DOCUMENT RESUME

ED 393 409 IR 017 745

AUTHOR Cambourakis, George; And Others
TITLE Education in Electronics: A Multimedia Metaphor.
PUB DATE 93
NOTE 8p.; In: Verbo-Visual Literacy: Understanding and Applying New Educational Communication Media Technologies. Selected Readings from the Symposium of the International Visual Literacy Association (Delphi, Greece, June 25-29, 1993); see IR 017 742. Contains some figures which may not reproduce clearly.
PUB TYPE Reports - Descriptive (141) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Authoring Aids (Programming); *Computer Assisted Instruction; Computer Simulation; Computer Software Development; Computer Uses in Education; *Electronics; Foreign Countries; Hypermedia; Instructional Materials; *Multimedia Instruction; *Technology Education
IDENTIFIERS *Software Tools; Technology Utilization

ABSTRACT

Those interested in creating or utilizing computer assisted instruction (CAI) tools for education in electronics have a wide range of multimedia options. Programs are becoming available which offer circuit simulations, mathematical packages for formula processing, instrumentation synthesis and simulation, drafting tools, hypertext construction aids and other authoring tools, object-oriented database capabilities, and multimedia peripherals for video and sound. This paper describes many of those options and proposes a CAI tool which contains an authoring module for the instructor, a student module containing the information content and help utilities, and an examiner module which creates questionnaires and other testing instruments. Optimum programming and implementation scenarios are explained. Five figures accompany the discussion. (BEW)

EDUCATION IN ELECTRONICS: A MULTIMEDIA METAPHOR

by

George Cambourakis, Eleftherios Kayafas, Vassili Loumos and Ioannis Tsatsakis

Introduction

Education, by means of computers, especially since CAI (Computer Aided Instruction) was introduced in the late 50's, went through a lot of stages between success and failure and faced a lot of criticism. After thousands of computer installations in elementary and secondary schools (Price, 1991) and the relevant studies, educational software has been enriched with important assistance tools from multimedia technology.

Great importance is given to the definition of the educational material by teachers themselves, something that in the first steps of CAI applications had been misunderstood. Today, nobody doubts the positive results of the computer's involvement in education. Some of the obvious advantages are:

- immediate access in every layer of knowledge.
- unlimited time of occupation,
- flexibility in the choice of presentation media, compared with the traditional supervisory educational assistance tools,
- reduction of the educational time (Price, 1991).

Education in electronics is a representative case that combines almost all the methods met in a CAI application, such as:

- hypermedia/hypertext,
- drafting, animation,
- topology recognition (in the sense of connection analysis),
- drill and practice methods,
- tutorial systems,
- simulations, similarities,
- metaphors,
- knowledge representation.

Aiding tools are available by now, in order to build an educational environment that simulates the tasks and the skills met in an electronics laboratory:

Circuit Analysis, Synthesis and Simulation Software

All these programs use sophisticated features developing complex circuits, which otherwise would require a great effort. But as the complexity of electronic design is increasing, the corresponding theoretical concepts are becoming more involved and complicated. From the student point of view, the effort spent in drawing lines, busses, etc., as well as learning basic and advanced theory is quite necessary, but also time consuming and very difficult to understand. Therefore, these sophisticated tools are not always suitable as teaching aids for basic courses in electronics and electronic design.

Mathematical packages for Formula Editing and Processing. One could proceed with these tools by constructing equivalents of transformation formula, and using them parallel with other applications.

Instrumentation synthesis and simulation, using special software and hardware peripherals. It could be an excellent assisting tool for simplifying the required materials.

Drafting tools for image processing and enhancement .

Context-sensitive help markers, that can help the author in the construction of hypertexts.

Authoring tools and authoring languages. Although authoring systems are easy to use and can reduce dramatically the developing time, a certain amount of time is spent in learning how to use them and how to build complex tasks, in which case, a third generation language would be easier, faster, and much more flexible.

Evolutionary database systems, using object oriented structures.

Multimedia peripherals implemented with the appropriate software such as video controllers and sound synthesizers.

Programming these tools is not always an effortless process. A certain capability in programming is required to connect and combine all these discrete tools into an integrated and interactive environment.

On the other hand, such an implementation using all these discrete programs as background processes would confuse the author and overcharge the trainees with additional commands and menus.

Furthermore, no simulation system will familiarize the trainees with electronic circuits, as much as a real electronic laboratory, but such an approach represents an inexpensive and very attractive solution.

Architecture and interface design have goals in common, to create active, attractive and operative environments. This principal will be kept in this paper.

Goals and Strategy

According to the above thoughts, a good approach is to build some of these tools considering the level of utilization. We will use the chapter of operational amplifiers as a tutorial example in introducing novice students to electronics through a computer environment, in a form of *experiment pages*.

This allows teachers to prepare effortlessly their educational material and the students to experiment on the behavior of electronic circuits. They can change the values of parametric elements and observe the output in a graphical window. The application is composed of three discrete modules: *The Author*, *The Student*, and *The Examiner*.

The Author is an interactive module that facilitates the preparation of the educational material, creates pages from images of electronic designs, organizes them in a object- oriented database, and defines text parts from the above icons as parameters, in order to obtain further calculation capabilities on transient formulae functions.

The Student module contains an introduction to the base theory of op-amps, help utilities, examples, and reference educational sources. It allows the construction of user sessions in conjunction with the desired skills taking into account the user's capabilities and knowledge. It enables the user to experiment on preselected *pages* by changing the values of the electronic elements and observing the output. In this way the user defined values are evaluated as formal parameters in the transient formula. The result is shown in a appropriate graphical window (virtual instrument).

The Examiner finally creates a questionnaire on the logic of the circuit to examine the user's understanding of the educational material. The module watches the user's achievement and degree of understanding. It controls the answers by scaled criteria and redefines the skill level by adapting the user's session to his performance.

Not all of the implementation depends on teacher's analysis, choice of educational, material, and authoring. An

effort should be made to solve the following problems.

1. Database consistency against improper operations by novices.

2. Compatibility between various available peripherals (e.g., different screen resolutions, different CPU speed that influences the animation speed, etc.).

3. Data exchange, while switching between various tasks (considering the logical synchronization and the computation steps).

4. Standard approach for various tasks in order to reduce the required amount of code.

5. Automation as possible, of the embedding and calling images, executable code, and other binary sources (MS Access User Guide).

6. Configuration of the application into a network.

According to the above list, it was essential to distinguish the implementation of the three modules in two development layers. These layers are authoring and programming, bounded by the criterion of independence between programs and data.

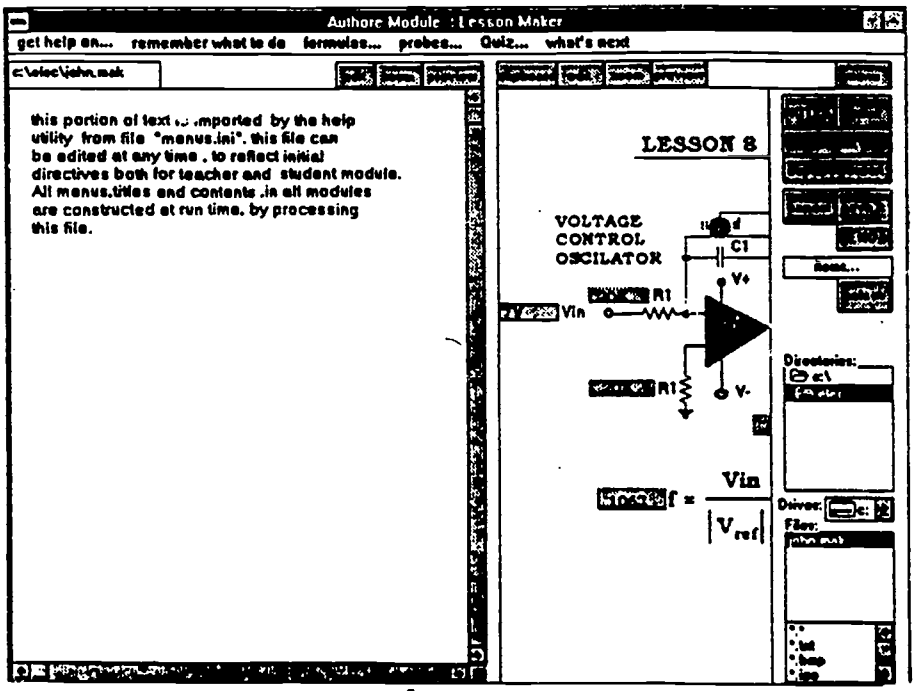
Authoring

The authoring layer is application specific and deals with the definition and implementation of:

- Data classification and structures (e.g., tables, indexes, trees, etc.).
- Authoring tools required for importing and processing data (forms, drafting utilities, etc.).
- Forms of the lessons (nodes in multimedia terms).
- Data capture for monitoring the student's activities.
- Protocols for data exchange between modules.

The diagram of Figure 1 is a representative form of the Authoring Screen.

Figure 1. Initial Screen of the Authoring Module



As we can see in Figure 2, the metaphors refer to seven basic tasks: Circuits, visual formulas, formula processing, virtual instruments, assist, query-examination, and activity acquisition. A further classification of data processing (icons, text, and keywords) leads to icons tasks, hyper tasks, and context tasks.

Circuits and visual formulas consist of bitmap icons edited to improve their appearance and facilitate hyper area boundaries definition.

Virtual instrumentation is based on LabView® which allows the measurement of a series of different parameters on the circuit, as well as their visualization.

In the case of formula processing, input variables are linked with the appropriate icons of the visual formulas and circuits. That way, a new value of an input parameter enforces updating of all related screen and background objects.

A hypertext case sensitive help is available through assist. Query and examination follows the predefined links in a sequential order according to the skill level.

Activity acquisition is the task designed to capture the answers and estimate the level of understanding by means of some explicit rules.

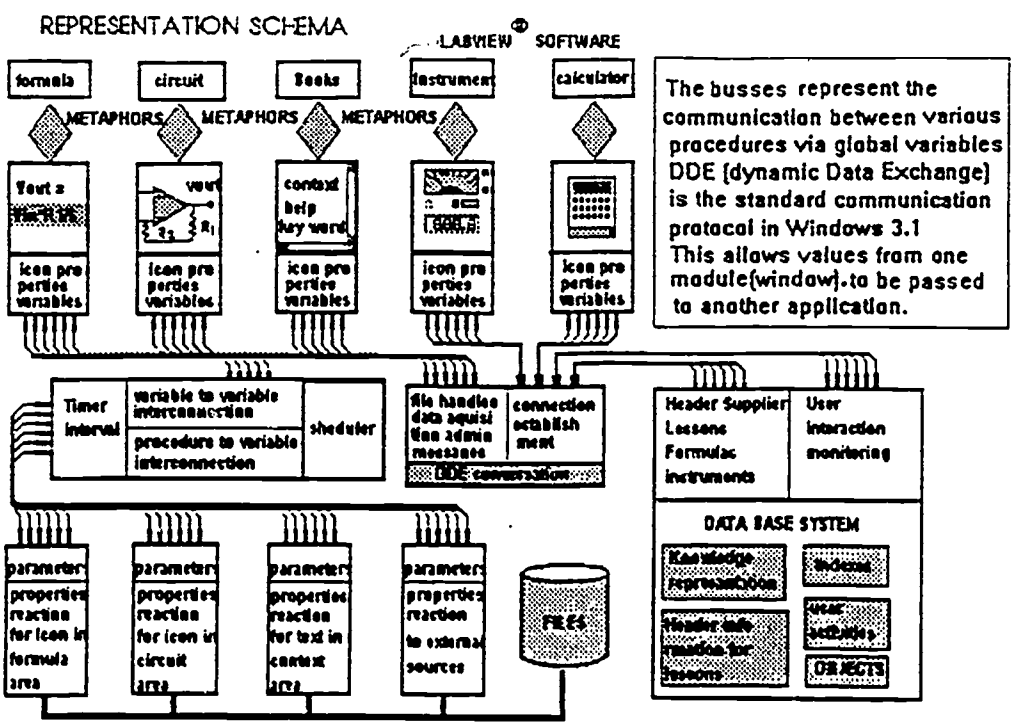
Programming

The programming layer deals with definition and implementation of instruction sets and procedures.

The instruction sets are used through the three modules in such a way that common tasks can share the same code.

On the other hand, procedures automate standard database tasks, reduce

Figure 2. The Structure of the Application



interaction with the operating system, and implement the data exchange.

In the center is the scheduler that performs all the necessary tasks that deal with the interconnections. It:

1. Retrieves information about users and the appropriate lessons, or stores constructed lessons.

2. Opens files and retrieves the necessary objects related to that lesson.

3. Gives the appropriate values and properties to the screen objects and at certain time intervals, updates global variables according to implied logic.

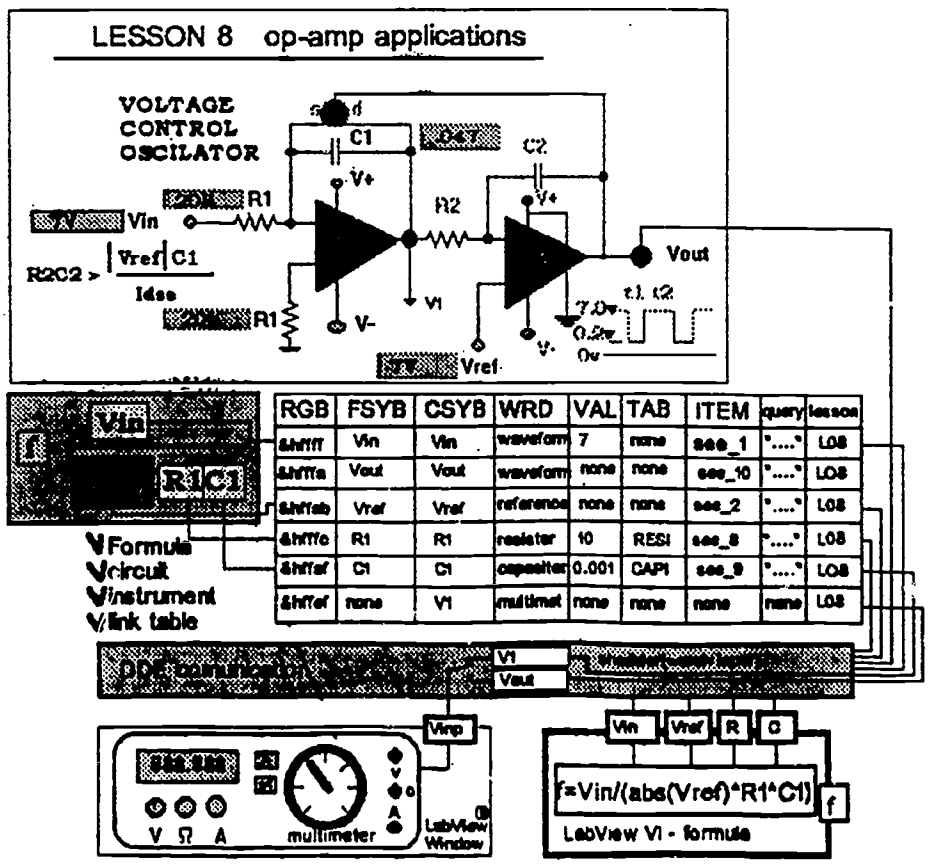
4. Establishes the communication with external applications (DDE method).

Finally, at run time, the scheduler is responsible for capturing the user's activities (mouse movements, key strokes, etc.).

According to an implied logic the scheduler interprets these activities into a tabular information. Any change to this information invokes the scheduler to check a table of values for integrity and to update the screen objects.

Figure 3 shows the database structure and all the necessary links between hyper areas, hyper-text symbols, and properties.

Figure 3. The Links Between Circuits, Formulas, and Instruments



Implementation

As it was previously mentioned, Assist is a hypertext case sensitive help system. To improve its performance, text is kept as items in a choice menu that organizes them in tree structure by means of menus and sub-menues. The titles of items serve as context help for all the application.

A very convenient way to implement a link between selected areas and their context can be done by using the hex code obtained from the selected (in sequential order) background color (see RGB column in Figure 3). These values (which must be reserved for hyper areas) can serve as an index in the symbol table. This method leads in a very fast and convenient hash function. The hyper areas appear in increment tones of a color.

The student has to invoke the examiner (questionnaire is opened at the lesson's session). Selected links and titles (see Figure 4) of questions are retrieved according to skill level. The questions are presented to the student and he/she has to click inside the right hyper area as a response to the question, or he/she has to calculate a result of a function and paste the answer in the calculator's box. The professor decides if any help will be available to the student. The same table of links used for the association of hyper areas and hyper text is also used for the examination.

The Acquisition task is designed to capture the answers and estimate the level of understanding by means of some explicit rules: Instead of considering a particular programming approach, a more generalized structure is constructed. It behaves as an interpreter of key words and expressions embedded in a script file. These keywords include logical operators, arithmetic operators, functions, directives, and application variables (field names). A rule then could be constructed simply as a line in the script file:

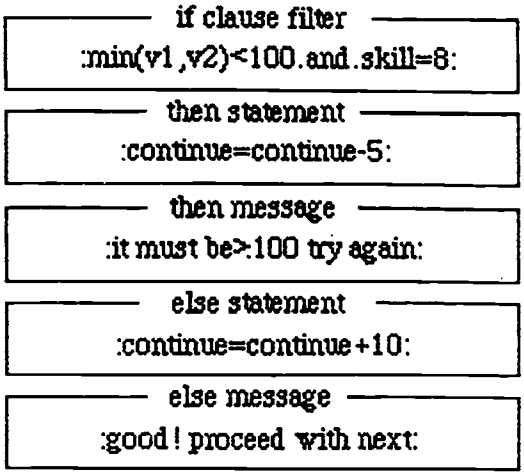


Figure 4. A Script to Implement a Simple Rule

The interpreter then is responsible to incorporate these instructions into the checking loop, in the form of macros and proceed with the control.

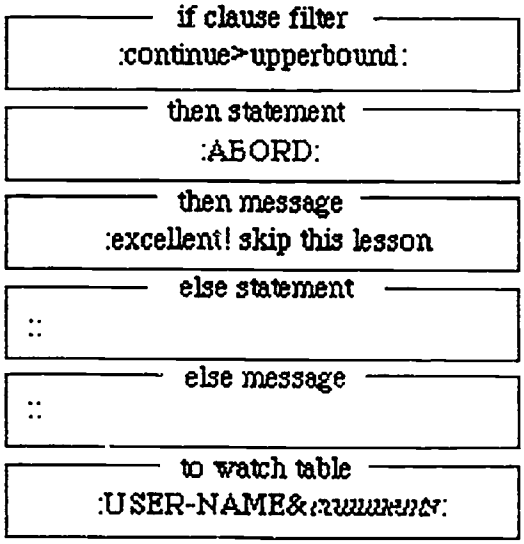


Figure 5.
A Script to Record Student's Score

By these means the teacher could construct without effort his/her own rules, without worrying about changing programs or adding new ones. Also he/she can obtain additional information, using the same structure, about user activities in a database table. As an example the above script file could contain a message for teacher purposes.

The Platform

- MS Windows 3.1® with the utilization of DDE. Dynamic Data exchange is a method supported by Microsoft Windows operating environment that allows two independent applications to communicate with each other in a client-server model and automatically exchange data.

- MS Access®, as database system for the examiner module implementation, MS Access Basic, as a programming tool for embedding user functions.

- MS Visual Basic® for stand-alone modules.

- LabView for Windows® (National Instruments Corp.), a graphical, visual programming tool for instrumentation (virtual instruments), such as signal generator (for drill and practice in students module), formula processing (by constructing background instruments as links to the formal parameters of transient functions).

Utilization of some Windows libraries (DLL) for screen capturing and construction of screen images.

Conclusions

In this paper we have demonstrated an integrated environment for education in electronics, both for novices and advanced students. Attention was paid to the role of the teacher. Course development time is reduced, to a minimum, without the utilization of special authoring tools and programming languages.

An effort has been made also to capture the user's activities in a flexible way so that a maximum interaction with the application can take place, both for the teacher and the student, with the minimum effort in programming.

References

Laurel, B. (1990). The art of human-computer interface design. Addison-Wesley.

MS Access user guide. MS Press.

Price, R. (1991). *Computer-aided instruction.* Brooks, Cole.

Van Horn, R. (1989). *Advanced technology in education.* Brooks, Cole.

Waern, Y. (1989). *Cognitive aspects of computer supported tasks.* Essex: John Wiley & Sons.

Waterworth, J. (1992). *Multimedia interaction with computers.* Ellis Horwood.