e - learning, CAL evaluation, multimedia, databases, Distance Learning

Jan PIECHA[*]

CHARACTERISTICS AND EVALUATION PROCEDURE OF E-LEARNING UNITS

The Computer Aided Learning (CAL) systems development principles have to consider many complex task for to do. Very sophisticated rules must be integrated to create capable entities to achieve the expected pedagogical result. Several principles of CAL applications development have been presented in this paper, explaining main tasks that have to be solved by the developer in a process of e-learning units' evaluation.

1. THE E-LEARNING TERMINOLOGY AND CHARACTERISTICS

In a Programmable Teaching (PT) the lesson is driven in accordance with guidebook that allows choosing the user an own route through the courseware. The course program can be individualised both in content and in the course duration. Before we start the application development one of the education model has to be chosen that defines the lesson purpose and organisation.

The early results in programmable teaching were noticed for the first time when the first Computer Based Teaching systems (CBT) have been implemented. The pioneer works in the CBT systems usage defined several general classifications of these systems usage. The fundamental educational models of CBT were described in (Eberts, 1986) and in (Piecha,1989, 1991).

The algorithmic way of problems solving and a multiple choice of choosing the problem solution make the computer very effective in PT. What is more not many guidebooks are needed.

1.1. THE LESSON FUNCTIONS DEFINITION

The author of e-learning units will undoubtedly use the CAL model for educational resources development. The first task that has to be done concerns the function of the lesson selection (Burke, 1982), (Piecha, 1991). The application will be differently organised in case it a primary means by which the lesson will be presented or it is a supplementary form for the teaching process; with several key solutions, as:

[*] University of Silesia, Dept. of Computer Systems, Katowice, Poland, e-mail: piecha@us.edu .pl

1.1.1. ***Tutorial course***, where the computer simulates the actions of a very good teacher. The best design allows individualising the course for the user keeping precise records of the course progress. This model is recommended for a new material presentation, with an individual approach to the course.

1.1.2. ***Problems-solving course***, concerns the way of the computer usage as an intelligent computing tool. The computer is activated to present the problem to the student and then carefully, step-by-step, to spot the student's action. The system is commenting silly mistakes and gives a necessary help for the solution finding.

1.1.3. ***Gamelike course***, is one of the most complex tasks for to do by e - applications developer. This model of education is recommended for very boring experiments of problems solving. The motivation of the application user benefits greatly. Anyhow, the CAL system developer has to analyse a big complexity of the task. What is more not every subject is convenient enough to be presented in a gamelike mode. This function can be recommended for development of supplementary units to a main course application. For main subjects repetition (of the courseware) a less demanding application structure can be implemented. These programs will be used for drill-and-practice means of teaching processes.

1.2. PHYSICAL FEATURES OF THE LESSON

Physical characteristic features of the application describe a second group of these courseware solutions. They define the way of going through the lesson and how the program path of the course is selected (Piecha 1989); among them:

1.2.1. ***Linearity***, as a most common feature noticed in majority of e-learning packages. It determines a serial presentation of subjects group; according to the paper-pencil screenplay. It usually concerns applications developed by beginners in e-learning content development. Sometimes the computer characteristics limit the developer plans. Anyhow the linear compositions of the application give also several benefits:

- the application is very simple for development, revision and validation,
- the application can be very easy modified.

In the linear courseware every user goes through a unique path of the lesson. The linear structure of the application is not recommended for distance learning databases content development, because of very poor guiding algorithms implemented in the application.

1.2.2. ***Branching*** determines conditions for several paths of the lesson selection. This problem is strictly combined with the answers validation system that has been described bellow. This branching ability takes advantage of a great power of the computer in an individual course content selection, providing a personal learning experience for every student.

1.2.3. ***Multitrack*** (also called a “multi-level CAL lesson”) is developed in accordance with Kay's strategy, where several distinctly different levels of the course are provided. They allow individualising the lesson flow. The highest level of the track(content) is the shortest one. It uses fewer explanations, less examples and more complex questions. The lowest track is the longest one, with many comments and simple questions.

Many of the lesson levels can also be defined as laying in between controlled by an interactive conversations within the computer program.

1.3. ORGANISATION OF THE LESSON

In the application organisation, working under CAL scheme, several characteristic features can be distinguished (Piecha,1989):

1.3.1. ***Problem orientation*** is one of the most important aspects the CAL unit scheme. The user has to know, at any time, when to go ahead or back. When the problem is not clearly oriented the user will waste too much time, respect for the program, patience and motivation. When the lesson run depends on additional comments it is a sign that the course is not properly oriented.

The exact orientation of the application shall also be understood as an opportunity of making in easy way step back (to the screen appointed for the repetition).

1.3.2 ***Problem succession*** controls the fundamental techniques that should be implemented for fill-up of the computer screen. All successive (separate) layers of the application content have to be defined by the application user. Any "timing technique" for putting the successive information onto the screen is the most depressing way of the application progress and interactivity validation. Unfortunately many developers are using this technology with satisfaction. The automatic timing system causes, that:

- duration of the CAL lesson can not be individualised,
- the CAL system user can not control a route of the lesson,
- this “competition” is very depressing for the user.

1.3.3. ***Screen format*** defines the courseware composition, where also several principles can be distinguished:

- textual information has to be reduced to a minimum although necessary range,
- the screen display can not be overloaded, therefore difficult to read,
- the lesson has to be divided into carefully balanced parts with well-matched graphic forms, comments and animation.

1.3.4. ***Question system*** provides the application with higher quality measures, judging user responses. Well constructed question system:

- permits a student/pupil to interact with the subjects within the CAL unit, in a proper way,
- allows to measure a distance to mastery,
- is a key measure in setting the user his individual path through the lesson.

1.3.5. ***Branching system***, makes that the logical flow of the lesson can follow one of several alternative paths, depending upon which of the conditions is met (Piecha 1989). It gives the user a very powerful feature of the system with a very good use of the computer capability; in choosing the conditional routes through the application.

The information is displayed on the computer screen in portions of successive layers. The full set of application layers, displayed as a complete screen, has been called a frame. It is the smallest unit of the lesson, assigned with a branching index of the lesson unit.

1.4. THE APPLICATION FRAME TYPES

In the CAL systems model two main frame types can be distinguished: (*teaching frame* and *criterion frame).*

1.4.1. *The teaching frame*, represents part of the information presented to the student. It has the identification number to which the criterion frame is related. After several teaching frames some questions have to be put.

1.4.2. *The criterion frame*, contains questions or tasks. The answers in the criterion frames are validated. Although several questions, within the teaching frames are also given they usually are not validated. They work as an activation factor of the application.

The *criterion frame* can be build up according to several frame formats (protocols):

a) An *alternative responses* format (multiple choice), is the simplest question format where the user selects the correct answer from several items. This criterion form gives the user very simple validation algorithms but the questions number that has to be given, to establish the user knowledge stage, is very big. This structure of the application cannot be recommended for the CAL content solution.

b) A *response fill-up* format, where the application shows a part of the answer then a student is requested to fill up the remaining part of the frame. This criterion protocol can carry a very valuable data for the validation system. It is the sequence of question in which a way of a student thinking can be analysed.

c) A *one-way-choice* format, in a chain (or set) of answers. It reminds the alternative response frame but not any single answer is validated. The user has to select the path with several stages of the answers (sequences of answers).

2. THE CAL SYSTEMS DEVELOPMENT FUNDAMENTAL STEPS

THE SCREENPLAY

The CAL unit development starts from a lesson screenplay analysis. The developer has to analyse this course screenplay for finding break points where the course program will be divided into separate parts (or group of problems). After that these units have to be discussed looking at the questions/answers validation system as a fundamental engine of the application control unit.

The programmable teaching principles (Burke, 1982) provide the CAL packages developers with several classes of the question-answer formulation, anyhow only two of them can be recommended for the control unit's formulation:

- using an anticipated correct answers table,
- using an anticipated wrong answers table.

The interaction results analysis in majority of applications uses the anticipated correct answer mode. Although the answer is correct, with slight difference from the pattern, it can be assigned wrong. The only way for making these responses judge reliable is to prepare a very strict application form for avoiding formal mistakes then the interactions lead to remedial comments.

In case the anticipated wrong answers table used for making conclusions, every mistake is very easy extracted. Each mistake can be precisely matched with the remedial block. This way very individual program of the lesson is created. This criterion format operates on a more complex database anyhow the answer validation algorithm goes faster then in a multiple-choice format.

Considering the types of the lesson mode we can distinguish several logical principles that govern the application. Among them *didactics* where the student obtains the information in small portions then is asked to do elementary synthesis or manipulation with this data. The modern CAL systems have to be equipped with interference that allows going to the conclusions: from *examples to rules* or from *rules to examples.*

THE EXAMPLE FLOWCHART OF THE APPLICATION

Each CAL system consists usually of several units with several frames in each unit. Several presentation units given linearly are bounded together by frames containing tasks and questions (Piecha, 1989).

Each unit is built up of several frames executed frame by frame, sometimes with branching to additional examples, questions, simple tasks or comments. Any time it is needed the CAL application can be stopped. The application frames are signed by their unique descriptors that allow recalling every individual unit.

The application design steps

I. The screenplay actions:
   - select the key problems of the subject for which the lesson is recommended,
   - extract the problems separate units,
   - define the lesson functions,
   - choose a suitable criterion frame and the validation format adequate to the CAL system definitions,
   - make a primary screenplay of the lesson; frames and lesson units,
   - define the CAL program interrupting forms and their help layers.

II. The program part:
   - work out a prototype program of all CAL units and for their control sections.

III. The revision part:
   - make the initial revision of the CAL program; designer-assistant student,
   - improve all troubles making structures,
   - check the program in small groups of students,
   - make the program validation in a real lesson, in full teaching cycle with a group of students.

IV. The initiation part:
   - write the CAL guide-book for the user,
   - write the CAL guide-book for the teacher.

Verification - make the final improvement of the application.

Very useful validation measure is a time delay of the answer. Statistically fast but not correct answer can suggest a wrong formula of the criterion frame. When the answer is correct but the time delay is too large we can expect that the lesson needs additional comments.

In our experiments a 90/90 criterion has been used (Burke, 1982) where statistically 90% of the users understand 90% of the CAL content - without any additional explanations. When the experiment results are too good we can be sure that these resources are too simple to be recommended as the CAL system. Statistically different results than the "90/90" suggest corrections of the system.

3. THE CAL SYSTEMS EVALUATION

Evaluating lesson performances

First step of the evaluation process involves the author or an assistant training with one student at a time, carefully observing the problems appearing through the lesson. After it the lesson should be redesigned as required and next courseware review shall be done.

Field testing

The program cannot be considered the CAL system until it has been empirically evaluated. It has to be done by field-testing with a sample of students/pupils who are representative of the future system users:

- to evaluate the relationship of the CAL resources to other curricula materials,
- to elaborate the teacher's guidebook for the CAL system,
- to examine the effectiveness of the CAL lesson (not a necessary condition).

The lesson effectiveness

Effectiveness of the lesson is one measure used in scientific investigations; expensive and long lasting. In frames of pedagogical values it is considered as a necessary stage in the evaluation processes.

To evaluate effectiveness of Programmable Teaching processes, several methods can be used. In all experiments done by author of the paper three of them were applied: pedagogical experiment, pedagogical tests and as a supplementary form - active observation.

The results of *pedagogical experiments* depend on two fundamental conditions:

- CAL resources are large enough (!),
- the effectiveness of an application can somehow be measured.

The simplest way of the effectiveness measurement is comparison of *two parallel working groups:* the experimental group taught by the CAL resources and the comparative one, using traditional methods. The group selection has to be done on the ground of the *initial test*. The groups have to be equivalent in member number and in a knowledge stage. The duration of the lessons, the lessons program and the examination methods have to be the same. It would also be desirable that the same teacher does the courses in both groups.

For our experimental groups we selected pupils who got lower score in the initial tests. This negative selection reduces unexpected (negative) effects of the experiment circumstances.

After a full cycle of the course final tests have to be done. They consist of several questions (the most advanced - 20%) of the initial test and the final questions and tasks (80%). The criterion frames of the CAL resources cannot be directly repeated in the final test, as they are in favour of the CAL users.

The test statistics allow us to estimate the student's / pupil's knowledge increase. After a short period of time (three or four weeks) repetition of the tests (*distance tests*) has to be repeated. These new statistics indicate the *knowledge persistence.*

Our experiments with the CAL system, widely described in many papers of the author, gave very good results. The experiments lasted six months and the experimental groups got an average score one stage higher then the comparative groups. What is more the knowledge persistence in experimental groups also occurred higher.

The active observation is an evaluation method that can be recommended for small CAL projects. The CAL lesson is observed by the experienced teacher then all remarks have to be sufficient to improve the CAL system.

4. SUMMARY

The paper shows how complex tasks have to do the developer team of CAL applications. They have to integrate many interdisciplinary rules to achieve the expected pedagogical result.

Very important role in this CAL design processes plays the project manager, who has to be not only very experienced in computer programming but also in the art of a very specific pedagogy frames.

BIBLIOGRAPHY

[1] BURKE R.L. (1982). CAI Source Book. Englewood Cliffs: Prentice- Hall Inc.

[2] EBERTS R.E. (1986) Learning strategies in CAI design. The International Journal of Applied Engineering Education, No. 2/86, Oxford 1986.

[3] PIECHA J. (1989).The multilevel model for microcomputer ICAI systems.The International Journal of Applied Engineering Education, Vol.5, No 3/89, Pergammon Press.

[4] PIECHA J. (1991). Remarks to CAL systems designs. Proceedings of The Int. Workshop "Computer Aided Learning and Simulation Technologies". Prague,June 1991, pp.26-31.

[5] PIECHA J. The programmable shell for multimedia applications development. *Journal of Applied Computer Science*, Vol.7, No 2, pp.31-43, ISSN 1507-0360, Łódź 1999.

[6] PIECHA J. The Intranet Databases and some Approach Troubles into Multimedia Files. Proc. Int. Conf. *Computer Based Learning In Science - CBLIS'99*, Enschede, the Netherlands, 1999, G7.

[7] PIECHA J. The e-learning units characteristic features and evaluation procedures – resources and studies organisation, Proc of Int. Distance Learning Workshop' 04. May, 2004, ISBN 83-909518-6-X