Evaluation of software components used in Multimedia Labs: A critical study

V.B.Sangeetha[2]
Assistant professor
Dept of Humanities and Mathematics, GNITS
Dr.P.Aparna[1] **,**
Professor & HOD,Dept of Humanities and Mathematics, GNITS

Article Received: 02/05/2022, Article Accepted: 03/06/2022, Published online: 04/06/2022
DOI:10.36993/RJOE.2022.7.2.46

Abstract

Technology has become an integral part of the language classroom. More and more language teachers are adopting different CALL materials in their classrooms. Typically teachers' would want to assess the attitudes and perceptions of students in a learning environment that involves language learning. Therefore, language teachers are interested in the viability and effectiveness of various methodologies and strategies of CALL, often to refine and improvise. CALL materials include language learning software, websites, online courses, CMC tools, etc.; there is a wide range of methodologies starting from a simple checklist to survey to complex longitudinal studies that may involve qualitative and quantitative approaches, to evaluate these materials. Well known frameworks of Hubbard (1987, 1988,1992,1996) and Chapelle (2001) offer a sophisticated foundation for a principled approach to evaluation in CALL. This paper tries to understand and discuss these two frameworks in detail. Further, an attempt is also made to identify the lacunae existing in these frameworks.

Keywords: CALL (Computer aided Language Learning), Evaluation, framework, Effectiveness.

To significantly enhance receptive language skills (listening& reading), multimedia language labs have been part of the curriculum for I B. Tech students at Jawaharlal Nehru Technological University since 2006. Colleges affiliated with the university were given the flexibility to choose from the different software components available. Most of the materials used in these labs are developed by the software vendors with little or no understanding of the pedagogy of material development. No attempt has been made to examine and analyze the software used by the colleges critically. Most of the teachers' have been using simple checklists to get feedback about these materials. Hence, the need to assess the software using a theoretical framework of Evaluation.

First, define CALL software, computer programs, *and accompanying* content Hubbard (2006) has outlined three different approaches: checklists, methodological frameworks, and SLA research-based approaches.

Why Evaluation? :

1) **Selection for a Course**: This is one of the most common reasons for Evaluation, where teachers evaluate some materials for using them in their classroom. Here a teacher can look into aspects like the infrastructure available, the student characteristics, specific objectives of the course, etc.

1) Checklists: Checklists have been in use since the earliest stages of CALL and remain widespread.

1) ***Methodological Frameworks****:* Methodological frameworks are compatible with some checklists but differ in two significant ways. First, methodological frameworks attempt to be primarily descriptive rather than judgmental in their form. Second, they attempt to fundamentally link with the language teaching and learning considerations outside of technology. Hubbard (1988) adapted the approach, design, and procedure constructs into categories describing the critical elements of Evaluation and renamed them teacher fit, learner fit, and operational description. The resulting framework became the evaluation module in a proposed comprehensive methodological framework that included modules for courseware development and implementation (Hubbard, 1996).

3)*SLA-based Approaches*: Given that teaching languages with software *is* a form of language teaching, another reasonable procedure for developing software evaluation rubrics is to build on recommendations from theory or research in instructed SLA. Ultimately, we might expect specific SLA results from learning with software research, but there has not been a sufficiently established base for such results. Consequently, this approach takes findings from non-CALL domains and interprets them in the CALL context.

An early attempt in this direction was Underwood (1984), who presented a case for a communicative approach to CALL based on generalizations from research and communicative theory of that period. His 13 points characterizing communicative CALL became a de facto evaluation rubric. Egbert and Hanson-Smith(1999) structured the characters in an edited volume on CALL to around eight generalizations for optimal language learning environments, again providing content for a research-based evaluation scheme. However, their work was not explicitly aimed at Evaluation.

The most ambitious project in this area to date is represented by the work of Carol Chapelle in the field she has dubbed CASLA—*computer applications in second language acquisition*—which includes not only CALL but also computer-based language testing and computer-based SLA research. Although parts of the model were developed in earlier articles, the work comes together in Chapelle's2001 book which is significant for (a) its characterization of Evaluation based on principles and (b) its specific SLA-based criteria. Concerning the first point, Chapelle offers a set of five principles for evaluating CALL, summarized as follows:

1. CALL evaluation is situation-specific;
2.CALL should be evaluated both judgmentally and empirically;
3.CALL evaluation criteria should come from instructed SLA theory and research
4.The criteria should be applied relative to the purpose of the CALL task
5.The central consideration should be language learning potential Hubbard (2004) argues that learner training should be a significant part of CALL software implementation and proposes five guiding principles for CALL learner training:
1.Experience CALL yourself from the learner's perspective;
2.Provide learners with some teacher training so that they can make better decisions when working independently;
3.Employ a cyclical approach, making training ongoing rather than relying on on one-time training sessions when the software is first introduced;
4.Use collaborative debriefings to encourage students to reflect on their learning process after using the software and to promote finding out about effective procedures from one another; and
5.Teach general exploitation strategies so that they can take greater control of the software and adapt it in ways beyond the designer's vision.

Model essential checklist: (about student characteristics)
1. Who are the target users?
2. What are the goals of your target group?
3. In what setting will the software be used: independent lab with no teacher available, lab associated with a class, a teacher-led class with one or a few computers?
4. How much do the teachers/lab assistants who will work with the students know about CALL?
5. Are Infrastructure facilities available in terms of hardware and technical assistance?
6. How much money do you have to spend?

Evaluating Student Outcomes:

A final area of the evaluation process that needs to be touched upon is determining the degree to which the software is used and whether how it is used has been thriving. This assessment process helps the teacher decide whether to use the software in the future and, if so, whether to use it in the same way or differently. It also adds to the teacher's general understanding of what students do with the software, which can influence future evaluations and implementation decisions. To this end, Chapelle (2001) provides a set of questions for determining the results of student use, empirically reflecting the six criteria presented previously for judgmental Evaluation.

1) **Observation*:*** The most direct way to get information on whether the software is having a positive effect on learning is by watching the students as they use it. In a lab situation, mainly when dealing with software new to the student, the teacher can walk around, note how students are navigating through the software, and interact with them as they use the software. Information gathered thus can be used both to evaluate the software and understand the ongoing learner training.
2) ***Tracking Systems:*** Perhaps the best way to get objective information on student use is to select software that includes tracking student actions or employ a screen capture device (Screen Cam) that will record the changes in the student display. Depending on the type of tracking system used and the nature of the data collected, this can allow for either an overview, for example, studentquiz scores or time on the computer (not necessarily the same as time on task) ora data set that is rich in detail but may be time-consuming to analyze. In this regard, a tracking system in software is seen as apositive feature.
3) ***Student Surveys:***

Another approach to gathering information on student perceptions of success or failure with the software is to ask them using a survey or questionnaire. While such information can be valuable, there are two concerns. First, if students know their responses are linked to some of the other assessments or believe (even erroneously) that this is the case, the results will be compromised. Thus, it is essential to ensure anonymity if possible. Second, even when students try to be completely honest, their reports may not correspond to their actions.

Fischer (2004) reported on a study of French reading software in which the student accounts of their use of program features were quite different from what was observed in the objective data in the tracking logs. If surveys are to be used, it is advisable to administer them either during or immediately after completion of a CALL activity to tap into fresh memories as much as possible.

4) Pre- and Post testing:

Evaluating student outcomes is a form of research, especially when done with software untried for a particular setting. Certain types of CALL instruction, particularly those which can be assessed with some degree of validity with discrete point tests such as vocabulary development, may be empirically evaluated using a pre- and post-test regime. While this may give helpful information on the outcome, it does not provide the data about the learning process that most of the other options do. It does, however, have strong validity with students and school teachers, especially when results are positive.

5) **Student Journals:** Kolaitis et al. (in press) report success in having students keep a "CALL journal" in which they include the time and description of the material worked

on and reflections on why they got specific answers wrong in exercises. Although this is mainly done for the students' benefit, this kind of journal also provides teachers with helpful information on how their students are progressing and using the software. However,like questionnaires, the data in student journals may not be entirely reliable and should be interpreted accordingly.

Conclusion

Software evaluation remains an important area of CALL, and there are indications that its role may be increasing, particularly in the domain of empirical Evaluation. However, most language teachers' software evaluation will remain judgmental, ideally with some empirical follow-up of the types described in the previous section. Even at the judgmental level, thorough software evaluation of the type often mandated by published checklists and procedures is time-consuming and might be impractical for many classroom teachers.

References:

Chapelle, C. A. (2001) Computer Applications in Second Language Acquisition:

Foundations for Teaching, Testing, and Research, Cambridge: Cambridge University Press.

Chapelle, C. and Douglas, D. (2006) Assessing Language Through Computer Technology, Cambridge: Cambridge University Press.

Hubbard, P. (1988). An integrated framework for CALL courseware evaluation. CALICO Journal, 6 (2), 51-72.

Hubbard, P. (1996). Elements of CALL methodology: Development, Evaluation, and implementation. In M. Pennington (Ed.), The power of CALL (pp. 15-33). Bolsover,TX: Athelstan

Sivakuraman, Thillainatarajan, et.al. 2012. Students' Perception of Multimedia Technology Integrated in Classroom Learning. International Journal of Humanities and Social Sciences, 2(11), 67-70

How to cite this article?
Gopika Nair U.I, " Evaluation of software components used in Multimedia Labs: A critical study",
Research Journal Of English(RJOE)7 (2), PP: 351-355,2022, DOI: http://doi.org/10.36993/RJOE.2022.7.2.46