Int. J. Engng Ed. Vol. 10, No. 2, pp. 215–221, 1994
Printed in Great Britain.
0742–0269/91 $3.00+0.00

Frame-Oriented Intelligent Tutoring System*

M. S. WALD
B. SOLTYSIAK
Hamburg Educational Partnership, Berliner Tor 21, 20099 Hamburg, Germany

The frame-oriented approach to the learning process often used in CAL (computer-aided learning) has been applied to developing learning programs, supported by an expert system technique. In traditional CAL, knowledge is contained in presentation blocks, called frames, prepared by an instructor. These frames are displayed to the student under given conditions. Such programs are often organized, like books, into pages, sections, a table of contents, and an index. The drawback to this kind of organization is that there is no dynamic access to the contents, and the program is not capable of changing its presentations dynamically, to adapt itself to learner needs or to his/her knowledge level. A prototype learning system containing introductory materials for an electronics module has been designed that incorporates such requirements. In this module learners can follow an individualized path through the subject matter based on an assessment of their prior knowledge. Learners can test their proficiency via a special simulation block which can access a database. The database contains technical data for elements in the learning program which are being simulated. In the example presented the database is a catalogue of transistors.

INTRODUCTION

IN TRADITIONAL computer-aided learning (CAL) programs, the subject matter is often organized like a book with all features emulating the hardcopy platforms. A book can be quite sophisticated and support various levels of reading, inviting the advanced reader to skip or select certain sections, or provide a glossary and cross-references between sections. If we imagine active books that can interact with the reader and communicate knowledge at a selected level, recalling relevant information, probing the reader's understanding, explaining difficult areas in more depth, skipping over known material, we are dealing with an *intelligent knowledge communication system* [1–3].

We are dealing with such a system here. Our purpose was to design a CAL system that can assess the learner's knowledge level before he/she starts his/her learning routine. The system selects with the aid of a filtering algorithm a customized learning path, and then modifies its presentation dynamically as accumulated skills and performance levels develop.

THE APPLICATION STRUCTURE

The core idea is contained in the structural layout of the tutorial. This block consists of units called frames, presented to the learner.

Each frame is divided into four subframes containing knowledge at different levels. These levels can be conceived of as basic, medium, advanced and expert, for example.

The general program flowchart for the tutorial is presented in Fig. 1.

An individual learning path can be assembled in two ways:

- A path recommendation determined by the expert system based on a series of test questions ('Input Test' section, Fig. 1).
- A path chosen by the learner, based on his/her desired starting level.

In the first case, different parts of the tutorial in the form of frames are presented at levels based on the learner's prior knowledge, which was assessed before beginning the learning process.

In the second path mode, the tutorial frames are presented upon demand.

After defining an input level, the program assembles an appropriate set of subframes and prepares an individualized learning path tailored for the learner.

The learner then begins the actual tutorial. His/her skills and performance level from the tutorial are registered, and then compared with the entry level.

Subsequently, a new set of subframes for further learning or repetition will be automatically set up, or an evaluation regarding the learner's performance will be given.

* Accepted 2 July 1993.

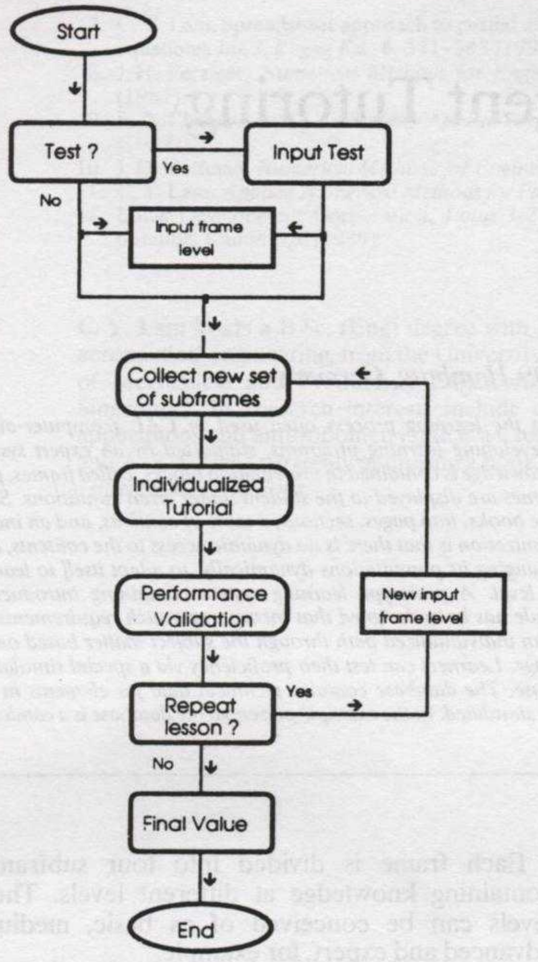

Fig. 1 The application structure flow chart

THE MAIN APPLICATION BLOCKS

The application consists of the elements known in Fig. 2. The main control block was designed with the *KnowledgePro object-oriented language* [4], and has five sections.

The sections have the following tasks:

- **Questions Section**—assessing the learner's knowledge based on questions to, and by the learner.
- **Help Section**—based on *Hypertext* techniques, using as a source, a special hypertext file.
- **Tutorial Section**—created with *Authorware* [5] (*authoring system*) as a CAL program with book-like features.
- **Inference Engine**—comparing the performance of the learner from the tutorial with the input knowledge level, and assembling an optimal path for further learning, or a final mark of the learner's learning achievements.
- **Simulation Section**—simulating the three basic types of transistor circuits with a link to the data base.

TUTORIAL SECTION

The tutorial section is programmed in Authorware as a separate standalone program block that has its own presentation window and is accessed from the main control block. The communication between them, for the process of transferring data, is provided by the external text file.

The tutorial receives, from the control block, an array of variables determining a set of subframes chosen for the presentation.

In return the tutorial sends back results of the learner's performance. This section on the main level (Fig. 3) appears like a conventional learning program prepared under Authorware, presented as an electronic book or a CAL program.

There is an introduction and a menu dividing the contents of the tutorial into different chapters. The calculation icon called *Read variables* is responsible for transferring values from the control section to the tutorial according to the level of knowledge of the course or section.

The structure of the chapter has the features shown in Fig. 4.

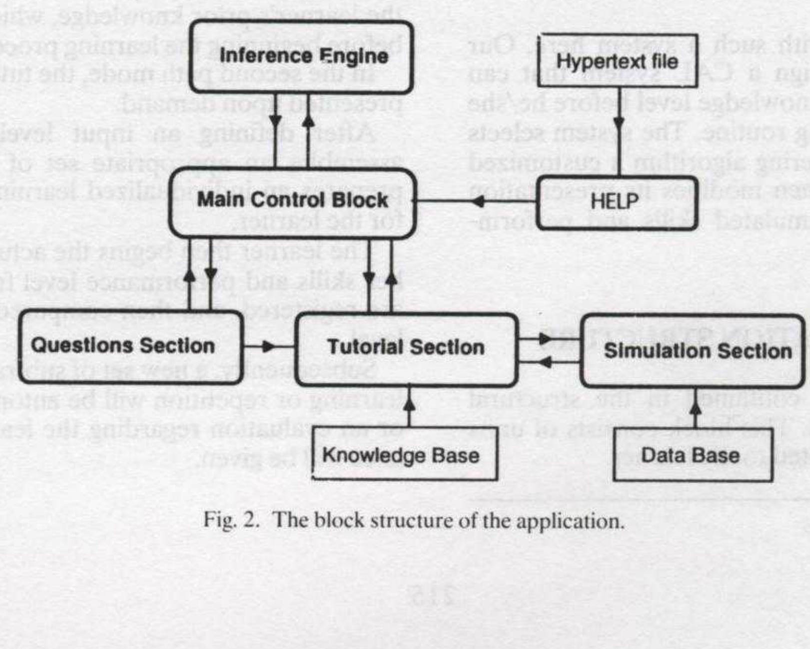

Fig. 2. The block structure of the application.

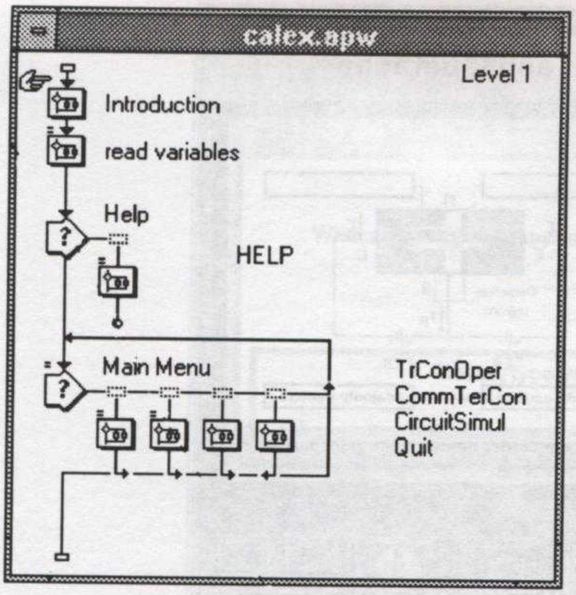

Fig. 3. The main flow chart of the tutorial.

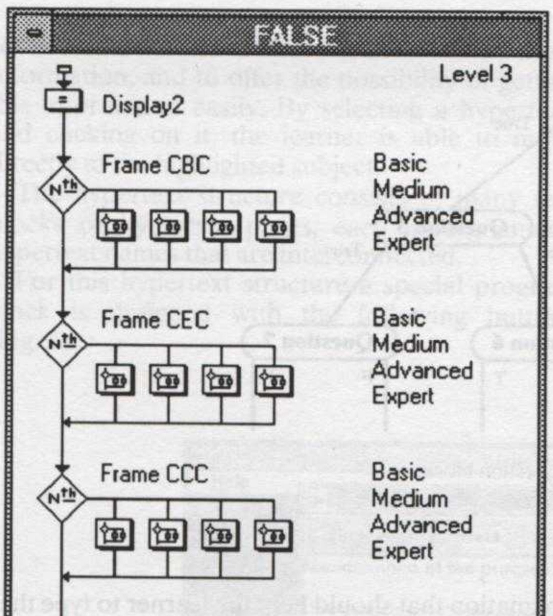

Fig. 4. The structure of a tutorial chapter.

Each chapter is divided into units, called frames, that contain small portions of a curriculum which are successively displayed on the screen like pages from a book. Some of them also have multimedia interactive features.

The knowledge concentrated in the *Knowledge Base* is in the form of elementary objects, e.g. text, graphics, stills, video, voice.

A frame can consist of a single topic (one page), or a block of topics made up of several pages. Each frame within the chapter includes four equivalent separate blocks of different knowledge levels (subframes). During the learning process only one of them will be presented to the learner according to his/her knowledge level or his/her deliberate choice.

This means that different frames have been presented at different levels according to the expertise from the questions section. If the learner does not want to get an assessment of knowledge, he/she can select the desired level, and decide that all frames will be presented at the same level of knowledge.

A typical screen from the tutorial, is presented in Fig. 5.

There are two screens with defined purposes. The screen on the left is used for text. The screen on the right is for images, animation, videos, etc. Sometimes the right-hand screen is used for interacting with the learner, e.g. asking him/her some questions, or giving him/her some tasks to do.

The correct answer to the question allows the learner to proceed. A wrong answer will effect some hint to help him/her to find the correct one.

Apart from these screens, there is logistic information, such as the name of the chapter, the number of the current page and number of pages in the chapter.

The following buttons are also included.

- INDEX—to access directly the topic the user is interested in.
- MENU—to return to the main tutorial menu,
- HELP—to obtain additional information on how to use this section,
- Direction buttons to the next or previous pages.

The number of pages in every learning loop may be different, because the program is collecting different subframes dynamically related to actual requirements.

QUESTIONS SECTION

This part of the application is responsible for assessing the learner's knowledge before he/she starts the actual lesson. Its structure is directly related to the structure of the tutorial. For all frames of the tutorial, separate sequences of questions are prepared. The structure of such a question block is a tree (Fig. 6).

The path of the learner through the tree depends on his/her answers. As a learner moves through the block, assessment of his/her knowledge on a certain topic is stored, and is directly linked with the question route the learner has taken.

How to stock such a tree with questions, and how to validate the answers, is a separate problem that should be solved with the help of an expert and experienced instructor.

During their interaction with the system the learner is allowed to ask questions about terms he/she does not understand. The type of questions

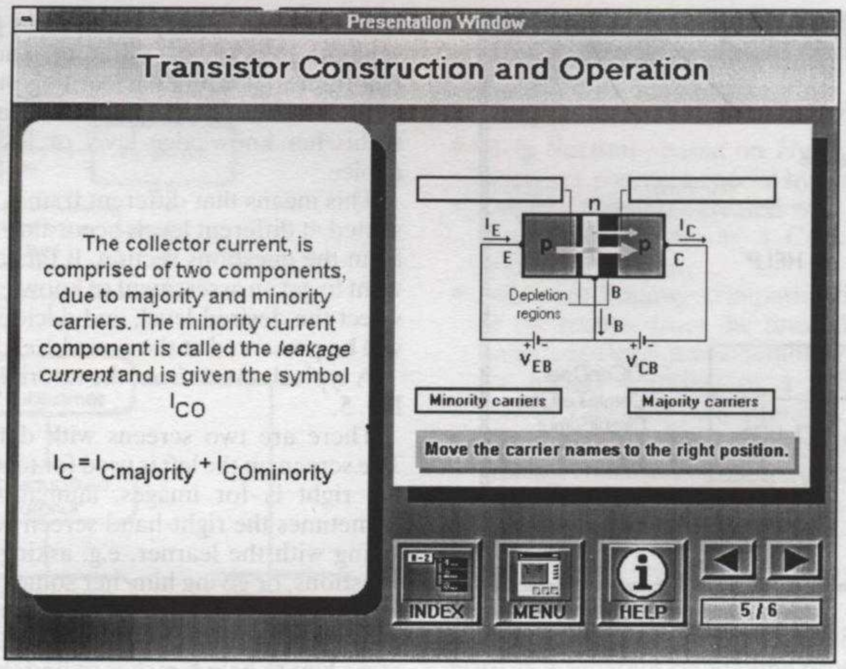

Fig. 5. A screen from the tutorial.

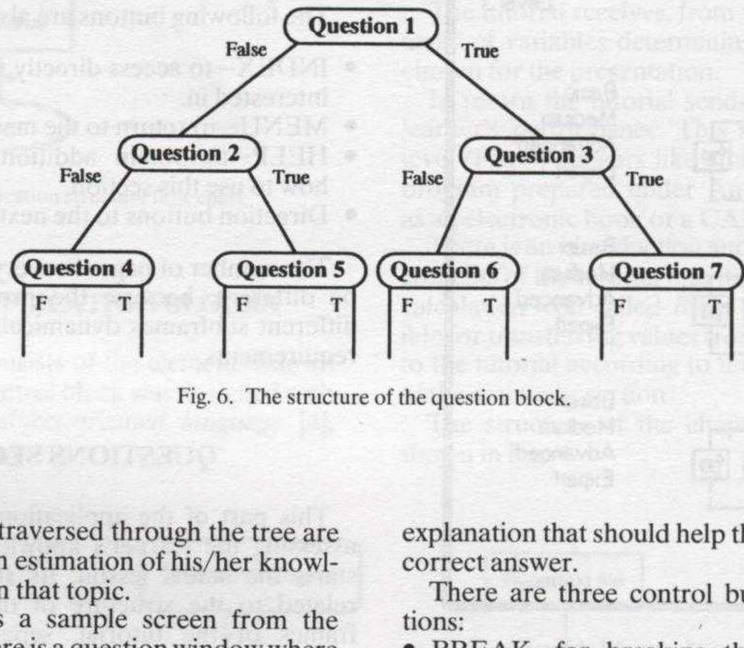

Fig. 6. The structure of the question block.

asked and the route traversed through the tree are used for preparing an estimation of his/her knowledge and expertise on that topic.

Figure 7 presents a sample screen from the question section. There is a question window where the learner is asked a question and should answer, in one of the following ways:

- answering YES or NO;
- choosing the correct answer from a set of possible answers;
- filling up a missing part of the answer;
- typing his/her own answer.

The second, smaller window, is opened upon request by the learner in case he or she wants to enquire about an unknown term used in the question. Once the question has been typed and processed, the program registers it and gives an explanation that should help the learner to type the correct answer.

There are three control buttons for the questions:

- BREAK—for breaking the present test and going back to the previous menu;
- QUESTION—for asking questions about difficult terms;
- NEXT—for going to the next question from the tree.

HELP SECTION

This section has been designed using *hypertext* techniques to highlight the advantages these provide in designing a CAL program. It is intended to

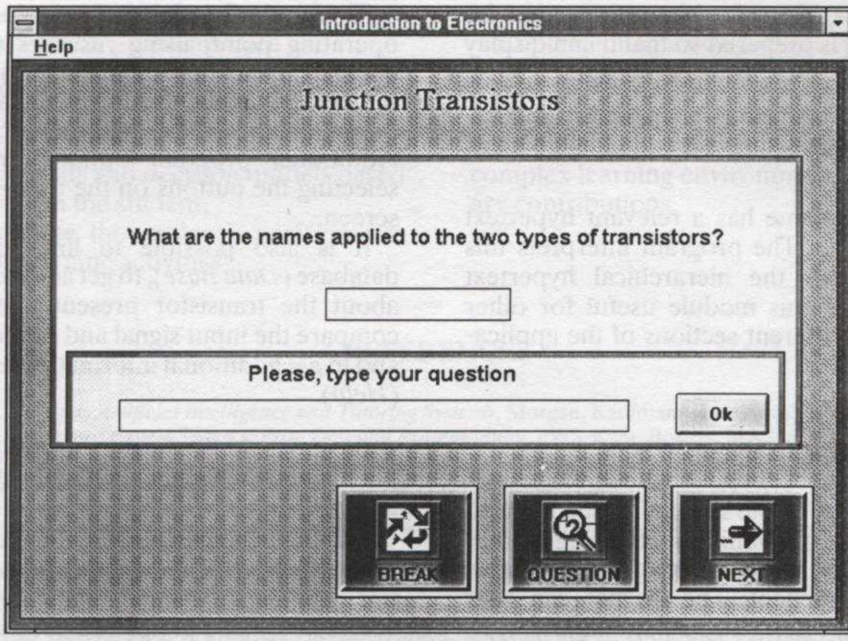

Fig. 7. A screen from the questions section.

let the learner know where he/she can find more information, and to offer the possibility of getting this information easily. By selecting a hypertext, and clicking on it, the learner is able to move directly to the highlighted subject.

The hypertext structure consists of many text blocks or hypertext pages, each having unique hypertext names that are interconnected.

For this hypertext structure a special program block is designed with the following buttons (Fig. 8).

- INDEX—to go to the main hypertext page (main page in the hierarchy tree);
- BACK—to return to the previous hypertext page;
- PRINT—to print the current page;
- QUIT—to quit the help section;
- HELP—to inform the user how to use this section.

The help section provides information about the whole learning program and the options from the

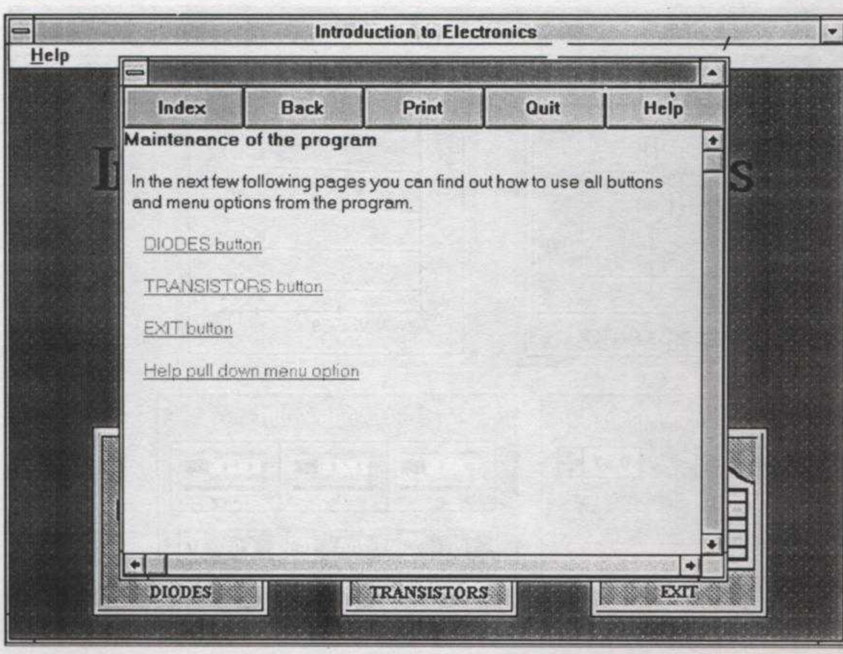

Fig. 8. The main hypertext help window.

main control block. To make this section useful for other purposes, it is prepared so that it can display almost any text file, using two special signs;

// points out the title of a hypertext page;
#m points out the start and end of a hypertext name.

Each hypertext name has a relevant hypertext page inside this file. The program interprets this text file and builds the hierarchical hypertext structure. It makes this module useful for other help purposes in different sections of the application.

SIMULATION SECTION

The main purpose of the simulation section is to give the learner the possibility of checking his/her practical knowledge, and simulating real situations in a virtual environment. The learner can thus simulate different, often extreme situations, without using expensive equipment.

The database connected to this block helps the learner to simulate many different types of elements in certain, stable circumstances, without the need for any physical changes.

This particular case is a simulation using three basic transistor circuit configurations. It shows the influence of the selected circuit parameters on the behaviour of the transistor and the performance of the whole circuit.

The linked database contains a catalogue of transistors with all the necessary technical parameters.

Figure 9 presents the main screen from the simulation section.

The learner's task is to determine the transistor's operating point, using resistors (R–R4), and to obtain a given output signal. During this operation, the screen shows the input signal, the transistor output characteristics and the circuit being tested. Additional program options are available by selecting the buttons on the right-hand side of the screen.

It is also possible to link to the transistor database (*Data Base*), to get additional information about the transistor presently in use (*Info*), to compare the input signal and output signal (*CRO*), and to get additional information about this section (*Help*).

SUMMARY AND OUTLOOK

The described learning system has been initiated and formalized. The structure for particular blocks, and interfaces between them has been designed and tested. An example has been developed as a prototype for a CAL system extending beyond presently common features of such programs. In addition to AI elements, the software tools KnowledgePro and Authorware are able to resolve programming problems easily and effectively. The incorporation of new multimedia features should make this system more interesting and efficient. However, it is only a step towards achieving the final goal, which is an application that is able to follow defined directions and fulfil its basic aims of designing effective learner support in a complex learning process.

The idea is to give the learner the possibility of navigating through the subject on their own with minimal tutoring.

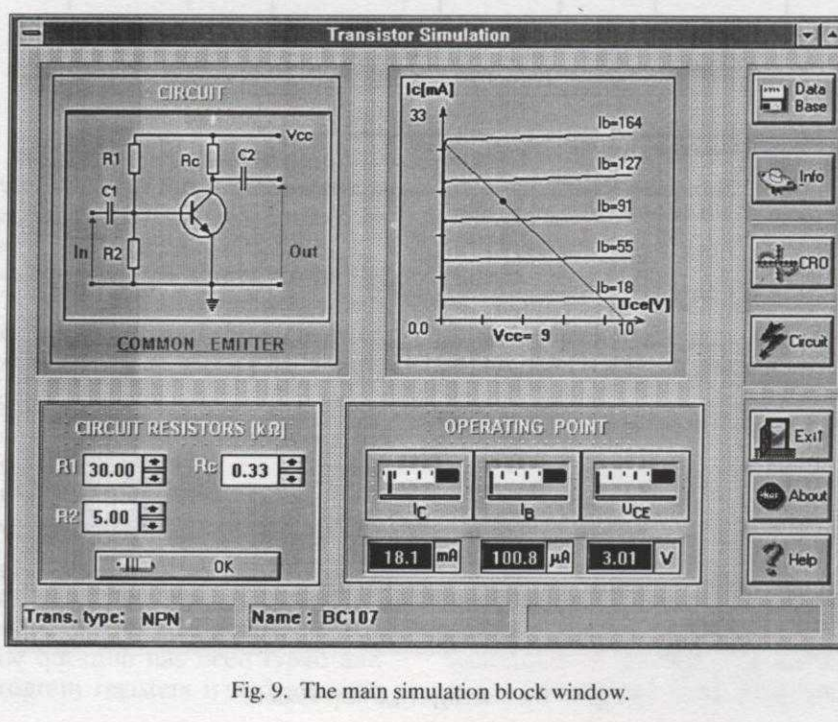

Fig. 9. The main simulation block window.

Further development of this software environment will concentrate on problems such as:

- how to formalize the choice of a scope of knowledge, for different levels of frames (subframes);
- selection of methods and decision models based on interaction with the student;
- criteria to estimate the student's performance during his/her learning activities;
- deduction processes that take place before implementing a new learning path.

The process of building a modern CAL program for engineering will incorporate elements of science education and engineering disciplines. It is a complex learning environment with interdisciplinary contributions.

REFERENCES

1. E. Wenger, *Artificial Intelligence and Tutoring Systems*, Morgan, Kaufmann, Los Altos, CA (1987).
2. J. Girratano, *Expert Systems: Principles and Programming*. PWS-Kent, Boston, MA (1989).
3. G. E. Luger and W. A. Stubblefield, *Artificial Intelligence and the Design of Expert Systems*. Benjamin/Cummings, Redwood City, CA (1989).
4. KnowledgePro Windows, Knowledge Garden, Inc. (1991).
5. Authorware Professionals for Windows, Authorware, Inc. (1991).

Bogdan Soltysiak is a graduate in electronics, specializing in computer science, from the Technical University of Wroclaw, Poland. He subsequently worked as a scientific assistant in the Institute of Technical Cybernetics at the University and as a logic simulator designer in the USA. He was recipient of a European Community TEMPUS grant in 1991. He is currently a Ph.D. student with a DAAD grant at the Hamburg Educational Partnership.

Michael S. Wald is editor-in-chief of this journal, and Director of the Hamburg Educational Partnership. One of his current interests is the development of integrated computer-aided learning tools for engineering education.