ReCALL **15** (2):251–268. © 2003 Cambridge University Press
DOI: 10.1017/S0958344003000922 Printed in the United Kingdom

CALL, commercialism and culture: inherent software design conflicts and their results

MICHAEL SHAUGHNESSY
Washington and Jefferson College, 60 South Lincoln St.,
Washington, PA 15301, USA
(*Email: mshaughnessy@washjeff.edu*)

Abstract

This article investigates commercial software design practices as they specifically relate to foreign language education. Commercial educational software companies currently produce the majority of language learning software available on the market. Commercial ventures producing CALL software share many design practices that call into question their educational validity. The design practices of commercial CALL software companies are incongruent with the goals of foreign language education. The problems associated with commercial CALL ventures can be seen in the cultural aspects of the programs, particularly when dealing with issues of cultural authenticity and representation. Practices that create these cultural problems are investigated and outlined in this article. The results of these problems are also highlighted to allow for better identification of problematic design practices. The identification of these problems is associated with educational software evaluation theory. A brief outline of educational software evaluation theory is given, as well as a proposal for a new framework for CALL software evaluation that incorporates issues of representation to better address the inaccuracies found in many commercial CALL software products.

1 Introduction

In CALL, there is a constant need to show the connection between theory and practice. Theory and academic research into the relationship between educational technology and language acquisition must ultimately be applicable in a real-life learning environment. In many ways, software is both a manifestation of years of research and an entity that often ignores academic theories of education/second language acquisition. I am arguing that, more often than not, it is the commercial element in CALL that ignores research. Commercial educational software companies produce more CALL software than academic or educational institutions.[1] By producing more, these companies are by fiat leaders, and their practices may be considered the standard in the industry. The design

[1] This claim comes from analyzing the *Germanware* database, a web site highlighting educational software for the study of German.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

process in commercial CALL software has developed to a point at which a certain uniformity exists, regardless of national boundaries.[2] In a way, CALL software has been globalized, so that production and distribution can seamlessly occur despite country of origin or target consumer. It is this uniformity in design that ought to be analyzed to see how the current practices among varying companies affect the overall quality of CALL software.

These claims of a gobalized CALL software market come primarily from two sources: first hand experiences with the commercial CALL design process, and from an analysis of a large sample of commercially available CALL software products. The author of this article has been involved on the commercial side of CALL software with two large publishing companies. Several 'practices' of these production companies were noticed in other programs, produced by different entities in different countries. The methodology for identifying shared design practices is therefore both inductive and deductive. The methodology is inductive in that first-hand experience is used to explain how one example can be applied to other products. It is deductive, as a number of other products are analyzed, to refer back to the specific causes of problems.

Primarily, this article will investigate how instructional software design affects *cultural representation* within programs for second language acquisition. The significance of culture within CALL was given by Cameron (1998:2) "It is virtually impossible to devise a CALL program which does not have some connection with cultural issues. By its very definition, CALL is about language and language is a cultural issue par excellence." Cameron then goes on to cite examples of how CALL researchers have incorporated 'culture' into the essence of CALL. My article also assumes that CALL and culture are inherently connected, but seeks to explore the role of culture within the specific context of commercially designed CALL software. Cameron stated (1998:6) "More research needs to be effected to ensure that the cultural context of new programs in on the right line." This article argues that it is the overall design process of commercially produced CALL software programs that creates an incongruence between the cultural context and the cultural reality of the L2 system.

Unfortunately, there has been very little written to specifically address the influence of commercially designed CALL software on the academic field. A laundry-list of articles and books dealing with CALL software design in general would not be appropriate here, as I argue that there is a stark difference between the products produced by commercial ventures and products produced by academic linguists.[3] Even Myles's (1998) article, 'The language Learner and the Software Designer: A Marriage of True Minds or ne-er the twain shall meet?', although analyzing design issues of CALL software within the framework of commercially produced products, still does not probe the underlying process of *commercial* CALL software design. CALL researchers seem to have a long history of using commercial CALL software for examples in their research, yet they do not specifically focus on the nature of that commercially designed software (England, 1984; Hannafin & Peck, 1988; Laurillard, 1991; Balajthy, 1996, and Boling & Soo,

[2] While I argue that the United States has great influence over the origin of the design process, European companies employ the same methods for designing CALL software.

[3] See "CALL Software Design and Implementation" by Gimeno-Sanz, Davies and Bangs for an example of an academic approach, as well as a thorough bibliography.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

1999.) The most relevant article I have found is by Brett and Nash (1999). Their description of the design process raises several issues of commercial viability that will be discussed in this article. However, this article will argue that issues surrounding commercial viability may be contradictory to the goals of a culturally authentic CALL experience.

Admittedly, the lack of attention to this issue does present a lack of corroboration for the arguments made here, which could pose a problem for the authority of this article. Since the individuality of the commercial design process of educational software is not explicitly discussed by the academic press, this article is breaking new ground, with little support for the claims it makes. On the other hand, the lack of attention paid to this subject is justification enough for an article that will primarily illustrate through examples, that I hope are often self-explanatory.

The inherent conflict I see is that most commercially produced software programs are not designed for a single language, but have shared components, so that the company can simultaneously develop learning packages for several languages.[4] This holds true for more commonly taught languages like English, French, German, Italian, and Spanish.[5] The design practices investigated in this article are consistent with software simultaneously developed for these languages. Software programs for less-commonly taught languages are usually developed as singular entities, and do not employ the commercial design practices discussed here. The purpose of this article, however, is to identify commercial *design practices* in CALL and show the results of these practices. I will also argue that these practices are inherently problematic, due to the nature of language and cultural education. In the end I will propose methods of identifying such commercially produced products, and will speculate on possible ways for commercial ventures to improve their design process to alleviate some of the items that I consider to run counter to the very nature of CALL.

2 The nature of commercial CALL software design

History has proven that there are only a few language professionals, be they professors of language and literature or experts in CALL, who (on their own) have the ability to produce high quality software for language learning with a wide distribution. Most large-scale projects involve teams of content providers and programmers, consuming time and human capital (Brett & Nash 1999, 19–20) Perhaps it is partially for this reason that the majority of CALL software published is commercial in nature. This, of course, does not preclude language professionals from working on commercial software projects and participating in software development. In fact, quite the opposite is true. Language professionals are almost always employed by commercial ventures as consultants, normally under the title of "language content provider". The nature of commercial considerations within the educational software market limits their 'professional' role in the overall design of CALL software. The function of the "language content

[4] This is mentioned in Brett & Nash (1999: 31): "The tools allow unlimited production capacity, and the production of multimedia language learning material in other foreign languages such as French, German, and Spanish has begun."

[5] See www.auralog.com, one of the world's leading commercial CALL software producers.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

provider" in commercial CALL production places the language expert in a secondary position. The weakened role of these language professionals in commercial CALL software development poses some inherent problems associated with design. These design problems also produce some typical results that call into question the validity of commercial CALL software effectiveness. Two fundamental practices in commercial CALL software production associated with the weakened role of language professions that tend to create inherent problems are *content/delivery isolation* and *software cloning*.

2.1 Content/delivery isolation

CALL software design, particularly in a commercial environment, involves the use of context experts and programmers. Content experts are the language professionals responsible for providing error-free language content for a pre-designed user-interface. The user-interface is the system that delivers the provided content. The interface is programmed in such a way as to work with a variety of possible content sources. Commercial software design practices often employ a strict separation of these two elements to better 'internationalize' the product. This approach is taken directly from the world of software engineering. Karkaletsis, Spyropoulos and Vouros (1998:312) use an 'internationalised approach,' to support multilingual functions in software development. "According to this approach, the culture, language, and user-dependant parts of the software application are isolated from the software core during development." The very nature of this relationship highlights the inherent problem within the educational context. The content experts provide content to a pre-designed shell, thus allowing the technical side of the software to pre-determine educational possibilities. The language content expert is asked to provide language only, and has little say as to the overall instructional design. Dudley-Maring and Owston (1987) imply that this leads to poor instructional content: "the most common explanation for this situation is that most software is written either by persons with expertise in computers but not in education, or expertise in instructional theory but not computers." This implication here is that there is a negative correlation between a software-engineering approach to CALL design and the educational validity of the product.

In order for there to be a negative correlation, reoccurring instances of poor educational content must occur. I argue that this isolation of delivery and content frequently produces inaccuracies of a cultural nature. The cultural inaccuracies that arise from content/delivery isolation are often not associated with the syntactic accuracy of the text. The text is, after all, written in most cases by a language expert. Inaccuracies associated with textual elements in programs can be seen in the omission of certain special characters associated with the language, or even the occasional inclusion of special characters that are not associated with the target language.[6] These types of mistakes probably stem from the programmer including these special characters at a later date. The isolation of delivery and content precludes the content expert from catching this mistake and usually

[6] This is evident in two textbook ancillaries for introductory German language education: *Deutsch Na Klar!* with omission of capital umlauts in the entire program and *Treffpunkt Deutsch* for inclusion of 'e' and 'i' umlauts.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

a formative or summative evaluation process will not notice these details of a minute cultural nature.

Most cultural inaccuracies in commercially produced CALL software can be found in the visual elements of the program.[7] Visual representation in CALL is becoming more common. Garret (1998: 16) states, "In many CALL cultural materials the visual context for and the visual aspect of communication are emphasized far more than the linguistic expression." Problematic, or culturally inauthentic visual materials stem from content-experts requesting specific images from the programmer, or the programmer requesting specific text from a pre-chosen visual. The isolation of content and delivery providers creates a situation where the wrong person is asked to provide an aspect of the program for which they have little or no expertise.

A commercial product needs to have the rights to reproduce and sell these images, so the programmer either has to create the image or purchase the rights to an image. This creates a dilemma between authentic materials and cost-effective materials. For reasons of cost, visuals are often chosen from sources that are not in the target culture of the product. Below is an example of a German language-learning package that is seeking to teach gas station vocabulary (*Speak German*, 1996). The gas station (see Figure 1) is obviously chosen from a non-German context. The price (marked 1) indicates this. There is no (German-speaking) context where gasoline could be priced at '39,9' per liter or gallon. The second identifier of the incongruence between content and delivery is the English text 'air' (marked 2). While the instructions were probably given (by the programmer to the content provider) to replace all of the English words, this was overlooked in one case and the programmer simply did not notice the cultural aspect of the gasoline price. By analyzing the image, the process for creating this exercise can be deduced. The programmer probably owned the rights or created an English-speaking gas station environment, then requested a word for word translation

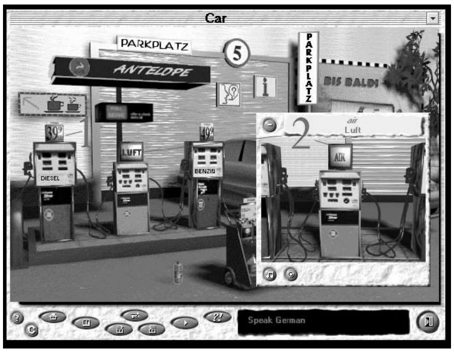


Fig. 1. Speak German.

[7] The problematic nature of visuals is closely tied to the commercial nature of proprietary imaging. Educational researchers have noticed the importance of authentic imaging. Abramson (1998:61) states the obvious, "All images should be clear and relevant to the content, and not simply decorative" (61).

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

of the items in the picture. These were then pasted in over the English words to create a 'German-speaking' environment. Unfortunately, this practice is not limited to this one software package, but is commonplace among commercial language-learning software developers.

The inherent pedagogical problem with content/delivery isolation is that the content experts often play a diminished role in the design process. Content experts are hired to provide language content only. They are often expected to provide this content based on the user interface and have little input into the construction of the program. Thus technology is determining the content, and mistakes of a linguistic or cultural nature are predictable.

The separation of content and delivery in a commercial setting may not be avoidable, however. The length of time to produce quality programming and the length of time to produce accurate content writing would force most projects to last over two years. This would mean producing software that is considerably out of date by the time of publication. Only by producing software in a team can a company expect to have a (technically) high-quality product in a reasonable amount of time. Academic linguists hired on as consultants for commercially developed software products are relatively speaking not low-paid workers. Commercial ventures seek to limit the involvement with the project by asking such academic linguists to provide information on demand, rather than comprehensive consultation services. The programmers come up with lists of needed language content. This speeds up the software production process, and significantly limits any additional involvement on behalf of the academic linguist.

2.2 Cloning

In conjunction with the frequent practice of content/delivery isolation, a process fairly unique to CALL software also exists: *Cloning*. While this term has been used to describe the process of software copyright infringement through reverse engineering, I am using it in a new context involving the purposeful use of user-interface adaptation in educational software across several languages.

Cloning involves the development of one user interface that is to be adapted across several languages. This process has two motivations:

- To develop a user-interface for a number of languages.
- To be economically viable.

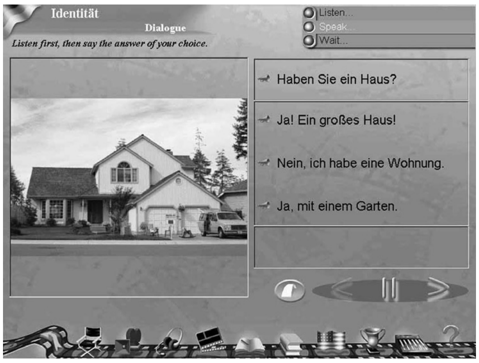
Currently in software engineering, there are two practices that closely resemble what I am calling 'software cloning': *Internationalization and Localization*. Carey (1998: 449) summarizes the two practices:

> *Internationalization* is the process of extracting the domestic context from a package. The end goal is to end up with a sort of generic product with an appendix or attachment that contains all the culturally specific items.
>
> *Localization* is the opposite of internationalization. Localization takes a generic product and adds features and elements to fit the target culture and market.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

While software cloning mostly resembles software internationalization, the educational nature of cloning makes it wholly unique. The unique structure of different languages, from the number of noun genders, to the use of different cases would imply that the instructional design of exercises would also be culturally significant to the programmer. Not only are images, subject headings, color schemes, and videos all shared in order to reduce costs, but the blueprint for learning a language remains the same. The internationalization of software programs implies an adaptation of cultural elements. In CALL, the basic construction and design of the program could be argued to be culturally significant. Since the core of CALL programs is significant, I argue that such programs are not internationalized, but cloned since they share a similar base structure.

Interestingly enough, the problems associated with cloned products have been virtually ignored by the academic press, with a few exceptions that tangentially identify this common practice. Again, the most obvious element in cloned CALL products is the visual aid. Waltje (2001) comments in a software evaluation, "…the program has the unfortunate tendency to employ culturally inauthentic materials a bit too frequently." We can see this in Figure 2. Waltje comments on the above program using a generic house. Clearly, the house does not come from a German-speaking environment. The mini-van, the two-car garage, and the basketball hoop outside are indicators of an ideal-



Identität

Dialogue

Listen first, then say the answer of your choice.

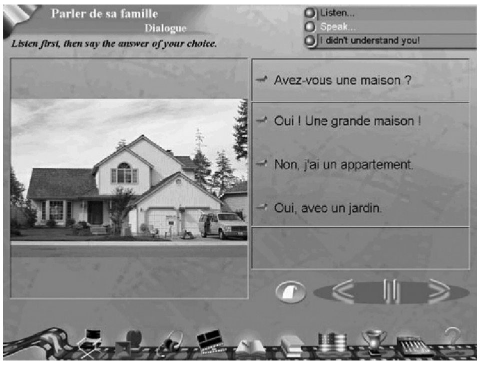
Haben Sie ein Haus?

Ja! Ein großes Haus!

Nein, ich habe eine Wohnung.

Ja, mit einem Garten.

Fig. 2. *Tell me More!* German.



Parler de sa famille

Dialogue

Listen first, then say the answer of your choice.

Avez-vous une maison ?

Oui ! Une grande maison !

Non, j'ai un appartement.

Oui, avec un jardin.

Fig. 3. *Tell me More!* French.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

ized home from the United States. To better understand the nature of cloned products, it is necessary to find the corresponding cloned version of the program to see what changes, if any, have been introduced. As the next image indicates, represented by the French version, the software program has not been internationalized, but truly cloned. The sentences are word-for-word translations of the German version. Here, the content provider was not even asked to come up with original material, but simply required to provide a translation (*Tell me More! French,* 2000) – see Figure 3.

Cloned products frequently only ask content providers to translate headings and individual words. Viewed from an educational perspective, this process is undoubtedly problematic. One needs to simply imagine taking a program designed to teach English grammar, then cloning it to teach Russian grammar. This basic presumption about the transferability of languages and language structure into a learning environment is inaccurate. While the cloning of grammar exercises is less common, even seemingly harmless products like cloned vocabulary programs produce striking inaccuracies.

While the goal of commercially produced CALL products is to produce a 'generic' user-interface core on which to build a learning environment across different languages, their construction is anything but 'generic'. The source interface of the program fundamentally affects the cloned product. If the original interface is designed for an environment in the United States, then the cultural aspects of the U.S. will be transferred onto the target languages. Incidentally, the screen shots above teaching German and French through the eyes of the United States originated from a French company. The commercial reality that most license-free imagery comes from the United States transfers into the probability that most commercial CALL software will present the syntactic content in one language, but a visual conception of the world that clearly reflects the United States. Unfortunately, this practice is not limited to the design practices within one individual country. Besides the example from a French company, and further examples given in this article from U.S. companies, a recent review shows that a company from the UK also employs cloning for distance learning software (Shaughnessy, 2002). This problem is not associated with the country of origin, but with the commercial association of the product.

One of the goals of foreign language teaching is to teach the authentic culture of the target language. In cloning, students are often presented with an ethnocentric view of the world that gives them the impression that simple word translations will allow them to communicate effectively in the target language. The previous example of the gas station highlights this view well. Perhaps a gas station might not differ sharply among cultures, but some of the more significant topics such as travel and food do differ drastically.

3 Results of these design problems:

While poor software is not guaranteed by products developed with a cloning method or content/delivery isolation, there are some typical results of these techniques that can be identified in language learning software. Analyzing German language learning software provided many examples of the following problems associated with cloning and content/delivery isolation. The examples provided below best exemplify the problems.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

3.1 Unreality[8]

Unreality is the misrepresentation of the target culture, either by providing incorrect or incomplete information regarding a subject. In foreign language education, this deals specifically with the issue of providing authentic materials. In the program below (*Think and Talk German,* 1996) ironically titled "A little geography", a map is given of Germany that roughly represents the map of Nazi occupied Europe. This is most likely the result of a content expert requesting a map of Germany to accompany the text in this given exercise. The programmer, isolated, and not having knowledge of the target language, inserted a map of Germany for which the company probably owned the rights. Simple editing and review of the product would have solved this problem, but content/delivery isolation often produces mistakes of this nature as this map and the gas station example highlight (see in Figure 4).

3.2 Culturally generic software

Culturally generic software is a prime identifier of cloned software. The software is developed in a culturally generic manner,so as to 'fit' other languages. As previously mentioned, the generic nature of the program often highlights the inherent cultural bias present in the program as manifested in the visual elements. Myles (1998:39) hinted at this in stating that "CALL software writers, especially if non-linguists appear to regard vocabulary as merely some body of words to be thrown at the learner." In the example below, a vocabulary list was given for food and restaurant words. Pictures were matched to the words and the word is pronounced for the user. In the graphic, marked 2, the picture is a culturally generic restaurant environment in which a person appears to be setting a table. Here this is given as a representation of the word 'busboy' (3). The generic picture is used despite the target language. It is unclear whether the picture actually represents a 'bus-boy' or not. The point is that when inserting license free images, the image need only roughly correlate to the text. More importantly, in a German-speaking

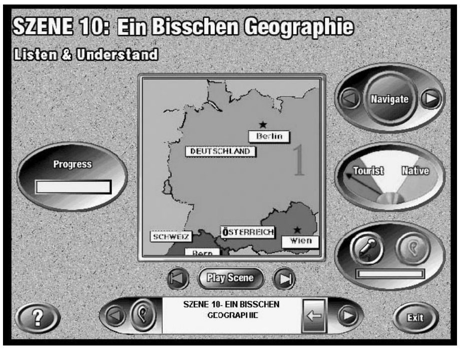
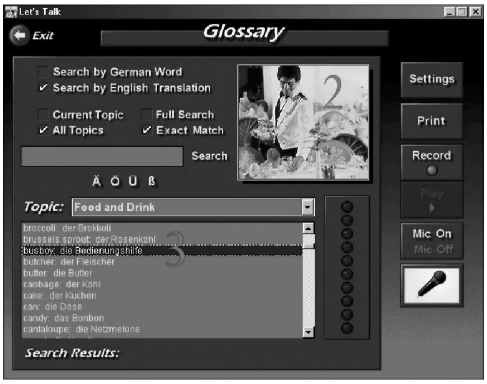


Fig. 4 . *Think and Talk!* German.

[8] The terms Unreality and Linguistic bias were cited by McCormick & Boney (1990), in which they cited an original study by Sadker and Sadker identifying bias in instructional materials.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press



□ Search by German Word
☑ Search by English Translation

□ Current Topic
☑ All Topics

□ Full Search
☑ Exact Match

Topic: Food and Drink

broccoli: der Brokkoli
brussels sprout: der Rosenkohl
busboy: die Bedienungshilfe
butcher: der Fleischer
butter: die Butter
cabbage: der Kohl
cake: der Kuchen
can: die Dose
candy: das Bonbon
cantaloupe: die Netzmelone

Figure 5 *Let's Talk!* German.

environment, there is no such word for 'busboy.'[9] In Germany, as in much of Europe, the server (or waiter/waitress) is responsible for clearing and setting the table. The list comes from the cultural environment of the United States and was matched across the languages (*Let's talk German*, 1996) – see Figure 5.

The structure and design of CALL software can clearly affect the educational aspects. By ignoring the cultural difference associated with specific topics, students only learn vocabulary from one cultural reference point. As in the case of the gas station example, the content expert is only expected to translate individual words, not knowing the context in which they will be used or the cultural significance of the exercises. The isolation of the content expert precludes error-correction. While print texts have generally abandoned ethnocentric teaching, the new media, due to the nature of software design, may be resurrecting old problems.

3.3 Diversity problems

Diversity problems arise from a lack of a proper content expert or recycling of older texts. When recycling older texts, both the cloning and content/delivery isolation problems climax due to the fact that the software engineer is ultimately responsible for cultural and linguistic accuracy. Negative examples of this can be categorized as follows: *Linguistic Bias* and *Sexist/Racist elements* in software.

3.3.1 Linguistic bias

Linguistic bias is a relatively old concept that is turning up again in the new media. Particularly in commercially produced CALL software programs, texts feature women in traditional roles. While it probably need not be stated, this transmits a linguistic bias to the learner of the foreign language. Particularly in languages where words differ based on the gender of the person, placing women in traditional roles limits the learner

[9] This word has been invented to describe the job of clearing a table in a restaurant, commonly associated with the United States. This word can be found in a German-Speaking environment when dealing with foreign internships, many with the Disney Corporation.

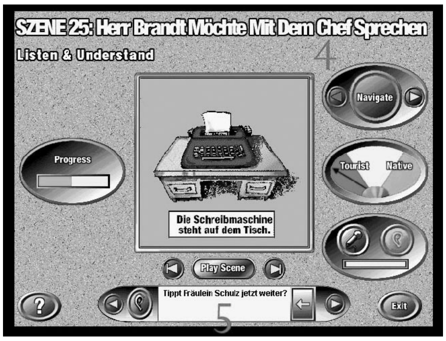
https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

to masculine nouns when using nouns involving professions: doctor, lawyer, boss, etc. The program below reserves all positions of power for men. Marked 4 in Figure 6, the *Chef* or ‘Boss’ is always a masculine person while the female is relegated to ‘typing’ (*Think and Talk German*, 1996:5).

3.3.2 Sexist/racist elements

To be optimistic, it should be assumed that sexist and racist language could only be a product of severe inexperience with the target language or the recycling of much older texts. Below is an example that includes both. This section of the CD-ROM appears to teach people how to mock the Japanese. Despite the possible older text, multimedia is added to complete the effect. Stereotypical ‘Asian’ music plays in the background while a European (denoted by the purposefully affected Japanese accent) dresses up as a Japanese person and says, “I am Mr. Nakumura. I am Japanese! Ha! Ha!”. (*Think and Talk German*,1996: 6). This is shown in Figure 7.

At the bottom of the screen, (marked 7) the use of the diminutive *Fräulein* for woman (Frau) indicates no sensitivity to the fact that the word *Fräulein* is considered professionally unacceptable and is considered to be socially unacceptable in many regions of the German-speaking world. Any contemporary German or student of German would be able to identify this. You can see this example in Figure 7.



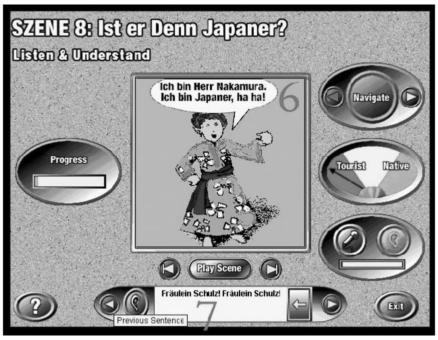
SZENE 25: Herr Brandt Möchte Mit Dem Chef Sprechen

Listen & Understand

Die Schreibmaschine steht auf dem Tisch.

Tippt Fräulein Schulz jetzt weiter?

Fig. 6. Think and Talk German.



SZENE 8: Ist er Denn Japaner?

Listen & Understand

Ich bin Herr Nakamura. Ich bin Japaner, ha ha!

Fräulein Schulz! Fräulein Schulz!

Fig. 7. Think and Talk German.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

4 Proposed solutions

The irony of the situation facing CALL now is that computer software was once perceived to be better than traditional materials due to the attention paid to content. Bright (1983:145) writes, "Computer-assisted instruction materials typically have an advantage over traditional materials in several respects. First, the content usually has been carefully structured so that the particular objectives are more likely to be attained." The drive and demand for software has created a commercial market where new programs must be created every year. It is unlikely that full-time language professionals will simultaneously work as content experts and programmers due to time constraints and the economics of the situation. There are a few dedicated language teachers with the knowledge of programming who are able to create quality software. Generally, qualified language professionals in control of both content and delivery will not face the content-related problems that commercial endeavors will. One only needs to look at a project supported by the European Union's Socrates/Lingua programme called LINC (Language Interactive Culture). LINC was developed across languages, yet retains cultural authenticity through the use of authentic materials.

However, this leaves the rest of the software market open for these mistakes. There is, I believe, an inherent probability that educational software will tend to be commercial in nature. Merrill *et al.* (1996:109) cite this probability through the claim, "Each hour of computer-assisted instruction developed by a commercial software house may take up to 400 hours in development time."

Merril *et al.* cite this as the reason why educators should not be programmers, but "should be educated evaluators of existing software." Levy (1999:98), specifically addresses the commercial nature of CALL software and the importance of evaluation, "Evaluation in CALL should begin by carefully reviewing existing programs and learning environments, including those produced commercially, and reflecting upon their content, structure, and design." For language professionals to be able to exercise influence over the commercial CALL designers, they need to become better software evaluators. This presumes that there is currently a problem with existing methods of educational software evaluation. Simply put, the problem lies in the fact that these types of mistakes are rarely noticed when CALL software programs are reviewed or evaluated. Emphasis is placed more on the technological side of software engineering, rather than on issues of content or cultural accuracy. The program including a map of Nazi-occupied Europe and the sexist and racist elements was described in the following way, "By blending sound, graphics, animation, and the enormous storage capabilities of CD-ROMs, these programs can offer something that approximates a real language experience." (Baig, 1995) This excerpt of a review highlights the trend over the past twenty years to associate quality CALL software with newer technical features. While there are many different evaluation methods for CALL software,[10] this article focuses on evaluation within the context of software design. Within software design, there are two primary evaluation methodology categorizations: *formative* and *summative* evaluation. In *Formative Instructional Product Evaluation* (1974:5), Lawson lays the basic groundwork for formative evaluation theory, which subsequently influenced theorists

[10] Selected Bibliography for CALL Evaluation. 24 December 2001 http://nflrc.hawaii.edu/aboutus/ithompson/flmedia/evaluation/callbib.htm.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

dealing with educational software: “Formative product evaluation can be defined as the appraisal of instructional sequences and materials during their stages of formulation and development.” Whereas formative evaluation is the process of evaluation during the design/development process, summative evaluation is the evaluation of the finished educational-software product. Price (1991:106) highlights the difference between formative and summative evaluation methods for educational software.

> “Formative evaluation should be more or less ongoing throughout the development process. Evaluations are made and revisions incorporated into a developing program with the goal being to make the lesson as good as possible by the time it is completed. Summative evaluation is usually conducted for the purpose of making major decisions concerning the entire program and is not conducted to determine specific features for modification or revision. It is typically used to make decisions regarding adoption or publication.”

My proposed solution begins with improved evaluation methods during and after the process of software development. Since the problems arise due to the design process, the attention must also be applied to the design process. Despite being a process that will ultimately lead to a better product, this formative product evaluation has a tendency to focus on the negative aspects of a given program. Ilgen-Lieth and Harzen (1987 I:16) state, “The purpose of formative evaluation is to identify weaknesses in instructional materials, to revise and improve them.” Most design evaluation methods focus on the formative aspect, in the hope of producing a product without flaws (Preece & Jones, 1985; Russell & Blake, 1988; Fidelman, 1990; 1993). Dealing specifically with CALL, Levy (1999: 99) highlights the importance and role of formative evaluation:

> “Formative evaluation of this kind is potentially most important in CALL program design. It can facilitate fast testing and evaluation of specific design elements. If a particular design is not successful it can be quickly discarded or refined and improved, then tested again until an optimal design is achieved.”

Despite the hopes for an optimal design through formative evaluation, there are many programs that exhibit inauthentic cultural representations, thus the need for a new method of software evaluation. In order to effectively incorporate new formative evaluation methods in commercial CALL software design, the relationship between the programmer and the content provider must radically change. There must be an improved line of communication between the educator and the commercial software designer. In some cases, this may involve direct involvement between the software designer and the teacher in the field (Krawchuk, 1998; Klemes *et al.* 1998). Within the design process itself, the content expert and the programmer need to re-think their roles, so that culturally inauthentic materials do not become the norm in CALL. One of the main questions surrounding CALL is whether it truly is effective or not. If it is to be effective, it must begin with being authentic. As Shen and Mo (1995:12) highlight in issues surrounding authenticity in children’s books, it is not merely about being sensitive to other cultures through the use of ‘nonstereotypes’. Authenticity within CALL is about the complete and accurate representation, both linguistically and visually, of the target language and culture.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

4.1 An intrinsic evaluation method

Within any educational software evaluation method, there must be a rationale and implementation philosophy to accompany it. A mere checklist for authentic materials will not serve the field well at all (McDougall & Squires, 1995). CALL software evaluation is inherently more flexible that general educational technology. Pederson (1987) highlights the general flexibility of CALL evaluation methods by the number of varying approaches available to evaluate CALL software. Examples abound of how CALL, despite an obvious and expected focus on technology, has repeatedly included issues of instructional content into evaluation methods. A prime example of this is the software review guidelines from the *CALICO* journal software evaluation guidelines. Despite placing a significant amount of importance on the technological features of the evaluated program, the guidelines also clearly state:

> No matter how technically brilliant a program may be or how rich the activities it provides, if its methodology is dubious, if it fails to adhere to its avowed instructional approach, or if it pays insufficient attention to linguistic accuracy or sociocultural authenticity, then it will be of limited usefulness.

In this sense, this methodology will be summative in nature, in that the products will be analyzed after the process of development. Despite the claims of some to focus on accuracy in cultural representation, there is still a tendency among many CALL researchers to focus on the technical side of design, and less on the humanistic side. Hom (2001) characterizes the process of design by stating that it "refers to user-friendliness and flexibility." As I argue, a user-friendly program can still be culturally inauthentic. Similarly, Balajthy (1996) provides a large study of commercially produced programs, but leaves out any mention of the issues brought up in this article.

The nature of CALL requires incorporation of both the cultural and the technological. Although many CALL researchers have clearly thought about issues of culture and representation in their research, the fact that most programs on the commercial market have generic cultural representations, which are often the same from language program to language program, highlights the necessity to incorporate this material at a deeper level. The comment from the *CALICO* guidelines is truly insightful, but it needs to be applied as an overreaching concept. If these points are raised at a deeper level, this will directly affect the commercial CALL software design process. A CALL design process that focuses on technology and less on education is partially responsible for the misrepresentation. An example of this technical approach is represented by the software engineering perspective in Karkaletsis *et al.* (1998). In seeking a better design model for international software, they propose "a methodology for the creation ion of a language independent knowledge base (KB) which can support the development of multilingual and user-tailored interfaces." Current design practices by many CALL commercial ventures produce a 'language-independent' user-interface. For CALL, the nature of a language-independent design structure poses inherent pedagogical problems that are incongruent with the basic aims of language education. The examples highlighted above characterize these pedagogical problems. The structure and design process, therefore, must be considered to be a function of socio-cultural forces and ought to be evaluated as such.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

Hubbard (1988) proposes a method of evaluation that is more 'holistic', implying more teacher-friendly. Hubbard makes support the idea of a 'framework', for evaluation, which is easy to incorporate into existing evaluation methodologies.

> A framework in this context means an integrated description of the components of something – in this case CALL materials – with respect to a particular goal – in this case evaluation. Rather than asking a specific set of questions, a framework provides the tool through which an evaluator can create his or her own questions or develop some other evaluation scheme.

It is exactly this type of methodology that should be applied in including issues of cultural representation into the CALL design process. A new evaluation scheme that merely focuses on one element would be as limited as many previous evaluation methods that offer up a checklist of good and bad elements. There have been numerous articles and books showing the drawbacks of such evaluation approaches since the mid-to late 1980s (Preece & Jones, 1985; Chang & Osguthorpe, 1987; Davidove, 1987; and Dudley-Marling & Owston, 1987)

The framework that I am proposing for CALL evaluation specifically reflects the current reality of the commercial CALL design process. CALL software, particularly in a commercial environment, should be viewed in regards to its process of creation. If mis-representations or culturally inaccurate material were included within the program, then the next step would be to analyze the design components to see if software cloning or content/delivery isolation have been employed. While this process seems relatively simple, it is reflective of an intrinsic evaluation methodology. The purpose of this is to raise greater questions about the strength of CALL as an educational discipline. This is based on a qualitative analysis model for educational software evaluation laid out by Werner (1980). Werner states that evaluation must have, "…an explicit methodological framework and purpose; public dialogue about the appropriateness of the framework …; and evidence that it has contributed to a better understanding of the curriculum" (Eraut 1989, 448).

5 Conclusion

I see software evaluation as a way for CALL to create a *public dialogue* about the purpose of our curriculum. The nature of CALL is international and reflects many different cultures. Why then, do so many software programs create a worldview that is singular in nature? Why also, is this design process that is technologically functional, not considered to be detrimental to the aims and goals of CALL? I argue that this framework for CALL software evaluation is fundamental, if commercial products want to have any educational validity at all. The past twenty years show a general focus on the technical/functional nature of educational software. Less attention is paid to issues of content or visual representation. It is generally assumed that the content will be mostly accurate in educational software. Particularly in CALL, where language content experts are more likely to be employed in commercial software projects, it would appear that this logic also holds true. The nature of commercial CALL software design, however, is regularly producing software that incorporates many cultural inaccuracies and mis-representations.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

CALL needs its own formative and summative evaluation methodologies that address the uniqueness of our discipline. Since the very nature of CALL involves a multi-cultural world, each evaluation, be it formative or summative, must address issues of cultural authenticity and cultural representation. The issues raised in this article need to become an integral part of any CALL evaluation framework. Finally, commercial CALL developers need to get away from true cloning practices that produce 'globalized' software and work towards an approach that incorporates issues of accurate cultural representation at all levels of the design process.

References

Abramson, G. W. (1998) How to Evaluate Educational Software. *Principal* **78**(1): 60–61.

Auralog: *Innovation in Foreign Language learning*. www.auralog.com. 10 Oct. 2002.

Baig, E. (1995) Parlez vous CD-ROM? *Business Week*, **29**: 99.

Balajthy, E. (1996) Trends in Literacy Software Publication and Marketing: Multicultural Themes. Paper presented at the *National Reading Research Center Conference on Literacy and Technology for the 21st Century* (Atlanta, GA, October 4–5).

Boling, E. and Soo, K. (1999) Call Issues: Designing CALL Software. In: Egbert, J. and Hanson-Smith, E. (eds.), CALL *Environments: Research, Practice, and Critical Issues*. Alexandria, Va.: Teachers of English to Speakers of Other Languages.

Brett, P.A. and Nash, M. (1999) Multimedia Language Learning Courseware: A Design Solution to the Production of a Series of CD-ROMs. *Computers in Education* **32**: 19–33.

Bright, G. (1983) Explaining the Efficiency of Computer Assisted Instruction. *AEDS Journal* **16**(3): 144–152.

Cameron, K. (1998) Culture and the Language Curriculum: An Important Issue? In: Calvi, L. and Geerts, W. (eds.), *CALL, Culture, and the Language Curriculum*. London: Springer.

The CALICO Review. 15 Oct. 2002, Burston J. and Abraham L (eds.) http://calico.org/CALICO_Review/index.htm

Carey, J. M. (1998) Creating Global Software: a Conspectus and Review. *Interacting with Computers* **9**: 449–465.

Chang, L. and Osguthorpe, R.T. (1987) An evaluation system of educational software: A self-instructional approach. *Educational Technology* **28**(6): 15–19.

Davidove, E. A. (1987) Evaluation and Selection of Courseware Development Software. *Educational Technology* **27**(7): 34–7.

Deutsch Na Klar! (1999) CD-ROM. McGraw-Hill Higher Education.

Disney Cruise Line – Fragen und Antworten http://www.disney.de/usa-resorts/waltdisneyworld/IV/G/ 15 Oct. 2002.

Dudley-Marling, C. and Owston, R.D. (1987) The state of educational software: a criterion-based evaluation. *Educational Technology* **27**(3): 25–29.

England, E. (1984). Design and Evaluation Issues on CALL Materials. *CALICO Journal* **2**(1): 11–13.

EPCOT-Center Florida http://195.185.214.164/iv/ausland/angebot/epcot.html 15 Oct. 2002

Eraut, M. R. (Ed.) (1996) *International Encyclopedia of Educational Technology*. Cambridge: Cambridge University Press.

Fidelman, C. (1990) But Will It Work? Formative Evaluation in Foreign Language Materials Development. *Journal of Educational Techniques and Technologies* **22**(4): 6–12.

Fidelman, C. (1993) Formative Evaluation for an Enabling Technology: The Case of "In the French Body". In: Borchardt, F. L. and Johnson, E. (eds.), *Proceedings of the 1993 CALICO Annual Symposium: Assessment*. Durham, NC: CALICO. **58.**

Garret, N. (1998) Rubrics for teaching Culture. Implications for CALL. In: Calvi, L. and Geerts,

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

W. (eds.) *CALL, Culture, and the Language Curriculum*. London: Springer.

Germanware http://williow.cats.ohiou.edu/~lrc/germanware/ 24 December 2001.

Gimeno-Sanz, A., Davies, G. and Bangs, P. CALL Software Design and Implementation. http://www.ict4lt.org/en/en_mod3-2.htm 15 Oct. 2002.

Hannafin, M. and Peck, K. (1988) *The Design, Development, and Evaluation of Instructional Software*. New York: Macmillan Publishing.

Hom, H. (2001) Selecting and Integrating CALL Software Programs into the EFL Classroom. Paper presented at the *Information Technology and Multimedia in English Language Teaching Conference* (Kowloon, Hong Kong, June 1–2).

Hubbard, P.L. (1988) An Integrated Framework for CALL Courseware Evaluation. *CALICO Journal* (December): 51–72.

Ilgen-Lieth, M. and Hazen, M. (1987) Procedure for Formative Evaluation of Instructional Software by Instructional Design Consultants – Part 1 – Overview. *Performance and Instruction*. **26**(7): 16–9.

Ilgen-Lieth, M. and Hazen, M. (1987) Procedure for Formative Evaluation of Instructional Software by Instructional Design Consultants – Part 2 – The Preliminary Evaluation. *Performance and Instruction*. **26**(8): 30–4.

Ilgen-Lieth, M. and Hazen, M. (1987) Procedure for Formative Evaluation of Instructional Software by Instructional Design Consultants – Part 3 – Group Evaluation, Feedback, and Conclusion. *Performance and Instruction*. **26**(9): 46–8.

Jobs und Praktika im Ausland http://www.arbeitsamt.de/zav/services/jobs/usa.html 30 Apr. 2002

Karkaletsis, E. A., Spyropoulos, C. D. and Vouros, G. A. (1998) A knowledge-based methodology for supporting multilingual and user-tailored interfaces. *Interacting with Computers* (**9**): 311–333.

Klemes, J., Eshet, Y. and Henderson, L. (1998) Educational Multimedia Implementation in Schools: Producer-Teacher-Student Links. In: ED-MEDIA/ED-TELECOM 98 *World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications*. Proceedings, Freiburg, Germany http://www.aace.org/conf/edmedia/98/98advprog.htm

Krawchuk, C. (1998) Meeting Software Needs of Public and Higher Education: A Collaborative Model. In: *Proceedings of Selected Research and Development Presentations at the National Convention of the Association for Educational Communications and Technology* (AECT) Sponsored by the Research and Theory Division (20th, St. Louis, MO, February 18–22).

Lawson, T. (1974) *Formative Instructional Product Evaluation*. Englewood, New Jersey: Educational Technology Publications.

Laurillard, D. (1991) Principles for Computer-Based Software Design for Language Learning. *Computer Assisted Language Learning* **4**(3):141–52.

Let's Talk German. (1996) CD-ROM. Syracuse Language Systems.

LearnPlus.com. http://www.learnplus.com/ 15 Oct. 2002.

Levy, M. (1999) Design Processes in CALL: Integrating Theory, Research and Evaluation. In: Cameron, K. (ed.) *CALL: Media, Design, and Applications*. Lisse: Swets and Zeitlinger.

LINC CD-ROMs – Language Interactive Culturehttp://www.uia.ac.be/linc/ 15 Oct. 2002.

McCormick, T. E. and McCoy, S. B. (1990) Computer Assisted Instruction and Multicultural Nonsexist Education. *In Computers in the Schools* **7**(4): 105–124.

McDougall, A. and Squires, D. (1995) A Critical Examination of the Checklist Approach in Software Selection. *Journal of Educational Computing Research* 1**2**(3): 263–74.

McNergney, R. F. and Reed, D. (2000) Evaluating Technology-Based Curriculum Materials. *ERIC Digest*. ERIC Clearinghouse on Teaching and Teacher Education, American Association of Colleges for Teacher Education.

Merrill, P., Hammons, K., Vincent, B., Reynolds, P., Christensen, L. and Tolman, M. (1996)

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press

Computers in Education. Boston: Allyn and Bacon, 109–119.

Myles, S. (1998) The language learner and the software designer: A marriage of true minds or ne'er the twain shall meet? *ReCALL* **10**(1): 38–45.

Pederson, K. M. (1987). Research on CALL. In: Smith, W. F. (ed.), *Modern media in foreign language education: Theory and implementation*. Lincolnwood, IL: National Textbook Company, 99–131.

Polonoli, K. E. (2000) What Makes Educational Software Educational? *Virginia Society of Technology in Education Journal* **15**(1): 6–31.

Preece, J. and Jones, A. (1985) Training Teachers to Select Educational Computer Software: Results of a Formative Evaluation of an Open University Pack. *British Journal of Educational Technology* **16**(1).

Price, R. (1991) *Computer-aided instruction: A Guide for Authors*. Pacific Grove, Calif.: Brooks/Cole.

Russell, J. D. and Blake, B. L. (1988) Formative and Summative Evaluation of Instructional Products and Learners. *Educational Technology* **28**(9): 22–29.

Sadker, M.P., and Sadker, D.M. (1986) Sexism in the classroom. From grade school to graduate school. *Phi Delta Kappan*, **67**(7): 512–515.

Selected Bibliography for CALL Evaluation

http://nflrc.hawaii.edu/aboutus/ithompson/flmedia/evaluation/callbib.htm 24 Dec. 2001.

Shaughnessy, M. (2002) Review of LearnPlus.com. *CALICO*. http://calico.org/CALICO_Review/review/learnplus00.htm

Shen, W. and Mo, W. (1995) *Reexamining the Issue of Authenticity in Picture Books*. ERIC Viewpoints.

Smith, M. and Brandenburg, D. (1991) Summative Evaluation. *Performance Improvement Quarterly* **4**(2): 35–58

Software Review Guidelines http://astro.ocis.temple.edu/~jburston/CALICO/softrev00.htm 15 October 2002.

Speak German. (1996) CD-ROM. Expert Software.

Think and Talk German. (1996) CD-ROM. Berlitz.

Tell me More French. (2000) CD-ROM. Auralog.

Tell me More German. (2000) CD-ROM. Auralog.

Treffpunkt Deutsch. (1999) CD-ROM. Prentice-Hall.

Waltje, J. Review of Tell me More German. *CALICO*. http://astro.ocis.temple.edu/~jburston/CALICO/review/tmm-germ00.htm

Werner, W. (1980) Editorial criticism in curricular analysis. *Curri. Inq*. (**10**): 143–54.

https://doi.org/10.1017/S0958344003000922 Published online by Cambridge University Press