# Intelligent CALL: The magnitude of the task

Cornelia Tschichold

# Intelligent CALL: The magnitude of the task

Cornelia Tschichold

University of Wales Swansea – CALS
c.tschichold@swansea.ac.uk

## Résumé

La qualité de la plupart des programmes ALAO n'est pas bien équilibrée en ce qui concerne l'utilisation de la technologie informatique et le contenu linguistique. Ce déséquilibre peut être expliqué par les contraintes très divergentes agissant sur le développement des didacticiels. Les produits ALAO commerciaux souffrent surtout d'un manque d'adaptation à l'apprenant et du manque de réponse intelligente à la production linguistique de l'apprenant. Les approches TAL n'ont pas encore atteint une qualité suffisante à cause de la distance énorme entre le langage des apprenants et les genres linguistique pour lesquels les outils TAL ont été développés. La solution proposée est une approche locale, centrée sur le lexique (bilingue), approche qui combine les ressources existantes pour arriver à des didacticiels plus créatifs et interactifs.

**Mots-clés** : méthodologie, évaluation et réponse, lexique, ALAO, TAL.

## Abstract

The quality of most CALL programs is not well balanced with respect to the use of computer technology and of language content and processing. This imbalance can be explained by a number of constraints pulling CALL developers in diverging directions. For commercial CALLware the poor learner fit and lack of feedback is a serious impediment. So far ICALL approaches trying to overcome this have not been of a sufficiently high quality due to the vast distance between most learner language and the text genres NLP is helpful for. The way forward suggested here is for ICALL to take a localized, (bilingual) lexicon-centred approach that combines sophisticated resources with improved learner fit for more creative and interactive CALLware.

**Keywords**: methodology, feedback, lexicon, ICALL, CALL, NLP.

# 1. Introduction

The two fields of computer-assisted language learning (CALL) and the applied branch of computational linguistics, natural language processing (NLP) would seem to be natural allies when considering ways to improve the way computers are used for learning foreign languages, and one might therefore wonder why the development of Intelligent CALL (ICALL) programs is not more advanced at the present moment. Multimedia computers certainly seem to be capable of representing reality in all its possible (and also in many impossible) forms, so the hardware cannot be the problem. There are numerous well-functioning applications of NLP, from spell-checkers and electronic dictionaries to sophisticated machine translation programs. And we also have a huge and constantly growing body of research on second language acquisition (SLA) that should allow us to make the right decisions in the framework of CALL development. So why does it appear to be so difficult to combine all of this knowledge with the necessary hardware to come up with a really intelligent CALL program?

A first non-trivial problem is of course the fact that ultimately it is not so simple to combine several very different types of knowledge needed for such a goal, knowledge which is

typically held by different groups of people, so that it becomes very challenging to achieve a fruitful cooperation in a converging work culture. Computer hardware and software specialists, computational linguists, SLA researchers and foreign language teachers tend to work in very different paradigms, with widely diverging background assumptions and typical constraints on their work. Bringing people from such different traditions together and balancing their contributions to the common goal is the first step that is necessary for an ICALL project on a reasonable scale to be successful.

If we take a look at existing programs, we can see that this is indeed not such an easy step to make. CALL programs are clear evidence that the emphasis during the development stage of the software was either on delivering good linguistic content or rather on good, up-to-date use of technology. Many of the commercial programs make excellent use of the technology available and look very attractive at first sight, but are disappointing where it comes to the language content and the pedagogical principles seen at play there. These shortcomings can often be explained quite easily by the commercial pressures at work in such projects. Not only do the developers have to work to tight time schedules, they also need to come up – at frequent intervals – with a new product that appeals to a wide market such as the worldwide EFL market.

On the other hand, we find programs where much more time and thought has been spent on the language content and how much and what kind of language exposure learners should have and how the learners' linguistic production is to be processed. These programs tend to be developed either by enthusiastic teachers with extensive teaching experience and a sound knowledge of their particular student group, or by academics with a background in computational linguistics.

The programs developed by teachers are typically small and well adapted to their own student population, but often use technology less than ideally when compared to commercial programs and are limited in their software functions. And then there are programs developed by groups of academically based linguists trying to use innovative approaches to process language and generate feedback, but this group is also less concerned about the use of up-to-date multimedia features.

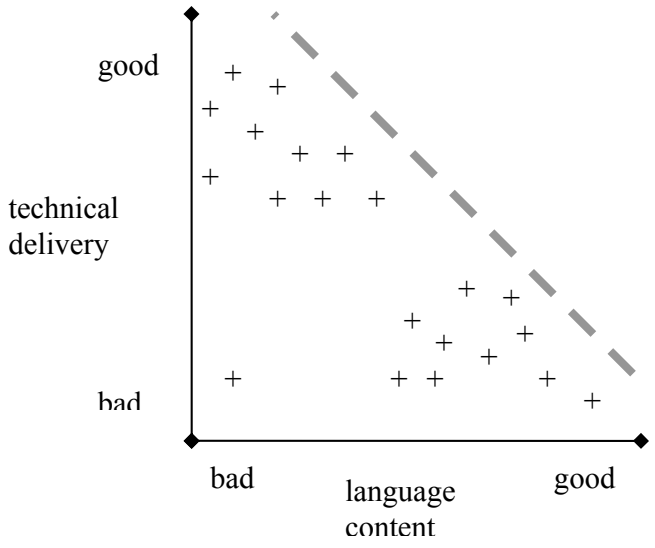



*Figure 1. Delivery mode vs. content quality in CALL*

The common element in all these types is that they are typically less than perfect in (at least) one of the two aspects of technical and linguistic components. If we plot the quality of a random number of them (each '+' standing for one hypothetical CALL program) on a graph, with technology and delivery on one axis, and language content and feedback on the other, we might get a picture as seen in figure 1. There seems to be a certain bar (- - - -), with an area above it that is difficult to reach for CALL programs. For a general evaluation, where all of these aspects have to be taken into account, technical delivery should meet a certain minimal threshold (albeit one that is rising all the time). The language content is less easy to evaluate, but ultimately it is the learning success that can be achieved with the help of the program which should determine the overall mark. This aspect can usefully be broken down into facets such as consistency, learner fit, etc.

The best existing programs are thus found near the bar; unfortunately, we also find a number of programs quite far away from that line. But no program to date can deliver the dream of a customized, all-purpose language course that makes learning a foreign language an entertaining and expeditious experience, gives us practice in small talk, corrects our written texts, clearly explains the errors we have made, and produces a couple of remedial exercises before proceeding to the next tailored lesson. Such expectations are well beyond today's technology. The reasons for this are to be found in a combination of forces or constraints pulling in different directions.

## 2. Three factors at play (at least)

These diverging constraints on the development of CALLware tend to lead to dissimilar end products by the different groups involved in authoring CALL. While commercial developers tend to yield to the pressures of time, money and the use of the latest computer technology, teachers will most likely focus on adapting the language content, and computational linguists will primarily be interested in improving feedback mechanisms by processing (more or less) free text. A number of other factors can easily be imagined to play a role in the development process, as illustrated in figure 2, but these three diverging constraints will be the focus here.

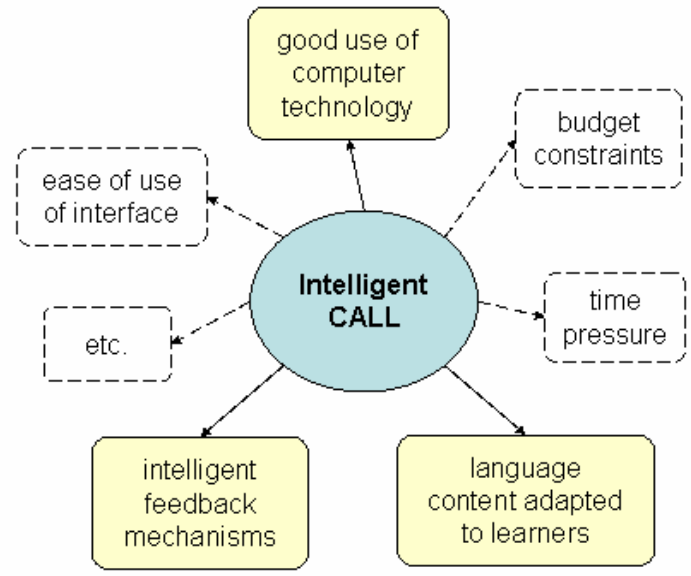



*Figure 2. Opposing constraints in CALL development*

## 2.1. Commercial CALL

While much of commercial CALLware uses the latest computer hardware and software for the delivery of the content in useful ways, the language content itself is frequently less than ideal. The attractive interface sometimes seems to be more important than the less easily visible language content (Hewer *et al.*, 1999). Commercially developed programs, but not only those, often seem to show a certain lack of awareness of (not to say disregard for) the complexity of the task. Clifford (1998) lists some of these in his seven misperceptions about CALL, and as his list has lost none of its relevance today, I quote it here.

1. language is a simple phenomenon;
2. language acquisition is a simple task;
3. presentation of information is all that is required for language learning;
4. access to information eliminates the need for presentations;
5. every learning activity is appropriate for all learners;
6. automating poor teaching practices will improve the instructional process;
7. poorly designed 'shovelware' is better than well designed, but costly, learning activities. (Clifford, 1998, p.1)

In such a (common, but unconscious) view, language is seen as a finite body of knowledge (1), which simply needs to be presented (2, 3) to the learner and then practised via a restricted set of exercises (2, 5). CALL programs developed with these misconceptions in their developers' minds typically use rudimentary presentation followed by multiple choice questions and other simple to mark tasks. The language meant to be taught is rarely presented in a way that would be conducive to learning it; and the possibilities of practising it are hampered by the lack of meaningful feedback beyond the "wrong – try again"-type.

Reasonably critical reviews of CALL programs such as those on the CALICO website often mention the lack of sound pedagogical principles behind much new CALLware. But descriptions of programs frequently focus on functionalities and other technical aspects[1] and say comparatively little on the language content and the didactic aspects of the program in question. In view of the aim of CALL, *i.e.* language learning, the largest proportion of text of a CALL review should be devoted to the content of the program. The recent review by Burston (2005) of a vocabulary training program is a case in point. While the words themselves (given as examples in the review) belong to a beginner's vocabulary, the example sentences shown are only suitable for advanced learners. Burston rightly questions some of the methodology and the feedback mechanism offered by the program, but the overall mark seems to evaluate the program more against other similar CALL programs than against other, non-computer-based methods for learning vocabulary.

Commercial CALL program developers are of course not completely oblivious to criticism by pedagogists and language teachers, and program descriptions often pay lip service to the developments in language teaching methodology in stressing the communicative aspects of the exercises[2]. In many cases, however, this has led to a move away from the earlier structure drills towards the delivery and display of growing amounts of linguistic material. Unfortunately, this usually also means fewer attempts at trying to deal with learners' language

[1] It has to be mentioned here that technical aspects such as compatibility are among the first criteria that need to be taken into account when considering whether to buy some piece of software. But in an ideal world pedagogical aspects would override such purely technical considerations.

[2] One program criticized quite sharply in Nesselhauf and Tschichold (2002) has since appeared in a much improved version.

production, leading to an increasing lack of any type of interactivity that goes beyond mouse clicks. This is a safe strategy as it minimizes the danger of giving wrong or otherwise unsuitable feedback to the user, a danger difficult to underestimate from a didactic point of view (Schulze, 2003; Tschichold, 1999). Such CALL programs can be excellent for the delivery of practice material, but they avoid dealing with the productive skill of learners, an aspect which is increasingly being recognized as crucial for the language learning process. Restricting teaching to the presentation of linguistic material leads to a learnability problem, as simple exposure to such material is not sufficient to acquire many components of a language. CALL programs which do not invite the learner to produce utterances in the foreign language can thus not claim to be complete learning packages.

### 2.2. CALLware designed by practitioners

A number of enthusiastic language teachers have designed and implemented their own small CALL programs. As these are primarily meant to be used by the author's own learners, they are particularly well suited to the needs of these students. The downside is that the limited time and programming experience available to this group of CALL developers often results in the technological aspects of such programs being somewhat simple and not necessarily up-to-date with the latest hardware developments, a consideration that could take on increasing importance as more and more language learners will have grown up with technologically highly sophisticated computer games. The number of software functions in these teacher-developed CALL programs is also necessarily restricted. However, these shortcomings are often balanced by the considerable advantage of the fine-tuning and frequent up-dating such programs make possible, at least as long as the author is willing to continue working on the program.

What such programs lack in technological sophistication and innovation[3], they often make up in learner fit. Given that the authors know their students well, they can rely on their experience and adapt the feedback to the expected answers provided by their students. By doing this, they can at least partially compensate for one of the major difficulties when determining feedback on learner input, namely that fact that the language taught to language learners, especially to beginners and those taught in a communicative framework, is highly ambiguous simply because this is the most useful type of language, adaptable to all sorts of contexts. This property of human language makes it difficult for computer-assisted instruction, however, to predict all possible (correct and wrong) answers to any but the most simple types of questions.

### 2.3. NLP in CALL

The impossibility of providing adequate feedback to any type of question that goes beyond the complexity of multiple choice questions in traditional CALL has led to numerous attempts (see the projects described in *e.g.* Holland *et al.*, 1995; Jager *et al.*, 1998; Gamper and Knapp, 2002; Heift, 2003 and others in that special issue of the CALICO journal; and Dodigovic, 2005) by linguists to use techniques developed in computational linguistics in order to analyse the language produced by learners and generate feedback that would allow the learner to identify and correct the error found by the program. This group of CALL developers is not

[3] Teachers who want to write their own CALL materials do not need to learn a programming language any more; they can use so-called authorware that allows them to input their own customized language content with the help of templates, while providing the interface for a number of simple exercise types. Frequent updates of authorware can help to overcome the lack of technological sophistication.

primarily concerned with the use of the latest technology in order to come up with a good-looking interface either, so their programs will not be as eye-catching as the commercial programs. Despite the didactic and academic interest they offer, none of these NLP-based projects have made it into the commercial market so far, partly because of their small scale, but partly also because their most interesting aspect, error detection and customized feedback, does not work very reliably outside the context for which it has been developed. Coupled with the fact that only a minority of errors produced by language learners are detected at all, this is a disadvantage that is difficult to ignore from a didactic point of view.

Many of the NLP components used in ICALL projects were originally developed for processing correct native speaker language, and not for the purpose of error detection in learner language. On the level of morphology, the obstacles for adapting an NLP component (an electronic morphological lexicon in this case) to error-prone learner language are not insurmountable, but on the level of syntax it can become very difficult to balance the need for relaxing the grammar constraints in order to be able to parse a sentence at all, and at the same time limiting the proliferation of analyses caused by such a method. The alternative method of using 'error grammars' that look for specific types of errors is safer, but will obviously only lead to the detection of a limited number of errors. It will always remain impossible to imagine and plan for all possible errors language learners could make. In order for the latter approach to be effective at all, the program has to be specific to a particular learner group, with users sharing the same or very similar native language(s). A number of recent systems, such as those described in Heift (2003), Reuer (2003), Schulze (2003), and Vandeventer (2001), take this approach. Once an error has been detected by such a system, the developers are then faced with the task of translating the machine-like code produced by the NLP system into a text message that is fit for human consumption, *i.e.* for language learners who are neither computer programmers nor trained linguists in most cases. Given the vast distance between the way most language learners think about elements of a sentence and the methods used and output produced in NLP, this is far from trivial.

While morphology and syntax doubtlessly account for substantial numbers of errors found in learner language, problems on the levels of semantics and the lexicon also represent a significant proportion of all learner errors (Cutting, 2000), especially for the most frequently learnt foreign language, English, with its relatively poor morphology and vast vocabulary. There is no viable NLP system for dealing with this aspect of language as yet. Existing (commercial) systems are only capable of treating a small part of any language or a subregister where the amount of vagueness and ambiguity so typical for general language has been drastically reduced.

The problem for CALL then is twofold: on the one hand the foreign language lexicon is one of the biggest obstacles for language learners at all levels of proficiency (Nation, 2001), leading to significant numbers of errors and other divergences from native speaker language use, and – on the other hand – the lack of viable NLP components to deal with the semantic level of language.

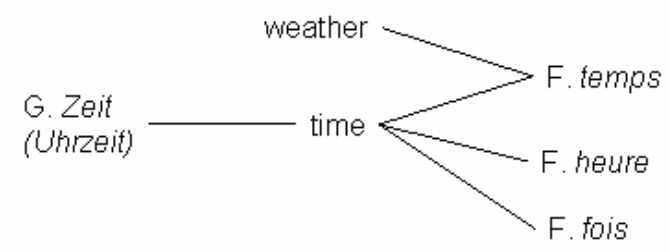
We find a corresponding twofold dilemma in computational linguistics: Along with the lexical bottleneck, *i.e.* the problem of scaling up small NLP systems to a sufficiently large lexicon, the semantic ambiguity in everyday language is one of the most persistent problems for computational linguistics. Those NLP components where this problem has been tackled to such a degree that viable products emerge are specific to highly specialized text genres, *e.g.* construction manuals rich in specialized terminology. In such text genres, the ambiguity inherent in most words can be reduced to such a degree that the NLP system can cope with the remaining problems to a satisfactory degree. This type of language, however, could not be

further away from the kind of language taught to and used by language learners. As the number of words language learners can be expected to learn within a few years is very limited compared to the total number of words in any natural language, the words that are chosen for beginners are necessarily those that will be most useful to them, in other words, those vocabulary items that are quite vague and ambiguous. In addition to this deliberately imprecise vocabulary, language learners are also often taught (and/or develop) strategies to paraphrase, *i.e.* to use even less precise words to describe the intended meaning. Briefly, the language produced by learners is about the worst imaginable type of language for NLP.

# 3. The dilemma for ICALL

Given that on the one hand, learners are usually taught highly ambiguous language because this is the most useful type of language for communicative purposes, and that on the other hand, NLP components can only deal reasonably efficiently with language genres where the natural ambiguity of everyday language has been massively reduced, we can see the fundamental dilemma for ICALL, a dilemma that remains even if other problems such as budget and time constraints and the quality of the interface and of the language content in CALLware are solved or disregarded. One possible conclusion CALL developers could draw from this situation is to concentrate on potentially more feasible and very specific subtasks rather than try and teach all aspects of the foreign language to all types of learners.

As computers are probably less than ideal to teach the truly communicative aspects of language, vocabulary learning could be seen as a subtask suitable for CALL. Learning hundreds and thousands of new words involves at least a certain element of drill and repetition for most learners, certainly those outside an immersion situation (Nation, 2001). While the vocabulary training programs on the market today have not necessarily been developed out of such motives, they can still serve as a suitable illustration of the argument. Most of them are specific to a single language pair, so one advantage in terms of NLP use is that the user group and the error types that can be expected are clearly more specific than they would be for CALL programs aiming to be L1-neutral. While there are vast differences in the didactic approach and quality of implementation of such vocabulary programs, most of them share one problematic aspect, the reduction of vocabulary to bilingual pairs of single words (Nesselhauf and Tschichold, 2002). This unfortunately supports the assumption by many beginner language learners that every text word in their native language has a one-to-one match in the foreign language, an assumption that has been called the naïve lexical hypothesis (Kesner Bland *et al.*, 1990). While such an assumption sometimes works reasonably well for some words from closely related languages (English *time* can very often be translated into German as *Zeit*), in most cases it is a dangerously simplistic view. Disregarding the most naïve level of this hypothesis, *i.e.* the assumption of complete isomorphism down to the level of morphology, we readily find examples of diverging polysemy to trap the learner. English *time*, for instance, has at least three common translations into French, with one of them, *temps* also being the translational equivalent of English *weather*.



Such complicated relations are the norm in the basic vocabulary of any language pair if we take into account the numerous combinations words enter into as soon as they leave the vocabulary list and are used in real utterances. Beginners' vocabulary lists are necessarily full of words that are quite highly polysemous, but unfortunately for both the language learner and the CALL developer, the polysemy of individual words in one language typically is not concurrent with the polysemy of one of the translational equivalents in the other language[4].

Linguists working on learner lexicography both inside and outside the field of CALL are beginning to address this challenge. Paper dictionaries like those in the Cambridge series "Word Routes" or the electronic learners' dictionary project Eldit (Abel and Weber, 2000) show a promising approach towards a possible solution of the dilemma described above. Computer-based lexicographic databases have a distinct advantage in this area as they are not bound to the linear presentation of data and allow for multiple links, offering almost limitless possibilities for presenting subsets of data to the user.

To conclude, I would like to argue in favour of a more localized approach to ICALL. The functional and communicative aspects of language are probably better left to human teachers for the time being, but this still leaves much room for CALLware. We should remember that lexical errors and lack of adequate vocabulary are the biggest hindrance to communication in a foreign language. It therefore makes sense for ICALL to concentrate on this aspect. But vocabulary needs to be taught in much more varied, complex, creative and interactive ways. Words need to be presented in several different contexts, gradually increasing in difficulty, so that learners can avoid the trap of the naïve lexical hypothesis. Words also need to be practised and revised through a range of exercises. These should be designed in such a way that the program can give intelligent feedback to the user. The aim should be for CALL activities to focus on the progression from controlled learning to automatic processing of linguistic forms, a step that is generally assumed to be achieved through practice and routinization. This fact would favour a role for CALL that is centred around vocabulary learning and a lexically centred approach to language teaching. We have the technology (large-scale lexicons and morphological analysers), the linguistic knowledge (research findings on vocabulary acquisition), and the language data (various corpora) necessary to design language learning tasks that put these principles into practice.

# References

ABEL A., WEBER V. (2000). "ELDIT – A Prototype of an Innovative Dictionary". In U. Heid *et al.* (eds), *EURALEX Proceedings* II. Stuttgart: 807-818. http://www.eurac.edu/eldit.

BURSTON J. (2005). *Review of WordChamp*. http://calico.org/CALICO_Review/review/wordchamp00.htm .

CLIFFORD R.(1998). "Mirror, Mirror, on the Wall: Reflections on Computer-Assisted Language Learning". In *CALICO Journal* 16 (1): 1-10.

DODIGOVIC M. (2005). *Artificial Intelligence in Second Language Learning*. Multilingual Matters, Clevedon.

GAMPER J., KNAPP J. (2002). "A review of intelligent CALL systems". In *Computer-Assisted Language Learning* 15 (4): 329-342.

[4] The popular literature on so-called (true and) false friends illustrates the phenomenon for the relatively small group of words that also have a formal resemblance, but the phenomenon is common across the whole bilingual lexicon.

HEIFT T. (2003). "Multiple learner errors and feedback: a challenge for ICALL systems". In *CALICO Journal* 20 (3): 533-548.

HEWER S., RENDALL H., WALKER R., DAVIES G. (1999). "Introduction to computer assisted language learning (CALL)". In *Module 1.4 of ICT4LT*. www.ict4lt.org.

HOLLAND V.M., KAPLAN J.D., SAMS M.R. (1995). *Intelligent Language Tutors: Theory Shaping Technology*. Lawrence Erlbaum, New Jersey.

JAGER S., NERBONNE J., VON ESSEN A. (1998). *Language Teaching and Language Technology*. Swets & Zeitlinger, Esse.

KESNER BLAND S., NOBLITT J., ARMINGTON S., GAY G. (1990). "The Naive Lexical Hypothesis: Evidence from Computer-Assisted Language Learning". In *The Modern Language Journal* 74 (4): 440-450.

NATION I.S.P. (2001). *Learning Vocabulary in Another Language*. Cambridge University Press, Cambridge.

NESSELHAUF N., TSCHICHOLD C. (2002). "Collocations in CALL: an investigation of vocabulary-building software for EFL". In *Computer Assisted Language Learning* 15 (3): 251-280.

REUER V. (2003). "Error recognition and feedback with lexical functional grammar". In *CALICO Journal* 20 (3): 497-512.

SCHULZE M. (2003). "Grammatical errors and feedback: some theoretical insights". In *CALICO Journal* 20 (3): 437-450.

TSCHICHOLD C. (1999). "Grammar checking for CALL: strategies for improving foreign language grammar checkers". In K. Cameron (ed.), *CALL: Media, Design & Applications*. Swets & Zeitlinger, Lisse: 203–222.

VANDEVENTER A. (2001). "Creating a grammar checker for CALL by constraint relaxation: a feasibility study". In *ReCALL* 13 (1): 110-120.