL courses that teach what language teachers really need to know” (Egbert, Paulus, & Nakamichi, 2002, p. 109). These researches also observe that many language teachers feel unskilled in computer technology, thereby hindering their ability to present convoluted or complex computer language programs to students. There are a plurality of influences that prevent teachers from integrating technology into the classroom,
including, but not limited to, time pressures, technology-related anxiety, lack of resources, inflexible guidelines, poor technical support, age, and the rapidity at which technology changes (Egbert et al., 2002, p. 112). Time is the one element most attributed to by teachers for avoiding CALL activities in a classroom. Otto and Pusack have provided what they believe to be six attributes of successful CALL properties to be considered by program designers, these include: 1) Suitability, or meeting the student and course expectations, 2) Interactivity, helping students become involved through challenging and motivating exercises, 3) Media, a growing field of web resources and support resources, 4) Record keeping, in order to track student progress and highlight weaknesses, 5) Ease of use, neither teachers nor students have the time to learn convoluted programs, 6) Accessibility, materials should be deliverable through the web in order for students to access lessons from remote locations (Otto & Pusack, 2009, p. 786).

As with the incorporation of computer assistance into the language learning spectrum, so too has the programming complexity developed. In the 1970s and 80s, the idea was that in order “[t]o provide an individualized learning environment, many developers have used a systems approach to design: A learning hierarchy is formulated, and a diagnostic mechanism is used so that either the computer program or the student can decide when the student needs to review” (Chapelle & Jamieson, 1986, p. 28). These early concerns revolved around the intent of individuality in students and that the person designing these lessons needed to understand how students acquired languages. Another key element was to attempt to design programs that would be appealing to students, as failing to do so could greatly influence the effectiveness and the
overall outcome of the CALL program. Further development, however, emphasizes the types of interfaces with which a variety of students can interact and learn in their own particular manner. Thus, much research focuses on the CALL program layouts with the intent to support the language learning needs of students. From the 1990s onward, a major push for Design-Based Research (DBR) has occurred, which is “a progression from the study of learning, teaching, and assessment as individual stages of the learning process, to a systemic whole that embodies all three” (Pardo-Ballester & Rodríguez, 2009, p. 86). For CALL purposes, DBR serves as an efficient method of incorporating research, instructional materials, and goals for learners. Pardo-Ballester and Rodríguez provide an example of a Spanish hybrid course in which students from diverse professions study the terminology which is suited for their profession. The objective of this DBR course design was to engage students with vocabulary related to their field of study. Division of vocabulary relating to profession was created through a digital program which focused on student interests. In this instance, combining traditional instruction, interaction with the teacher through the internet, and the field specific multimedia interaction, the hybrid design provides students the flexibility they want as well as the specific language training they need (Pardo-Ballester & Rodríguez, 2009).

Considering the example discussed in Pardo-Ballester and Rodríguez, blended learning is a method by which designers of CALL programs have realized an important element when programming the design of a language application. Further designing of CALL programs need to abreast of the trends that are being generated in the social media realms as “we are moving away from a passive information age towards an active participation age”

…three hypotheses for Design-Based Research, 1) “Target input needs to be made salient,” 2) “Learners need help comprehending linguistic input,” 3) “Providing opportunities to produce target language”
(Farmer & Gruba, 2006, p. 149), in part due to technology advancements. Pardo-Ballester and Rodriguez have narrowed their review of second language acquisition research into three hypotheses for DBR, 1) “Target input needs to be made salient,” 2) “Learners need help comprehending linguistic input,” 3) “Providing opportunities to produce target language” (2009, pp. 93–94). All three of these inputs—saliency, comprehension, production—are now fairly understood to be requisite elements in CALL programming. Fraser expresses concern for the programming of CALL applications to stimulate students to use these electronic resources. Writing significantly before the hypotheses of Pardo-Ballester and Rodriguez, Fraser realized that “[w]ithout specific goals or tasks to motivate the browsing, the materials I had assembled could not enrich the students’ reading and understanding” (1993, p. 128). Therefore, it is essential that programmers anticipate more thoroughly their audience by furnishing not only the environment to enable learning, but to provide exercises which will entice participation.

Along with Design-Based Research, there is also End-User Development (EUD) that entails “the active participation of end-users, as nonprofessional developers, in the software development life cycle” (Farmer & Gruba, 2006, p. 149). This differs significantly from DBR programming in that EUD developed programs take into consideration the feedback from active participants (teacher and language students) during the actual programming process. However, this development method does not intend for the nonprofessional (i.e. a non-programmer) to be involved in the nitty-gritty details, rather the “end-users should focus their attention on gathering correct requirements and on efficiently transforming these requirements into high-level conceptual designs” (Farmer & Gruba, 2006, p. 152). These requirements, as such, can be
assembled from current research and professional classroom experience. Farmer and Gruba provide a useful visual of the interaction between the software engineers and the end-user.

Complications occur, however, in this method of interaction due to the responsibility placed on the end user. Concerns arise over costs, capability, responsibility, and a number of other technical and legal issues which add a level of difficulty to the implementation of EUD projects (Farmer & Gruba, 2006, p. 153). However, many of the issues over arriving at CALL programs that meet SLA needs could be designed through EUD methodology. Again, the problem arises between the rigid nature of engineering programming and the organic and diverse nature of language acquisition. That linguists are still attempting to grasp the complex processes of the mind—and all the variations of learning styles—tends to make the cooperation between the programming required for EUD and a functional CALL program very complex.

Both EUD and DBR have valid arguments for their distinct approaches to programming for SLA, but the industry has been unable to fulfill the spectrum of demands. To be capable of pinpointing certain theoretical advancements that benefit the overall outcome of CALL is the only substantial ground on which researchers can agree is necessary. This difficulty in agreement is primarily because “[d]evelopment is neither strictly bottom–up nor top–down but both: It is an act of examining the potential of technology through the lens of SLA theory” (Otto & Pusack, 2009, p. 786). The mixing of technology and theory complicates these applications because SLA is also part of a greater social and cultural setting. The very nature of these environments means
that planning and designing is extremely complex and far from encompassing all aspects of SLA. Farmer and Gruba further explain that “[r]equirements in CALL are socio-technical, and are therefore ill defined and multifaceted. Problems with language learning are seldom discrete and binary, making design decisions extremely hard to plan and implement” (2006, p. 155).

Unfortunate for past progress, there are a number of researchers who feel that the rapid promotion in CALL programming occasionally results in the abandonment of previous computer-aided language advancements. In order to create the latest and most sleek program, designers are often displacing valuable language tools and some CALL developments are tossed away regardless of the fact that these features offer viable solutions to SLA program issues. One proponent for reclaiming past CALL progresses has been Mike Levy, professor at The University of Queensland, and a renowned scholar in the realm of CALL research. Claire Kennedy and Levy have voiced the need “for more emphasis to be placed on the need to build on and learn from past experience, and attention to what we refer to as sustainability in CALL” (2009, p. 446).iii Incorporation of past experience (to a greater degree), as Levy would argue, may assist with the difficulties which Farmer and Gruba note in the multifaceted nature of SLA design.

Incorporating foundational language learning elements into technology is crucial for a basic universal CALL platform. Similar to Levy’s argument, Garrett argues that “[w]e need to develop templates and models for translation and interpretation courses to be used by language programs across the country” (2009b, p. 729).iv These templates are possible since “CALL is more applicable to certain areas of language learning than others. All disciplines have areas where rote learning is essential, where there is no basis for discussion or interpretation, and basic
language grammar is one of these” (Cushion, 2005, p. 276). Repetitive methods for learning vocabulary and basic grammar functions are aided by CALL programs through the integration of grammar models, potentially creating a unified format across the gamut of CALL programs. The incorporation of these basic unified rote methods helps reduce production time for the programmers but also reduces the time required for teachers to learn new computer functions. Every time a new product is released there is a learning curve on the part of the language laboratory instructors or classroom teachers who are meant to implement the program. Paul Bangs and Lesley Shield, in “Why Change Authors into Programmers?” remind us that there is a dearth of commercial programs which can implement both the pedagogical and the complexity of a language. One of the primary financial problems in designing the ultimate CALL application rests with the consumer market: “[c]ommercial publishers are very willing to ensure the academic credibility of materials on offer, and often collaborate with educational institutions. However, the marketing imperative inevitably dominates the situation, leading to a concentration on mass market potential—lower levels, domestic appeal, major languages, etc.” (Bangs & Shield, 1999, p. 19). Purchase of CALL systems is often restricted by educational funds and limiting a schools ability to purchase a fully loaded program with all the advancements of CALL technology. If, however, more universal platforms were adopted in the design of CALL applications, this might result in reduced financial overhead of the programming.

An intriguing challenge for computer programmers lies in the realm of AI (Artificial Intelligence). Phonological limitation of computer technology prevents CALL programs from
producing dialogue at the fluency level of a native speaker. M. Rafael Salaberry, of Cornell University, addresses the issue of ICALL, or Intelligent CALL, arguing that “[i]ntelligent machine tutors that teach language require the following base components: a representation of the subject knowledge, a model of the learner, a language understanding system, and a means of selfadjustment, i.e., learning from experience” (Salaberry, 1996, p. 10). Interaction and involvement of students within the CALL framework establishes the distinction between effective programming and merely trying to convey information. Within ICALL, the difficulty arises when “machine tutoring systems resides on the fact that they constitute a particular type of individualized instruction which is cost-effective even if not as pedagogically effective as the human counterpart. [...] machine tutors cannot achieve the same level of responsiveness to the learning situations as humans” (Salaberry, 1996, p. 12). This area of ICALL is one in which, if progress were made, the functionality of language programs would more likely entice more users to participate in CALL programs.

Another term which is closely associated with ICALL is Natural Language Processing (NLP), which Elena Cotos believes “is capable of diagnosing errors, providing detailed explanations about the nature of those errors, and responding simultaneously to more than one problematic aspect of language use that may occur in learner’s production” (2009, p. 104). This description is idealized since these are the theoretical properties of NLP applications in ICALL programs. As of 2009, Cotos emphasizes the flaws in the NLP technology which hinders its successful incorporation into the language classroom. As with many of these advanced product designs, research is limited and how well they benefit language education and feedback mechanisms remains unclear. A crucial flaw in past NLP products has been that “such systems
[are] often theoretically uninformed in their design;” additionally, “they also have weak empirical support. To date, research on automated systems has largely been funded and carried out by the companies that have produced these commercial products” (Cotos, 2009, p. 117).

Online forums have been gradually implemented in the classroom and future CALL programs will undoubtedly utilize chat environments to support language learners with native speakers. Wiki pages are an option in providing students with their own mediums for inspiring communication and learning environments: “Wiki technology provides for the easy creation and editing of pages by students collaboratively, and new tools such as fora or blogs can be added incrementally as the need arises” (Levy, 2009, p. 777). In addition to blogs and online editing sites, synchronous computer-mediated communication (CMC) or chat tools are also invigorating the language classroom. Chat environments are productive because they engage “communicative behaviors;” therefore, “learners appropriated the chat room environment, transforming it into a learner-centered discourse community governed by communicative autonomy and the use of language and discourse functions that go beyond those encountered in the typical L2 classroom” (Darhower, 2002, p. 249). The Daedalus Interchange, a CMC designed by the University of Texas at Austin is a chat program that has been incorporated into both L1 and L2 classrooms with positive success. Citing a study from 1992, Darhower proceeded to duplicate the project in a similar, more modern setting (2002). The results revealed that online chatting provided

Chat environments are productive because they engage “communicative behaviors;” therefore, “learners appropriated the chat room environment, transforming it into a learner-centered discourse community governed by communicative autonomy and the use of language and discourse functions that go beyond those encountered in the typical L2 classroom”
(a) increased participation from all members of a work group, (b) allowed students to speak without interruption, (c) reduced anxiety which is frequently present in oral conversation, (d) rendered honest and candid expression of emotion, (e) provided personalized identification of target language errors, (f) created substantial communication among L2 learners, and (g) demonstrated a significant reduction of certain grammatical errors over time (2002, p. 250).

While Darhower does not make the connection clearly, it is obvious from his study that incorporating a chat forum in CALL programs can advantageously improve student interaction with their L2.

The historical record of CALL programs includes a number of diverse pathways leading to progressively superior venues, but also abandoned methods that should have remained incorporated. Researchers like Mike Levy opine that technology, although essential to reaching linguistic goals for CALL, tends to cast aside previous developments due to their outdated technology. Today, there is the expanding focus of CALL applications that require a more precise employment of linguistic research and an increasing demand for standards spanning distinct applications. Applying the linguistic research of CALL programs to the actual life experience of the users demands the incorporation of applicable feedback from users. The provision of more communication between the end-user and the front-end of the program development will ensure that designers meet the expectations of users and induct an interactive learning environment where language acquisition occurs in a stimulating manner.
References


**About the Author**

Wayne E. Arnold is a fifth year Ph.D. student at the Department of English of University of Louisiana at Lafayette, USA, with a focus of study in modern American literary studies as well as TESOL. He holds an MBA from Wright State University and a Masters Degree in English from Western Kentucky University. Scholarly interests revolve around culture and its relation to language development as well as literature as learning aid in the second language classroom.

---

i Kenner provides a valuable list of professionals associated with the beginning stages of qualified academics interested in further developing this field: “Our final list of participants reads like a ‘who’s who’ of the field of CALL: Ingrid Berljawsky, Joel Bloch, Chris Candlin, Carol Chapelle, Tom Dillon, Pat Dunkel, Irene Dutra, Gary Esarey, John Esling, Antoinette Gagne, Sandy Harris, Chris Harrison, Paul Hardin, John Higgins, Joan Jamieson, Ann Johns, Tim Johns, Chris Jones, Randall Jones, Stan Jones, Roberta Lavine, Don Loritz, Bill McMichael, Peter Mueller, Anne Merkel, Donna Mydlarski, Frank Otto, Michael Palmquist, Dana Paramskas, Nancy Post, John Roberts, Gail Robinson, Joan Rubin, Judy Scholefield, Frank Smith, Vance Stevens, Kim Merriam, Gerry Strei, Michael Sutton, Bill Vanderwerf, Judy Vernick, David Wyatt” (Kenner, 2009).

ii In a recent study (albeit limited in scope) of the use of CALL applications in the sign language classroom, Maria Mertzani noted that their research using the CALL program SignLab showed “evidence that CALL (SignLab) creates a positive environment for the teaching and learning of a sign language” (2011, p. 153); students became more autonomous through their use of SignLab and additional feedback was generated between teachers and students and peer-generated feedback.

iii Levy also notes that he has supported preserving past accomplishments in CALL for over a decade, citing his 1997 article in which he propounded, “the CALL community needs to build upon what has gone before rather than be led purely by the capabilities of the latest technological innovation” (as cited in Kennedy & Levy, 2009, p. 446).

iv Garrett’s call for templates has in fact been addressed by programmers and universities in the past, yet it seems to have failed to become a universal standard across CALL formats. He provides an example of an effort to create a universal structure by UK’s Open University, they have “for some time been investing considerable resources in developing a flexible set of reusable language learning activity-type shells which are content independent” (Bangs & Shield, 1999, p. 24).

v Particular early programs such as Grammar-Debugger, Syncheck, and LINGER, all designed during the early 90s, addressed parsing issues in language production (Cotos, 2009, p. 104).