

---

# *Teaching Arabic with Technology at BYU: Learning from the Past to Bridge to the Future<sup>1</sup>*

MICHAEL D. BUSH

JEREMY M. BROWNE

*Brigham Young University*

---

## **ABSTRACT**

Reporting in 1971 on research related to computer-based methods for teaching the Arabic writing system, Bunderson and Abboud cited the potential that computers have for language learning, a largely unfulfilled potential even in 2004. After a review of the relevant historical background for the justification of computer-aided language learning (CALL) and pedagogical considerations for instructional materials development, this article describes recent advances in online technologies, justifying the conclusion that the field is poised to make great strides. Given the high costs for materials development, it is essential (a) to not abandon existing materials, (b) to use the most effective techniques possible for new materials, and (c) to conform to existing standards to ensure the widest possible materials delivery. The authors of this article discuss efforts at Brigham Young University to work within these standards in the re-engineering of materials to make them more useful, maintainable, and accessible, describing at the same time important principles for creating materials that are interoperable with existing online delivery platforms. In this project, hundreds of Arabic activities from Apple's *HyperCard* environment were converted to Unicode-compliant, template-driven, XML-based, Web-deliverable activities. In addition to discussing Unicode, SCORM, and MPEG-7, the authors provide background and justification for important development decisions.

---

## **KEYWORDS**

language learning, technology, standards, Unicode, Arabic

Contrary to the hypercritical view of the archhumanist that anything associated with computers must by definition be sterile, rigid, and inhumane, it is seen that tastefully-designed CAI can be a most profitable and useful tool for the serious student. Indeed, bright and motivated students can "leapfrog" a great deal of tedious classroom work (Bunderson & Abboud, 1971, p. 68).

## INTRODUCTION

The quote above is from what is most certainly the first example of research conducted on the effectiveness of computers for learning any aspect of Arabic. The results observed are noteworthy, given the limited nature of the technology<sup>2</sup> available at the time of the study, especially when compared to systems of today.

This study has not been alone over the years in demonstrating the effectiveness of technology for learning. Indeed, ample evidence has accumulated during the past three decades, particularly in the form of meta-analyses and reports such as Kulik and Kulik (1987), Glennan and Melmed (1996), and McArthur and Lewis (1998) just to name a few. The following conclusions exemplify the reported results:

1. students generally learned more in classes when they received help from computers,
2. students also learned their lessons with less instructional time, and
3. students also liked their classes more when they received computer help.  
(Kulik & Kulik, 1987, p. 224).

There is also important evidence of the potential benefit of technology specifically in language learning. For example, one researcher conducted a review of the literature and an associated meta-analysis of the results of nine studies on the effectiveness of technology on language learning. He concluded that “the application of technologies can be effective in almost all areas of language education” (Zhao, 2003, p. 21). He went on to list several specific areas in which technology can be of benefit:

1. enhancing the quality of input,
2. enhancing the authenticity of communication,
3. providing relevant and useful feedback,
4. bringing authentic materials into the classroom,
5. enabling communication between students and distant speakers of the target language, and
6. providing opportunities for researchers to study language learning itself.

Unfortunately, despite the obvious benefits signaled now three decades past in the Bunderson and Abboud study (1971) and affirmed many times over (J. R. Allen, 1972; R. Allen, 1993; Chun & Plass, 1997; Blyth, 1999), up to and including the work reported by Zhao (2003), technology has failed to achieve the potential for learning recognized throughout these many years. One has only to walk into just about any language classroom in the world on any given day to see that the actual use of technology for language learning is slim to nonexistent. For example, Ehsani and Knodt maintain that the “practical impact” of technology on language learning “has been rather modest” (1998, p. 46). This is in some ways startling, especially considering the challenges and limitations of delivery systems of the past and the capabilities of the microcomputers to which students have access today.

For example, consider the increases in features that have happened since IBM released its first PC in 1981: incredibly fast processors (750 times faster than the original IBM PC), expansive random access memory (400 times more), absolutely huge hard drives (easily 24,000 times larger), startlingly high-quality audio in incredible quantities, life-like video, eye-popping (high-resolution) color displays, and mind-expanding Internet interconnectivity, significantly surpassing predictions made 25 years ago (Atkinson, Attala, Blitzer, Bunderson, Charp, & Hirschbuhli, 1978).

Given these advances, it should be clear that hardware capability is not the issue. To what then does one attribute the failure to exploit the capabilities of technology for educational purposes?

One author documents several barriers that slow the use of computer-aided language learning (CALL):

1. financial barriers,
2. availability of computer hardware and software,
3. technical and theoretical knowledge, and
4. acceptance of the technology (Lee, 2000).

With respect to financial barriers and hardware, while computers with the instructional capability of those used in the 1971 Bunderson and Abboud study were very rare within universities and almost nowhere to be found in schools at the K-12 level, powerful systems are now found almost everywhere. For example, the ratio of students to computers at the K-12 level in recent years evolved from 125 to 1 in 1984 to 5 to 1 in 2001, and fully 84% of classrooms today are connected to the Internet (Hayes, 2002). In August 2000, 51% of households in the US had one or more computers, up from 42% in 1998 (US Census Bureau, 2001). In addition, 42% of the households in the United States had at least one member who used the Internet at home. While Hayes (2002) also reports that 90.3% of K-12 teachers use the Internet in their teaching, one has only to visit the typical language classroom at any level to see that the number of students who regularly learn language with technology is almost as limited today as was the presence of computers in schools when the Abboud and Bunderson research was conducted.

Considering the lack of availability of hardware and software, it is clear from the discussion above that hardware is not a problem, given that hardware costs for increasing functionality continue to drop. Even the previously cited confusion with choices in hardware platforms (Bush, 1991) is disappearing with the advent of Web browsers and DVD for delivering video. With respect to software, scholars have long lamented the lack of good materials (Jorstad, 1980; Scebold, 1983). Despite the fact that most new textbooks include not only an audio program but also some form of computer software and most likely video of some sort, one can still ask why more interesting and useful software is not available. Garrett and Hart offer additional reasons, explaining that the fragmentation across incompatible platforms was a serious problem. They also explained that university faculty “understandably tend to focus on publishing efforts that will count towards tenure

and promotion, rather than on the development of teaching software” (1987, p. 441). This is indeed unfortunate since the greatest source of the content of the textbooks sold by publishers is found among professors in higher education.

While one can imagine that the natural source of software for language learning would be textbook publishers, the fact is that publishing houses must add the extra costs of video and software production to an already significant level of investment for textbook development. Furthermore, publishers make this additional investment often without any expectation of financial return for their effort, given that teachers in most cases expect these “extras” to come for free with the textbook (Dorwick, 2002).

With respect to technical and theoretical knowledge and acceptance of the technology, these factors remain a major part of the teacher training problem that is being addressed in conference presentations as well as by teacher education programs. Luckily, as technology becomes more available, people are also becoming more amenable to its use.

It is also important to observe that while hardware costs have dropped and system capabilities have increased, teacher expectations have also increased, at times even outpacing what has been practical or at least cost effective. It seems that there has been a consistent push to exploit the capabilities of each succeeding generation of hardware and software with developers never stopping to take full advantage of each preceding configuration. Noting this tendency, Dorwick (2002) pointed out in a recent banquet speech at the annual CALICO Symposium that it is important to remember that just because technology can do something, it does not necessarily have to follow that it has to be done.

Recognizing the potential for technology in language learning and its increasing power and availability, we arrive at the crucial question, “How much technology is enough to justify its use with language learners today?” We can only address the issues raised with such a question by asking two additional and related questions: (a) “*What* should we do with technology with respect to interesting and useful pedagogy?” and (b) “*How* do we do whatever we *should* do in languages such as Arabic and others that are represented with non-Roman orthographies?” In other words, in order to determine the demands to be made on technology, it is essential to address the challenge of deciding what should be done as well as how to do whatever makes good pedagogical sense. Indeed, rather than focus on whether or not technology works, Burston (2003) argues that it is more important to address the contribution that it makes to the pedagogical aims of the teacher and/or learner.

### **EXPECTATIONS AND INSTRUCTIONAL DESIGN STRATEGIES: THE WHAT**

Deciding what to do with technology in language learning environments is a process that draws on the essence of instructional design. Over the years, instructional design strategies have of course implemented the theories and principles of instruction of the day. Early efforts involved text-based exercises on the computer and dealt with dictation, translation, and vocabulary, supplemented at times with audio, and relied on simple string matching (Ruplin & Russell, 1968; Cur-

tin, Clayton, Finch, Moor, & Woodruff, 1972). The rationale of at least certain instances of this generation of materials was to relegate to the technology those elements of instruction for which there was no “time found in class under the audio-lingual method” (Ruplin & Russell, 1968, p. 85). The software of these researchers implemented “ear training” through “audio discrimination drills” so the student can learn “to associate sound and symbol” (p. 86). Other activities consisted of substitution-transformation drills, translation exercises, dictation and listening comprehension drills, and vocabulary drills. The mere use of the word drills illustrates quite well the period in which this work took place, a period in which audiolingual methodologies reigned supreme in the foreign language classroom.

Subsequent to this era, language learning theories of other sorts have provided guidance for selecting appropriate instructional design models. After stating that the computer “clearly has something to contribute to language teaching,” Krashen went on to explain that “Computers can be used to supply comprehensible input in the students’ area of interest via films, TV shows, and lectures” (1989, p. 393). He also stated that reading is an important “source of comprehensible input that has been nearly completely ignored in foreign language pedagogy” (p. 393). After his discussion of how reading can contribute to language learning, Krashen concluded, “Light reading requires no knowledge of DOS, no knowledge of Basic or C. And just think of how many comic books and paperback novels you can buy for the price of one 286 computer” (p. 403). There is a certain allure to an instructional approach that simply tells students, “Read!” Yet, one also has to wonder how the power of reading might be combined with the increasing availability of computers and the interesting capabilities they now provide.

In the years following Krashen’s pronouncements, prices have continued to drop even as capabilities have increased. Taking advantage of these trends, various researchers have demonstrated the value of new features for language learning. For example, Borrás and Lafayette (1994), investigating the effects of multimedia on listening and speaking, demonstrated that subtitles are useful not only for helping students to better comprehend authentic input delivered via video but also for producing more comprehensible output in communication. Chun and Plass (1996) found that visual, multimedia-based annotations combined with verbal annotations helped learners retain word meaning better than verbal annotations alone, illustrating how reading can be enhanced with new technologies.

Other researchers have concluded that “drills don’t lead to skills” (Blyth, 1999, p. 40) and have directed their efforts at using technology that requires that students deal with meaning rather than just form. This conclusion was not supported at least by the subjects in Blyth’s study. He noted that “students reserved their highest praise for the ‘drill-’n-kill’ rote grammar exercises and their greatest disdain for the Internet activities that emphasized critical thinking skills and synthesis of information” (p. 52).

Yet, the idea that form-only activities (“drills”) are not beneficial for language learning is a notion supported by recent research. For example, Wong and VanPat-ten wrote in their detailed review of the research in this area, “As far as acquisi-

tion is concerned, drills are simply unnecessary and at best a waste of time for the development of communicative language ability” (2003, p. 418). This is not to say that practice does not make perfect, but it does say that the sort of practice in which students engage is important. Indeed, these researchers assert that a focus on form that “is informed by what we know about processes involved in acquisition is highly desirable” (p. 418).

Researchers have been seeking appropriate ways to use technology that are compatible with current theories of the nature of language and how it is learned. For example, Chappelle (1998) proposed a model for the use of technology and put forth her “Suggested Criteria for Development of Multimedia CALL” (p. 27). The list of principles establishes potential connections between the multimedia characteristics provided by the technology and beneficial psycholinguistic responses that the software can evoke from learners

1. making key linguistic characteristics salient,
2. offering modifications of linguistic input,
3. providing opportunities for “comprehensible output,”
4. providing opportunities for learners to notice their errors,
5. providing opportunities for learners to correct their linguistic output,
6. supporting modified interaction between the learner and the computer,  
and
7. acting as a participant in L2 tasks (Chappelle, 1998, pp. 27-28)

Other recent work has been laying the groundwork for the development of “courses designed systematically in response to learners’ precisely specified communicative needs, for developing functional foreign language proficiency without sacrificing grammatical accuracy, and for harmonizing the way languages are taught with what SLA research has revealed about how they are learned” (Doughty & Long, 2003, p. 50).

The guidance that Doughty and Long provide for the creation of “an optimal psycholinguistic environment for language learning” entails development that is based on “Pedagogic Procedures” derived from “Methodological Principles” that can be classified into four categories: Activities, Input, Learning Processes, and Learners (Doughty & Long, 2003, p. 52). Their list of principles reflects a great deal of what has been learned from research with respect to language teaching and learning. For example, they conclude from Long’s writing regarding conclusions drawn from empirical studies that rich input is essential and that some attention to form is necessary as learners learn by doing. After pointing out that input that is based on elaborated texts is superior to either genuine texts (“authentic” texts as some call them) or simplified texts, they also discuss the importance of negative feedback that is part of instruction that is individualized to meet the needs of each learner. While the thrust of their proposals is based on Long’s work on classroom language teaching, this particular article connects this work to less commonly taught languages in the distance learning setting, establishing an important foundation for increasing the impact that distance learning technology can have on language acquisition.

It is clear, based on this short review of several principles that can guide technology development, that the language teaching profession is making progress in finding the best ways to use online technology for learning languages. Given this observation, it is also important to recognize and underscore a self-evident truth of language learning: students will learn what they are taught and what they practice. If they are assigned lists of vocabulary words to memorize, then they will be able to demonstrate mastery-of-meaning recall on tests of vocabulary. While technology can deliver incredibly sophisticated, flashcard-based interactions that can facilitate the memorization process, unfortunately, the connection between memorization of word lists and using language for communication is tenuous at best. Indeed, the language profession has a lot of experience with the notion that such memorization does not lead to the ability to use the targeted vocabulary in real communication. For example, Beheydt asserts that “The simplistic view that vocabulary learning is nothing more than the memorization of a series of word-forms with fixed meanings ought finally to be discarded” (1987, p. 55). In other words, if we want our students to be able to communicate in a language, then the learning activities in which we require that they be engaged must be significantly infused with activities that require them to use language for communication, activities that target receptive as well as productive skills.

Thus, the ideal software for language learning has to be something that (a) involves either the understanding or creation of meaningful messages or content to accomplish some purpose relevant to the student, (b) is filled with activities that are guided by current thought from second language acquisition theory, (c) is perceived by students to be useful, (d) is designed to be as motivating and engaging as students of the video generation expect, and, yet, (e) is cost effective to produce, making it as readily available and accessible as comic books. This is nothing if not a very tall order. Indeed, the problem to date has been that meeting expectations a-d has been diametrically opposed to meeting expectation e!

## **TECHNOLOGY IN ARABIC LEARNING: THE HOW**

### ***Background***

As challenging as such a list might be in any language, creating something interesting and useful in languages such as Arabic is even more difficult. The first author of this article remembers very well from the first CALICO Symposium in 1983 an evening meeting that addressed a similar set of issues. A small group of professors from various universities and representatives from several federal agencies involved with language training met to discuss, among other things, problems related to “non-Roman orthography languages” and how they relate to computer-aided language learning. A portion of the discussion centered on the capabilities of the Xerox Star system, pretty much affordable only by organizations within the federal government, and on what could be learned from the system’s use, as well as other technical challenges that many in the group felt needed to be addressed.

Fortunately, in the intervening 20 years, researchers have begun to write about



software in Arabic in particular, illustrating the progress that has been made, at least with respect to certain aspects of the problem. Ten years ago for example, one writer addressed the use of the computer for learning Arabic as being ideal for removing “grammar instruction from the classroom itself” (Allen, R., 1993, p. 8). More recently and in a similar vein, Nielsen and Carlsen have stated that “structured grammar drills and exercises which aim at making the learner more confident in the use of specific rules should preferably take place outside the classroom” (2003, pp. 96-97). Along this same line, other researchers have developed various types of activities for Arabic including “listen and fill the gaps,” “listen and sort the jumbled text,” “listen and repeat,” as well as “nine text-based exercises” (Cushion & Hémar, 2003, p. 260), all created with an Arabic authoring tool developed at London Guildhall University (see <http://www.lgu.ac.uk/langstud/call/details>).

### ***Efforts at BYU: Learning Some Important Lessons***

Early work in Arabic at Brigham Young University (BYU), pretty much followed these same philosophies with early software entitled *EMSA Companion*.<sup>3</sup> Designed around *Elementary Modern Standard Arabic* (Abboud & McCarus, 1983) and *Let's Learn Arabic* (Allen & Allouche, 1986), they were created using a HyperCard-based template approach. The exercises were used “for drilling vocabulary and grammar, and for reading and working with texts” (Belnap, 1993, p. 15). As Belnap wrote, however, “developing and maintaining BYU’s CALL materials has been a time-consuming and sometimes frustrating experience. It has come with a price” (2001, p. 370).

Also associated peripherally with BYU was a project to create software for using commercial movies for language learning. Developed by the first author of this article, under contract with HRB Systems, this software was used by teams at Alpine Media, BYU, as well as at universities around the United States to develop a series of 16 feature films in nine languages, one of which was *al-Aragouz (The Puppeteer)*, a 1989 Egyptian movie starring Omar Sharif and directed by Hani Lachine.<sup>4</sup> This movie in Egyptian dialect has been used over the years in various courses at BYU but, unfortunately, has for the past few years been available on only one legacy, interactive-videodisc-based system (Belnap, 2001).

As part of a Federal grant we received at BYU in January 2001 from the National Security Education Program, we conducted a review of efforts such as these over the past years to establish a foundation from which to launch a new generation of development. Our purpose has been to implement with today’s technology what was possible with past generations of software and hardware, all the while expanding into new and interesting areas. In particular, we derived some fairly simple but important considerations for the development of online learning materials for Arabic, summarized in the following topics that are intimately connected to our recent research and development goals:

1. high quality materials that are also easily distributable,
2. universal standards for the representation of Arabic script, and
3. implementation of cost-effective development strategies.



While these topics are related to Chapelle's list of characteristics for ideal materials provided above, it is nonetheless different. Where the earlier list dealt mainly with the nature of the materials and how they would be perceived by learners, this particular list deals primarily with how the materials are to be created. Underlying our work are some basic assumptions that have guided our efforts and that have come from several years of experience working in the development of online, multimedia-based language-learning materials. On the one hand, one assumption is that static, predictable exercises that converge on a restricted number of correct answers should be done online. On the other hand, exercises that are open and dynamic, thus requiring human intervention for assessing their correctness, should be reserved for the classroom where teacher-student or even perhaps student-student interactions can take place. Stated another way, we believe that if something can be done by students with the technology, then this is what should be done. This aspect of learning stands in contrast to activities that, at least for the foreseeable future, can only be accomplished by instructors with their students. We recognize of course that the technology-teacher distinction is an area that is ripe for investigation. Indeed, a complete and in-depth assessment of this subdivision of instructional tasks between the technology and the teacher should be one of the targets for research and development in the field of foreign language education.

### *Developing a Solid Technical Approach*

There have been in the past perhaps three primary technical approaches for creating interactive, multimedia instruction. One has been the use of classical software development strategies implying the creation of software programs that enable the interactions between the system and the learner. A second is the use of general-purpose authoring systems for the creation of instructional materials. A third is the use of special-purpose authoring systems that were created with a particular instructional strategy in mind, often for particular subject areas such as language learning. Each approach has advantages and disadvantages with respect primarily to cost and flexibility in design.

The first approach uses conventional programming languages such as JavaScript (for cross-platform flexibility) or Visual Basic (for PC-only delivery) and has the advantage of providing a great deal of flexibility to developers for either standalone delivery or delivery based on Web technologies. Unfortunately, this flexibility comes at a cost. Indeed, the programming approach has the significant disadvantage of being quite labor intensive, an effect that translates into higher development costs.

The second approach employs software known as "authoring systems" that are by nature general-purpose tools for creating instructional content. Example systems are *Authorware* from Macromedia and *ToolBook* from Click2Learn. While it is not the purpose here to provide an exhaustive overview of authoring systems, one other system warrants mentioning: the *Multimedia Instructional Tutoring and Authoring System (MITAS)* (see <http://www/maad.com/index.pl/mitas>) given that its use has been reported in the development of materials in Arabic (Alhawary,

2001). In addition, *MITAS* has capabilities for using speech recognition as stated on the Web site for the software—a “corpus-based, continuous, speaker-independent speech recognition system” in Arabic—a feature that interests quite a few practitioners in the area of technology-enhanced language learning. Unfortunately, the software does not appear to provide for Web-based delivery.

While solutions such as these have the advantage of being more cost effective than the programming approach, their use can still require a significant investment in resources for materials development. One approach to increasing cost effectiveness has been the creation of specific templates for use within systems such as *ToolBook* and *Authorware*. Unfortunately, however, the lack of multiplatform capability for authoring has perhaps limited their acceptance, especially in education. While both companies have to some extent addressed this issue for delivery, each of these two systems runs only under Windows for authoring and development.<sup>5</sup> Nevertheless, for content delivery both companies are wholehearted supporters of the Shareable Content Object Reference Model (SCORM) specification that will be discussed below, a definite plus for each system, but especially for *ToolBook* since SCORM in theory provides cross-platform delivery via technologies that are Web compatible.

Finally, over the years numerous specialized authoring tools dedicated to the development of language learning materials have appeared on the scene. From *Dasher* and *WinCALIS*<sup>6</sup> to *SuperMacLang*<sup>7</sup> and *Libra*<sup>8</sup> (now *Gemini*) among others, these systems have provided interesting capabilities for language materials developers. However, such systems come with the disadvantage of reducing the flexibility that many developers would like to see in the materials development process, due chiefly to the fact that authoring systems often implement a very limited number of instructional strategies and techniques. This restriction often causes subject matter experts and instructional designers to feel constrained, serving to limit the appeal and widespread success of such systems. Furthermore, the lack of cross-platform availability, combined with the overwhelming development and ubiquitous nature of the Web, have kept these systems from widespread adoption.

Two systems for developing foreign language learning materials that in fact do address Web delivery are *WebCALIS*<sup>9</sup> and *MaxAuthor*.<sup>10</sup> The *WebCALIS* system is an updated version of *WinCALIS* that, according to the software’s Web site, “supports about 60% of the features of *WinCALIS*, but also includes some features which *WinCALIS* does not” (Humanities Computing Laboratory, 2003). The *MaxAuthor* system provides capabilities not unlike the various other specialized authoring tools listed here, but Arabic is not listed as one of the written languages supported. The *WinCALIS* system does list Arabic as a supported language.

Unfortunately, neither *WebCALIS* nor *MaxAuthor* meets the test of yielding materials that can be easily integrated into whatever course management system (CMS) an institution might use. For example, a recent report of CMS usage on college campuses stated 94% of the colleges surveyed reported the use of a course management system. Of those using a single CMS, *Blackboard* led all others with 46% of the campuses using their software followed by *WebCT* with 35% (Market

Data Retrieval, 2003, p. 2). While it is not out of the question that software created using several different authoring systems could technically be implemented by any given school or college, issues of system maintenance and support argue against such an approach. Even more problematic is the implementation of learner-performance tracking, something that is not possible with the delivery systems of the past but is something that is imminently possible on the systems of the future.

With respect to technical approach, there are several lessons to be learned from consideration of the various approaches to date for developing interactive materials for learning language. First, the use of software that is locked to a particular technology is doomed for the dust bin. If there is anything we know today about technology is that it changes and that that change is increasing at an exponential rate. It is therefore pure folly to commit the significant resources necessary for creating interactive materials without insuring a migration path for the materials to follow that will overcome the inexorable obsolescence built into any technical approach. Thus by definition, proprietary solutions cannot over time meet the requirements for the creation of high-quality, easily distributable materials in Arabic.

It is also important to use the best tool for each task to be accomplished. As an example, early authoring systems had proprietary software for editing graphics and audio files, but such tools have been replaced by widely available, general-purpose tools for multimedia development. It is also important to recognize that the capabilities of specialized authoring tools are quickly being matched by other general-purpose tools that can be easily adapted for use in areas such as language materials development and that provide instructional designers with the creative flexibility they need. Essential, therefore, is a software engineering approach that (a) takes advantage of general-purpose tools of all sorts and within which tools are tailored to create authoring solutions that are not only powerful and flexible but also cost effective and that (b) supports the creation of materials that are deliverable on all new machines currently being purchased by students and installed in schools.

### ***Implementing Unicode and XML for Exercises in Arabic: A Saga***

In seeking to develop such an approach, it became clear to us very early that our software needed to build on capabilities that were an integral part of the Web. This meant that the software needed to be Web-browser-based and platform independent, a deceptively simple statement that implies the creation of something that will run on a significant majority of all personal computers. This would have been an incredible challenge over the past 30 years, but a terrific outcome of recent developments associated with the Web has been the creation of an environment that enables just such an accomplishment: the ability to develop software that runs on almost all personal computers (PCs as well as Macs), using any browser (Internet Explorer, Netscape, Safari, or Mozilla), and ensuring distribution that could not even have been imagined in the past. In order to achieve the production of the necessary high-quality software, we needed the representations of the

language used in the materials to be of the highest possible fidelity with respect to the real world in which the language is spoken. This means using audio and video of real people speaking real language at the highest affordable quality of reproduction, accompanied by textual representations of what is said in the audio and video texts, all factors alluded to by Chappelle (1997) in her overview of ways that CALL can foster or mirror real communication. Finally, we recognized the need for working in conjunction with established standards at every phase of the development process, a lesson learned in previous settings (Bush, 1989a). With respect to the writing system, this means using Unicode (Cushion & Hémond, 2002), and for software this means SCORM (Bush, 2002). The situation for video gets a bit more complicated with the proliferation of video formats on the Internet. Although there has been some consolidation in recent years, we are still faced with a wide variety of video formats and playback mechanisms: MPEG-1, MPEG-2, MPEG-4, QuickTime, Windows Media, Real Video, and now video for Macromedia's *Flash* using the Sorensen codec. To address this problem, we have been participating in the development of an approach to implement MPEG-7 for instructional applications that use video, a development that will be described in the section below on implementing standards and specifications.

Unicode is a system that "provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language" (Unicode Consortium, 2003a). The Unicode Consortium works with member companies (most of the major software and hardware manufacturers), standards organizations such as the International Standards Organization (ISO), as well as the Worldwide Web Consortium (W3C) to ensure that the stated goal of the system becomes a reality. It began in 1986 as a group at Xerox undertook to "map the relationships between identical Japanese (JIS) and Chinese (simplified and traditional) characters for quickly building a font for extended Chinese characters" (Unicode Consortium, 2003b). This early work expanded rather quickly to Apple and, by 1989, included Microsoft and IBM. The consortium was incorporated as Unicode, Inc. in January 1991, and the Unicode Standard Version 1.0, Volume 2 was printed in June of 1992. Support for Unicode has been emerging during the intervening years, bringing the standard now to Unicode 4.0.0. With Microsoft Windows XP and Internet Explorer 6.0, Apple's OS X 10.3 and Safari, as well as Linux and Mozilla, it is now possible to be reasonably confident that Unicode-based Web documents created on one system will display on the others.<sup>11</sup>

By far one of the greatest challenges we have had in working to create the next generation of Arabic materials has been to bridge the gap between the past and the future. As part of various projects to produce online materials, we began the process 2 years ago of modernizing our institution's computer-assisted Arabic language instruction. To jump start the process and avoid reinventing any unnecessary wheels, we decided to address the computer-based interactions that had been created for Apple's HyperCard environment in the early 1990s (Belnap, 1993). Although these materials had been used successfully and extensively for almost 10 years by hundreds of students (and continue to be used), the age of the software was beginning to show and its demise appears to be certain, given that

*HyperCard* is no longer supported under Mac OS X.

Reflecting on how to bring these interactions into the 21<sup>st</sup> century, we decided to make templates of each question type in such a way that the templates could be assembled in any order to form a complete activity spanning perhaps several HTML “pages.” The rationale for this approach was to create a new system that could be easily corrected, updated, and supplemented. Where revisions had been quite difficult during the intervening 10 years since their creation, this new implementation strategy would significantly enhance our ability to keep moving the activities into the future, despite inevitable changes in technology and delivery systems.

When the word “template” is used in discussions of instructional technology, it brings to mind an activity with a set structure and that only requires content. We took the concept a step further by making templates of each individual type of question or content. For example, there are text-, image-, and video-content templates, stem-with-options (multiple choice), classification (matching), and fill-in-the-blank question templates. These micro-templates can then be assembled into a fully interactive activity or even nested within each other. In fact, nesting is sometimes required. (A multiple choice question template would be nothing without the content that populates its stem and options.)

Also very early in the planning for the conversion of these interactions, we decided that our development and delivery goals would be best met if we were to strictly adhere to standardized technologies. HTML 4.0 and ECMAScript (JavaScript) were the major components for software development in our lab at BYU, but, more importantly, we understood that these standards were supported and implemented by several vendors on every consumer platform. It was natural for us to also use XML, also standardized and widely accepted, to store the content of our next generation activities. Based on previous development efforts, we determined that we would have two basic types of XML elements: expository content elements and inquisitory question or assessment elements (Merrill, Li, & Jones, 1992). The attributes of the elements would further define their purpose and store other descriptive information to be used by the rendering mechanism for the various activities we were creating.<sup>12</sup>

To render the files that contain the content marked up by using XML, we created a few dozen server-side scripts. Although we chose PHP for these scripts, the architecture could be applied using almost any language or server-based system such as ASP or JSP. The structure of the XML (the elements it contains as well as their order) determines which scripts get executed and in which order. The scripts, in turn, render the content of the XML files by writing the necessary HTML code to a string that is saved for later transmission to the client using standard HTTP protocols.

Much of the HTML written by these rendering scripts depends upon the descriptive attributes of the XML elements. When the last element of the XML has been rendered, the string of HTML and JavaScript code can either be sent to the client computer as a completed HTML/JavaScript-based activity, or written to a static HTML file for export as learning activities that are appropriately formatted for later delivery.

An important characteristic of this entire system, however, is that it is completely dynamic. Each time a user requests an activity, it is rendered anew. For example, given that each content element of type text also contains an ISO language declaration, the rendering scripts use that declaration to determine the size and justification of the text in the HTML. When the first Arabic activities were being tested, it became apparent that a text size of 18-points was too small for the Arabic to be readable by novice learners of the language. Because the size of the rendered text was not stored in the XML source files, a change to a single integer in a single rendering script was sufficient to make the change to every instance of Arabic text in every activity created before or after the change.

Choosing an encoding scheme for our text-based content was the easiest decision of the entire development process, given that Unicode fits very well with the other standards in use in our lab. The specification is available electronically for free and, just like HTML and XML, it is implemented by multiple vendors. It also solves most of the problems we would have faced if we had tried to shoehorn Arabic text into some proprietary approach to encoding using ASCII.

Two issues regarding Unicode, however, came to the forefront very early in our development efforts: (a) not all Web browsers that could be used for viewing our standard HTML were capable of rendering Unicode-encoded Arabic text, and (b) the text that had already been entered into the *HyperCard*-based activities would have to be either re-typed into a Unicode-enabled editor or converted programmatically from the previous Arabic encoding scheme into Unicode.

The first issue was the practical reality of installed systems. We had decided on standard HTML for the presentation of our content so that anyone with a browser capable of rendering HTML 4.0 would be able to view our activities. Wanting to adhere to standards, we had also decided upon Unicode for encoding and storing our content, significantly reducing (at that time at least) the number of browsers that could properly display our activities, given that Apple's OS 9 operating system was in use in BYU's Arabic department. At the time, only the open source Mozilla browser rendered most Unicode text correctly on OS 9. On the Microsoft Windows platform, however, the built-in Internet Explorer Web browser used the operating system's text rendering API, enabling the viewing of almost all Unicode alphabets correctly.

After some debate, we concluded that our development towards standards was more important than the number of browsers which would successfully render the content correctly. Stated simply, we predicted that if we built to standard technology, such as Unicode, browsers would eventually support the technology.

This prediction came true with the release of Apple's OS X, putting the Macintosh on the path to full Unicode compliance. Given that Mozilla was the only browser to render Arabic, we continued to have difficulties until Apple announced their Safari browser project. Of significant interest to us concerning this browser, even before it was available for download, was the fact that it was being built upon the open source HTML rendering program KHTML, the code base of Konqueror, a Unix-based browser that we knew could support Unicode-encoded Arabic text.

Furthermore, there remained an annoying bug in the Macintosh rendering



which caused separate forms of Arabic letters to appear several sizes smaller than the rest of the text, a problem addressed in the release of OS X 10.3 (“Panther”). Although there remain some small, diacritic rendering issues on this platform, our expectation that browsers would evolve towards the Unicode standard has been fulfilled.

Our second issue, the fact that we had possibly hundreds of staff hours invested in typing the non-Unicode encoded Arabic text used in the *HyperCard*-based version of our exercises, addressed technical challenges more than the practical reality of installed systems. Given that *HyperCard* used a proprietary format for storing its programming information, we first feared that in the worst case we were going to have to view the exercise information on the screen of one computer and then retype it all on another system. Because the creator of the *HyperCard* stacks had left a backdoor in the software to view the content of each activity as plain text, we were able to easily copy the text into a text document, the first step in the conversion process.

The technique for rendering the Arabic text in the *HyperCard* exercises was very common in the pre-Unicode world. First, the text was input using a transliteration format. For example, “A” had been used when the Arabic alif “ا” would appear. Of course, not every Arabic letter has a Roman equivalent, but since the opposite is also true (especially Roman capitals), complete transliteration was accomplished. Although it was difficult to read by someone not familiar with that particular transliteration system, the predictability of the system made it possible to convert, with some level of difficulty of course.

The fact that not all Arabic letters have Roman equivalents is just the beginning of the challenges the developers of the *HyperCard* activities had to overcome. Arabic script must also be displayed right to left, so an algorithm was written to reverse the text of each line. Arabic, as a script, also has different forms for each letter. A haa may appear as an initial (ح), medial (ح), final (ح) or separate (ح) form. Again, our team developed an algorithm to shift each letter to its appropriate form, resulting in illegible text in the intermediate state of the phrase.

The need for a character for each form of each letter quickly exhausted the normal Latin character set and necessitated the use of “upper-ASCII” characters, which some programs actually used to elicit certain commands. This meant that even opening the converted text in some programs, or trying to do any sort of processing on it, would lead to problems. Such transliterated text was therefore only usable by programs which understood the specific transliteration in use for a particular project. Since most solutions such as this were created on an as-needed basis by individuals, there was little hope of using new programs to sort, search, edit, or otherwise manipulate the text.

The text was then displayed within the *HyperCard* activities using fonts from the Al-Kaatib Arabic word processor. Figure 1 provides an overview of the process of displaying a simple Arabic message using this technique.



Figure 1

Process Step	Text Appearance
Original Arabic message	الشرق الأوسط
Transliterated Arabic	alprq al;ws,
Text reversed to read right to left	,sw;la qrlpla
Converted for presentation	u\$zIzrÆ?öZ
Masked with special font	الشرق الأوسط

Our colleague Dilworth Parkinson, the professor who supervised the *HyperCard* project and who was responsible for a major portion of the technical development, was available to facilitate the conversion process. With his help and with only minor tweaking of the old text files, we were able to use a pair of his Perl scripts to convert files from the old encoding approach to Unicode-compliant files.

Many of the problems for which the *HyperCard* developers had created Mac-specific workarounds that worked fine for that platform have since been pretty much solved in Unicode by leveraging the power of today's computers. (Figure 2 shows the data representation for each step of the process of creating and viewing Arabic text encoded as Unicode.) While problems remain, in particular with how word processors mix right to left and left to right characters in the same paragraph, we are much closer to "create once, run anywhere" than ever before.

Figure 2

Process Step	Text Appearance
Original Arabic message	الشرق الأوسط
Text in Unicode word processor	الشرق الأوسط
Text stored in Unicode file	الشرق الأوسط
Text rendered by Unicode browser	الشرق الأوسط

First and foremost, Unicode provides for more individual character encodings. ASCII and extended ASCII only provide for 8-bit (single byte) encoding, which means a total of 256 "code points." Unicode's basic "code space" is double byte, or 65,536 "code points." While even this large number had to be extended to handle nonalphabetic languages such as Chinese, the problem is much simpler for Arabic, given that this code space is enough to represent each form of each letter, thus eliminating the need for transliteration. Although this would have caused worries a few years ago when our hard drives could only store a "few million" bytes, now, with hard drives commonly available with 160 gigabytes, using extra bytes to store texts created with non-Roman orthographies is very plausible. Secondly, given the processor power available in the current generation of personal computers, Unicode can delegate the decision as to which direction a text must

be rendered to the rendering program, be it a Web browser or a word processor, making it possible for all text to be stored left to right. Finally, again thanks to an increase in processing power and an increase in memory, Unicode can also rely on the rendering program to change each letter into its appropriate presentation form, the actual data only stores the letters in their separate forms.

The ability of Unicode-based systems to change or morph text is one of the most difficult features for Unicode neophytes to comprehend. Unicode editors and renderers are not dumb digital typewriters in the tradition of the text pad applications to which we are accustomed. As one types the text, the letters change depending on where they appear in a word. If one were to then type a series of numbers, the cursor would automatically begin typing from left to right because, as all good Unicode editors know, although letters in Arabic are written right to left, numbers go left to right. A wonderful demonstration of this in action is to type some letters, some numbers, and then some more letters. Backspacing through the string will reveal just how aware a particular editor is of the Arabic alphabet.

### *Implementing Standards and Specifications for Interoperable Content*

SCORM is a specification that has been designed by groups from industry and government to facilitate the development of components of online learning materials that are accessible, interoperable, reusable, and durable (Bush, 2002). SCORM builds on the IEEE Learning Object Metadata (IEEE LOM<sup>13</sup>) model and the IMS Content Packaging Specification. SCORM provides (a) a standard approach for creating metadata that makes content searchable, thus accessible as well as (b) an approach for packaging online instructional materials for moving content across platforms. By implementing materials in accordance with SCORM, developers can be assured that their materials will run under any Learning Management System (LMS) or Course Management System (CMS) that has implemented the SCORM specification<sup>14</sup> (Tansey, 2004). The long term implication of such a system is the fostering of the development of an “instructional object economy that rewards content creators for developing high quality learning objects and encourages the development of whole new classes of products and services that provide accessible, sharable and adaptive learning experiences to learners” (Dodds, 2001, p. 1-12).

As is obvious to anyone working in this area, the ability to represent the script is an essential element for creating online materials for learning Arabic. The various software approaches mentioned earlier all used nonstandard or even locally developed approaches for representing Arabic characters.

While the XML-based approach for representing content described above has been extraordinarily successful in enabling the efficient creation of new exercises in Arabic, a course that we developed in Swahili in the ARCLITE Lab of the Center for Language Studies at Brigham Young University was the first project that benefited from the implementation of SCORM. While most people who are aware of SCORM think first of metadata, the descriptive information that will allow instructional objects to be easily located, we did not initially address the issue

of metadata with the Swahili materials. It is our contention that there is a great deal of work that needs to be done in all languages with respect to how exercises should be described with metadata, a problem that is being addressed in work at the Defense Language Institute and on LangNet at the National Foreign Language Center. Both of these organizations are overseeing a great deal of development in SCORM-conformant language-learning content, work that will provide some much needed direction in the area of metadata for language learning materials.

The positive impact that we have gained from implementing SCORM has come in the conversion of the standard Web-based materials (HTML and JavaScript) to modules that run under a SCORM-conformant learning management system. Working under a research and development license from Click2Learn, we made slight modifications to recently completed online Swahili materials that had taken our team about two years to develop. In a matter of days we were able to get the course running under the *Aspen Enterprise Suite*. BYU currently uses *Blackboard* for course management support, but, although *Blackboard* has been participating in the development of SCORM, their software was not yet completely SCORM-conformant at the time we did this work. The benefit that we will get from using SCORM is that once *Blackboard* has completely implemented SCORM, it will be an easy matter to port these materials from *Aspen* to *Blackboard's* system or from *Blackboard* to *Aspen* or any other SCORM-conformant LMS for that matter.

We are presently completing the conversion of our delivery system for the new XML exercises for Arabic from the server side rendering described in the previous section to a SCORM-conformant approach that runs similarly to the one we are currently using for Swahili. For Windows machine delivery we also will be able to use XSLT on the client machine in conjunction with the Web browser, removing the need for server-side rendering and creating another level of flexibility in delivery. On the Macintosh side, for the moment, we will continue to do the rendering on the server. As part of this overall development process we are converting our XML schemas to adaptations of those specified in the IMS Question and Test Interoperability Specification (QTI).<sup>15</sup> While QTI does not perfectly suit our purposes, given that it was designed for assessment items, we feel very strongly that it is a significant and important first step for representing content in such a way as to enable the separation of content and presentation, a proven development paradigm for efficient content development (Bush, 1989b).

MPEG-7 is an international standard for describing multimedia content. Where MPEG-1, MPEG-2, and MPEG-4 specify the compression algorithms and file structures for *delivering* digital content, MPEG-7 specifies metadata structures for *describing* the content delivered via MPEG-1/2/4 (Day, 2001). MPEG-7 descriptions, which are XML document instances, can be used to retrieve digital content both at the macrolevel of an entire multimedia asset and at the microlevel of a small subset of an asset, such as a subscene of a film.

Our interest in this type of standard goes back to videodisc. Indeed, almost 15 years ago, the first author of this article circulated among members of the CALICO Videodisc SIG a recommendation for standardizing how members could represent video content on videodiscs for language instruction. Our dream then was

that such a development would open up numerous opportunities for sharing interactive content and associated language learning materials across institutions.

It would appear that this dream is now much closer to reality, given the existence of (a) widely available mechanisms for distributing video content in digital form (e.g., MPEG-1, MPEG-2, MPEG-4, and DVD) as well as (b) MPEG-7 standards and the associated MPEG-7 profiles that meet the goals of the disparate groups represented by the people with whom we have been working.

Making such a standard a reality is an excruciating process involving many people from around the world. An industry group came together in 1996 as the ISO/MPEG working group and began by assembling requirements into what is now called the "Multimedia Content Description Interface" or MPEG-7. After a great deal of initial work, MPEG-7 entered the final editing phase in 2001 and has just recently been published as an international standard ISO 15938<sup>16</sup> (Manjunath, Salembier, & Sikora, 2002).

Our group at BYU is part of the current stage of the development process and is a key player in the creation of what is called a Core Description Profile (CDP), essentially a subset of the MPEG-7 specification. CDPs use a portion of the descriptive capabilities of MPEG-7 to serve a special purpose; in our case, education. Building on common interests, we are working on the development of this CDP with three other groups: NHK (the Japan Broadcasting Company), the National Institute of Standards and Technology (NIST, the standards agency of the US Federal Government), Motorola, and TV GLOBO Internacional (a 24-hour Portuguese language channel from Brazil that broadcasts worldwide via satellite the programs from the top Brazilian television network).

We anticipate the end result of this development to be enormous amounts of materials that can be easily adapted to meet the needs of language learning. Preceding the availability of materials will be the creation of tools that are interoperable and allow for the cost-effective development of materials, some examples of which are annotated movies, news programs, and so forth.

### ***Putting It All Together***

The approach we have been developing at BYU focuses on the use of software engineering principles and the application of standards that together will provide the benefits of programming combined with the cost effectiveness afforded by authoring systems. While the first implementations of such a system requires an important investment in programming for the creation of tools, when it is done correctly, economies of scale quickly provide payback for this initial outlay. To accomplish this, solid software engineering principles must be applied, not only in the development of programs for materials delivery, but also in the development of content management tools as well as organizational approaches for content development, all of which are built to conform to the various standards that are emerging.

The first example with which we are addressing the full implementation of these various standards and software approaches is the conversion of the movie,

*al-Aragouz*. As discussed above, while we have implemented some of the principles discussed in the conversion of existing exercises and the creation of new activities to support the *al-Kitaab* textbook (Brustad, Al-Batal, & Al-Tonsi, 1997), it is with this movie that we are exercising all of the lessons we have learned over the past 15 years. In doing this, we are taking the *al-Aragouz* materials that were created using an authoring system that was available in 1993 and converting the content using XML, Unicode, and MPEG-7 for delivery using all Web browsers on all computer platforms. Given the amount of video material that makes up the movie, the video will be available on DVD, implementing another principle of our work: “delivering the control bits over the Web and the video content bits on plastic.” This particular approach will have application until the Web is more reliable for the delivery of bandwidth-intensive video than it is today. Indeed, this coming transformation will pose no implementation problems, given the development strategies that we are employing. In essence, this means the separation of content from presentation using standard schemes for creating and storing content, such that it becomes relatively easy to convert the materials from one technology to the other.

The conversion of *al-Aragouz* is a perfect example of the challenges of moving from a technology generation of the past to one of the future. Belnap lamented that this particular application was “not a priority for the distributor,” stating that students “continue to use this valuable program, around which we have built an entire course” on one dedicated 486 machine (2001, p. 368). Even if that one machine would have been enough to meet the needs of the 30 or more students in the course to be taught during Winter Semester 2004, the hard disk drive on that machine was no longer operational, increasing the urgency for finding a longer term solution. Finding a compatible disk drive and installing Windows 3.1 and the necessary delivery software were almost insurmountable tasks, illustrating the previously described problems that arise when materials are dependent upon one particular technology.

To migrate this application to current technology requires putting the video into digital form, authoring a DVD, reformatting the text files containing (a) the transcript of the movie, (b) the language and culture notes, and (c) the dictionary entries. In addition, it is necessary to convert videodisc frame numbers to time codes that will be compatible with the DVD being produced. Finally, it is necessary to create new software to tie all these elements together for use by students. Unfortunately, in the grand scheme of things, the technical challenges that arise with all these steps might not be the greatest difficulty faced by this project, at least if we are to try to make the results of this work available to other institutions. While BYU was granted rights to the materials that we have, we do not have the right, for the moment, to distribute the application outside the university, illustrating the point that one of the biggest challenges faced in the digital age is that of digital rights management for intellectual property.

Nonetheless, it is useful to review from a technical standpoint the work that is going into the conversion of this software. First, the various text files are being converted using an approach similar to that used in the conversion of the

*HyperCard* exercises, a process in which the character representation used in the proprietary Windows font was converted to Unicode. The transcript of the video and corresponding time codes are being converted to the MPEG-7 Core Description Profile (CDP) mentioned earlier. In addition, we are producing tools that will enable the creation of new video materials using this same standard approach. The software for viewing is being created using Macromedia *Director* that uses the Macromedia *Shockwave Player* for cross-platform (Mac OS X and Windows PC) delivery. Delivery of this type of content under Linux is not presently possible, but the Linux community has started a petition, imploring Macromedia to release a compatible version of Shockwave.<sup>17</sup>

The dictionary entries were stored in the Microsoft-developed “rich text format” (RTF) and will be converted to a standard format based on the TermBase eXchange format (TBX). Also under investigation is the Open Lexicon Interchange Format (OLIF) as well as the Language Resource Management work of ISO TC 37/SC 4 N088.<sup>18</sup>

With the approach we are taking in our development, the dictionary content is being converted to an XML schema that is by definition display-independent, as opposed to the RTF approach that contained information for text presentation. This is one more example of the separation of content from strategies of instruction, presentation, or display, ensuring easy migration of this content from current delivery systems to those of the future.

## CONCLUSION

This article cites research and meta-analyses that demonstrate that online instruction can be effective for learning in general, for language learning, and for specific instances in Arabic and, furthermore, demonstrates how materials can be developed in a cost-effective manner. Successful implementation of the technology, however, requires the integration not only of pedagogical know-how, but also some level of expertise in software engineering if the capabilities of today’s hardware and software platforms are to be fully exploited. The advent of the Web provides materials developers the incredible possibility to take the dream that has been emerging for the instruction of Arabic over the past 30 years and finally make it a reality. Virtually all personal computers sold today come standard with a Web browser that will correctly render materials created in accordance with the Unicode standard. While Unicode does not solve all of our problems, it does remove many significant obstacles that have inhibited the sharing and interoperability of instructional interactions in Arabic. Furthermore, despite the fact that SCORM is a developing technology, it holds significant promise that we have been demonstrating in our work at BYU. Despite the proliferation of video formats, we are finding that the developing MPEG-7 standard will allow an effective approach for marking up video content that will ultimately allow video to be easily transcoded where necessary and run on various video playback systems, those that exist today and those that will be created in the future. By using Unicode, SCORM, and MPEG-7, we can be assured that the instructional materials we create for current platforms will not be obsolete with the development of new and

improved generations of hardware and software. Although a lot of technical work remains, we are finally at a point where we are limited not so much by technology, but rather by our imaginations and ability to make pedagogically sound instructional design decisions.

---

## NOTES

---

<sup>1</sup> The research and development on which this article is based has been made possible through generous funding from Brigham Young University, the National Security Education Program, and the National Middle East Language Resource Center.

<sup>2</sup> It is quite interesting to note that the technology for representing Arabic in the Abboud and Bunderson study involved a computer screen limited to 32 lines of 40 characters with interactions possible with a typewriter terminal connected to the computer or with input via a “light pen” that students pointed at the computer screen and pressed a button on its side. The computer was connected to a tape player that provided a limited level of access to audio.

<sup>3</sup> This software can be ordered at <http://creativeworks.byu.edu/hrc/index.cgi?userid=008-1073174136-278&search=&arabic=>.

<sup>4</sup> HRB Systems was a defense contractor that was purchased by E-Systems, which was later purchased by the Raytheon Company. Alpine Media is a company in Orem, Utah. For a brief overview of this movie series see <http://www.thejournal.com/magazine/vault/A58.cfm>.

<sup>5</sup> It is interesting historically to note that *Authorware* initially ran only on the Macintosh for development.

<sup>6</sup> See [www.humancomp.org/wincalis.htm](http://www.humancomp.org/wincalis.htm) for information on *WinCALIS*.

<sup>7</sup> *SuperMacLang* is available for download for \$25.00 at <http://schiller.dartmouth.edu/~SML/>.

<sup>8</sup> For information on *Libra/Gemini*, see <http://www.libra.swt.edu/New/default.html>.

<sup>9</sup> An overview of the *WebCALIS* beta version can be found at <http://www.humancomp.org/webcal.htm>.

<sup>10</sup> Readers can find information on *MaxAuthor* at <http://cali.arizona.edu/docs/wmaxa/>.

<sup>11</sup> For an excellent discussion of Unicode issues see <http://www.alanwood.net/unicode/index.html>.

<sup>12</sup> The term “rendering” is used to mean the preparation for display of the content elements that are stored using XML schema.

<sup>13</sup> The IEEE Learning Object Metadata standard is the product of Working Group 12 of the Learning Technology Standards Committee of the Institute of Electrical and Electronics Engineers, Inc. (IEEE, pronounced “I-Triple-E”). See <http://ltsc.ieee.org/wg12/>.

<sup>14</sup> Information on SCORM is available at <http://www.adlnet.org/index.cfm?fuseaction=scormabt>. Information on the IMS Content Packaging specification is available at <http://www.imsglobal.org/content/packaging/index.cfm>.

<sup>15</sup> The QTI Specification is described at <http://www.imsglobal.org/question/index.cfm>.



<sup>16</sup> For a very succinct yet thorough overview of MPEG-7 see <http://gondolin.rutgers.edu/MIC/text/how/mpeg7ref.pdf>. The full documentation for MPEG-7 can be purchased at the Web site of the International Organization for Standardization (ISO): (<http://www.iso.ch/iso/en/CombinedQueryResult.CombinedQueryResult?queryString=15938>).

<sup>17</sup> The petition is located at <http://www.petitiononline.com/linuxswp/petition.html>

<sup>18</sup> Information on these various efforts in standardization can be found at <http://www.olif.net/>, <http://www.lisa.org/tbx/>, and [http://www.tc37sc4.org/doc1/ISO%20TC%2037-4%20N088%20Lexical\\_Markup\\_Framework.pdf](http://www.tc37sc4.org/doc1/ISO%20TC%2037-4%20N088%20Lexical_Markup_Framework.pdf).

## REFERENCES

- Abboud, P. F., & McCarus, E. N. (Eds.). (1983). *Elementary Modern Standard Arabic* (Vols. 1-2). Cambridge: Cambridge University Press.
- Alhawary, M. T. (2001). Changing teacher roles and input-feedback medium: Authorware and Web-based assisted language learning of Arabic as a foreign language. In K. Cameron (Ed.), *C.A.L.L.—The challenge of change: Research and practice* (pp. 329-336). Exeter, UK: Elm Bank Publications.
- Allen, J. R. (1972). Current trends in computer-assisted instruction. *Computers and the Humanities*, 7 (1), 47-56.
- Allen, R. (1993). Proficiency and Arabic: The increasing role of technology. *Computer Assisted Language Learning*, 6 (1), 3-12.
- Allen, R., & Allouche. (1986). *Let's learn Arabic: A proficiency-based syllabus for Modern Standard Arabic*. Philadelphia: University of Pennsylvania Press.
- Atkinson, R. C., Attala, E. E., Blitzer, D. L., Bunderson, C. V., Charp, S., & Hirschbuhli, J. J. (1978). Where will computer-assisted instruction (CAI) be in 1990? *Educational Technology*, 18 (4), 60-63.
- Beheydt, L. (1987). The semantization of vocabulary in foreign language learning. *System*, 15 (1), 55-67.
- Belnap, R. K. (1993). Computer-assisted language learning and Arabic: The Brigham Young University experience. *Computer Assisted Language Learning*, 6 (1), 13-19.
- Belnap, R. K. (2001). Arabic CALL: Lessons from the past, opportunities for the future. In K. Cameron (Ed.), *C.A.L.L.—The challenge of change: Research and practice* (pp. 367-371). Exeter, UK: Elm Bank Publications.
- Blyth, C. (1999). Implementing technology in the foreign language curriculum: Redefining the boundaries between language and culture. *Journal of Educational Computing Research*, 20 (1), 39-58.
- Borrás, I., & Lafayette, R. (1994). Effects of multimedia courseware subtitling on the speaking performance of college students of French. *The Modern Language Journal*, 78 (1), 61-75.
- Brustad, K., Al-Batal, M., & Al-Tonsi, A. (1997). *Al-Kitab Fii Ta'allum Al-'Arabiyya: A textbook for Arabic (Pt. 1 & Pt. 2)*. Georgetown: Georgetown University Press.
- Bunderson, C. V., & Abboud, V. C. (1971). *A computer-assisted instruction program in the Arabic writing system* (Technical Report No. 4). Austin, TX: The University of Texas at Austin. (ERIC Document Reproduction Service No. ED052603)

- Burston, J. (2003). Proving IT works. *CALICO Journal*, 20 (2), 219-226.
- Bush, M. D. (1989a). Banquet Address—CALICO '89 at the U.S. Air Force Academy. *CALICO Journal*, 7 (1), 5-15. Also retrieved October 1, 2003, from <http://calico.org/journalarticles/Volume7/vol7-1/Bush.pdf>
- Bush, M. D. (1989b). Reasons the interactive videodisc industry should do windows. *Multimedia Monitor*. Retrieved November, 2003, from <http://arclite.byu.edu/digital/windows.html>
- Bush, M. D. (1991). Hardware for language training: Coping with confusion. *Applied Language Learning*, 2 (2), 77-91. (ERIC Journal Reproduction Service No. EJ445917)
- Bush, M. D. (2002). Connecting instructional design to international standards for content reusability. *Educational Technology*, 42 (6), 5-13.
- Chapelle, C. (1997). CALL in the year 2000: Still in search of research paradigms? *Language Learning & Technology*, 1 (1), 19-43. Retrieved October 1, 2003, from <http://llt.msu.edu/vol1num1/chapelle/>
- Chapelle, C. (1998). Multimedia CALL: Lessons to be learned from research on instructed SLA. *Language Learning & Technology*, 2 (1), 22-34. Retrieved October 1, 2003, from <http://llt.msu.edu/vol2num1/article1/>
- Chun, D. M., & Plass, J. L. (1996). Effects of multimedia annotations on vocabulary acquisition. *The Modern Language Journal*, 80 (2), 183-198.
- Chun, D. M., & Plass, J. L. (1997). Research on text comprehension in multimedia environments. *Language Learning & Technology*, 1 (1), 60-81. Retrieved October 1, 2003, from [http://llt.msu.edu/vol1num1/chun\\_plass/](http://llt.msu.edu/vol1num1/chun_plass/)
- Curtin, C., Clayton, D., Finch, C., Moor, D., & Woodruff, L. (1972). Teaching the translation of Russian by computer. *The Modern Language Journal*, 56 (6), 354-360.
- Cushion, S., & Hémar, D. (2002). Applying new technological developments to CALL for Arabic. *Computer Assisted Language Learning*, 15 (5), 501-508.
- Cushion, S., & Hémar, D. (2003). Designing a CALL package for Arabic while learning the language ab initio. *Computer Assisted Language Learning*, 16 (1), 259-266.
- Day, N. (2001, November 12). MPEG-7 tackles multimedia content. *EE Times*. Retrieved October 24, 2003, <http://www.eet.com/story/OEG20011112S0058>
- Dodds, P. (2001). *The SCORM Overview. Advanced distributed learning network, sharable content object reference model (SCORM) Version 1.2*. Retrieved November, 2003, from <http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=40&bc=false>
- Doughty, C. J., & Long, M. H. (2003). Optimal psycholinguistic environments for distance foreign language learning. *Language Learning & Technology*, 7 (3), 50-80. Retrieved November, 2003, from <http://llt.msu.edu/vol7num3/doughty/>
- Dorwick, T. (2002, March). *The changing paradigm: Are we ready for it?* Keynote Banquet Address presented at the annual meeting of CALICO, Davis, CA.
- Ehsani, F., & Knodt, E. (1998). Speech technology in computer-aided language learning: Strengths and limitations of a new CALL paradigm. *Language Learning & Technology*, 2 (1), 45-60. Retrieved December 30, 2003, from <http://llt.msu.edu/vol2num1/article3/>

- Garrett, N., & Hart, R. S. (1987). Foreign language teaching and the computer. *Foreign Language Annals*, 20 (5), 441-443.
- Glennan, T. K., & Melmed, A. (1996). *Fostering the use of educational technology: Elements of a national strategy* (MR-682-OSTP). RAND. Retrieved October 1, 2003, from: <http://www.rand.org/publications/MR/MR682/contents.html>
- Hayes, J. (2002). *K-12 technology update for CoSN. Quality education data*. Retrieved October, 2003, from <http://www.k12schoolnetworking.org/2002/presentations/hayes.ppt>
- Humanities Computing Laboratory. (2003). *WinCALIS computer assisted language instruction system*. Retrieved October, 2003, from <http://www.humancomp.org/wincalis.htm>
- Jorstad, H. L. (1980). New approaches to assessment of language learning. In T. H. Geno (Ed.), *Northeast conference reports: Our profession: present status and future directions*. Middlebury, VT: Northeast Conference on the Teaching of Foreign Languages, Inc.
- Krashen, S. D. (1989). Language teaching technology: A low-tech view. In J. E. Alatis (Ed.), *Georgetown University Round Table on Languages and Linguistics 1989* (pp. 393-407). Washington, DC: Georgetown University Press.
- Kulik, J. A., & Kulik, C. L. C. (1987). Review of recent research literature on computer-based instruction. *Contemporary Educational Psychology*, 12, 222-230.
- Lee, K. (2000). English teachers' barriers to the use of computer-assisted language learning. *The Internet TESL Journal*, 6 (12). Retrieved in October, 2003 from <http://iteslj.org/Articles/Lee-CALLbarriers.html>
- Manjunath, B., Salembier, P., & Sikora T. (2002). *Introduction to MPEG-7: Multimedia content description interface*. West Sussex, England: John Wiley & Sons, Ltd.
- Market Data Retrieval. (2003). *Wireless network technology skyrockets on college campuses*. Press release dated February 21, 2003, citing *The college technology review, 2002-2003 academic year*. Retrieved November, 2003, from [http://www.schooldata.com/pdfs/colltechpr\\_03.pdf](http://www.schooldata.com/pdfs/colltechpr_03.pdf)
- McArthur, D., & Lewis, M. (1998). *Untangling the Web: Applications of the Internet and other information technologies to higher education* (MR-975-IET). RAND. Retrieved October 1, 2003, from: <http://www.rand.org/publications/MR/MR975/>
- Merrill, M. D., Li, Z., & Jones, M. K. (1992). Instructional transaction shells: Responsibilities, methods, and parameters. *Educational Technology*, 32 (2), 5-27. Retrieved January 14, 2004, from: <http://www.id2.usu.edu/Papers/TxParameter.PDF>
- Nielsen H. L., & Carlsen, M. (2003). Interactive Arabic grammar on the Internet: Problems and solutions. *Computer Assisted Language Learning*, 16 (1), 95-112.
- Ruplin, F. A., & Russell, J. R. (1968). A type of computer-assisted instruction. *The German Quarterly*, 41 (1), 84-88.
- Scebold, C. E. (1983). Equal time. *CALICO Journal*, 1 (1), 13-14. Retrieved October, 2003, from <http://calico.org/journalarticles/Volume1/vol1-1/Scebold.pdf>
- Tansey, F. (2004, January). Hoping for a year of standards in 2004. *Syllabus: Technology for Higher Education*. Retrieved January 8, 2004, from [http://www.syllabus.com/news\\_article.asp?id=8757&typeid=155](http://www.syllabus.com/news_article.asp?id=8757&typeid=155)

- Unicode Consortium. (2003a). *What is Unicode?* Retrieved December, 2003, from <http://www.unicode.org/standard/WhatIsUnicode.html>
- Unicode Consortium. (2003b) *History of Unicode*. Retrieved December, 2003, from <http://www.unicode.org/history/>
- US Census Bureau. (2001, September). *Home computers and Internet use in the United States: August 2000* (Special Studies). Retrieved November 14, 2003, from <http://www.census.gov/prod/2001pubs/p23-207.pdf>
- Wong, W., & VanPatten, B. (2003). The evidence is IN drills are OUT. *Foreign Language Annals*, 36 (3), 403-423.
- Zhao, Y. (2003). Recent developments in technology and language learning: A literature review and meta-analysis. *CALICO Journal*, 21 (1), 7-27.

### **AUTHORS' BIODATA**

Michael Bush has a wide range of experience in the uses of high technology in education to solve problems of various types over the past 35 years. He received his Ph.D. in Foreign Language Education and Computer Science from The Ohio State University in 1983 and served in various capacities in the Department of Foreign Languages at the United States Air Force Academy, playing an instrumental role in the installation of the Academy's unique, interactive videodisc-based language learning center. Since his retirement from the Air Force in 1992, Professor Bush has been serving as Associate Professor of French and Instructional Psychology and Technology at Brigham Young University where he is currently the Associate Director of the Center for Language Studies.

Jeremy Browne is a doctoral student in the Instructional Psychology and Technology Program at Brigham Young University, where he previously completed his undergraduate studies in French. He has had a wide range of experience in programming, instructional design, graphics design, and software development.

### **AUTHORS' ADDRESS**

Michael D. Bush  
Brigham Young University  
Provo, UT 84602  
Phone: 801/422-4515  
Fax: 801/224-8881  
Email: Michael\_Bush@byu.edu

Jeremy M. Browne  
Brigham Young University  
Provo, UT 84602  
E-mail: browne@byu.edu