TEACH: an approach towards Intelligent CAI

A. N. Kaplanis, M. Karavelaki

University of Athens

ABSTRACT

This paper presents an effort towards the design and implementation of an educational program according to ICAI techniques. Its components - knowledge network, teaching rules, diagnostic rules and student model are explicitly defined around the course-manager module, which controls the coordination of all the components. This approach allows the teacher to access and adjust each component according to the specific needs of the educational environment he is going to use.

These features are discussed each in turn, looking at the general requirements/specifications and then following them through the design of a small prototype: TEACH. This prototype has two purposes: to examine the possibility of creating such a program and to explore, by extension, the generality between types of ICAI programs. In order to gain from the embedded inferencing capabilities, TEACH was implemented in Prolog.

INTRODUCTION

The use of computer-driven delivery systems for education and training has experienced a dramatic increase over the last decade. This increase can be explained by the rapidly maturing field of computer assisted instruction and its demonstrated effectiveness in education and training [PARK-87].

A more recent form of computer-delivered instruction is intelligent computer assisted instruction (ICAI), the application of A.I. technology to instruction. The ICAI programs are required to improve tutoring, solving the problems that appear on the use of CAl programs. Some of these problems are:

a. The CAI programs are created first by determining how a good teacher would respond to each possible student action and then building a branching program with each response explicitly programmed.

b. In case someone attempts to make any changes in the program, he has to restruct it to make room for the additional information and to connect the changes to the appropriate instructional rules.

c. The adaptive capability to the individual student`s learning needs is minimal because the questions are randomly selected within a given range of difficulty level.

ICAI is an effort to develop more powerful and accurate adaptive instructional systems by applying AI principles and techniques.

THE COMPONENTS OF A TYPICAL ICAI PPOGPAM

An ICAI program aims in allowing both the student and the system a flexibility in the learning environment that closely resembles what actually occurs when student and teacher sit down one-by-one and attempt to teach and learn together.

Its structure consists of the major components of an instructional system [WALLACH-87]: the domain expert, the teaching expert, the diagnostic expert and the student model.

The domain expert contains the knowledge (procedural and factual) that the student needs to learn. This knowledge is used to answer student questions in a mixed initiative tutor or to solve problems generated by the teaching expert.

The diagnostic expert uses rules to analyze student responses. Based on those responses, it makes hypotheses about what knowledqe the student has acquired or what misconceptions the student may have.

The student model is the model of the student's understanding of the domain knowledge as perceived by the ICAI system.

The teaching expert selects the appropriate strategy for teaching the student based on the current state of the student model. This includes selecting and presenting new information and posing questions or problems to the student.

AN ICAI PROTOTYPE: TEACH

Features from the above modules are used for the implementation of an ICAI prototype named TEACH. The major characteristics of TEACH are the following:

* Provides an environment where individualized instruction is offered.
* Its structure is based on the modules described above, so that the modification of each module can be easily performed by a trained teacher.
* The domain knowledge consists of first-degree equations.
* Incorporates a solver capable of solving any equation.
* Provides a limited kind of student answers` analysis.
* It has been implemented with Turbo-Prolog (version 1.1)

More analytically, the domain expert contains the knowledge that the student needs to learn and the exercises/problems that are going to be posed to the student. For each exercise the teacher provides the system with a scale of difficulty, an exercise id-number, as well as the body of the equation. The scale of difficulty is necessary for the correct application of the tutoring rules.

The diagnostic expert contains the diagnostic rules and the solver. This is capable of solving first-degree equations of any form, while the diagnostic rules provide a simplified form of answers' analysis.

The teaching expert contains the tutoring and therapy rules, needed to form the appropriate strategy according to the particular student. Rules are activated based on information taken from the student model.

Finally, the student model registers the student behavior, as he solves each exercise presented to him.

These components are explicitly defined around the course-manager module, which controls their coordination. It consists of a set of inferencing rules, with which it can produce a detailed instructional sequence, defined according to the particular needs of each student.

TUTORING SCHEMATA

The teachers' classroom decision making can be seen as the implementation of previously planned activities accompanied by a monitoring of classroom events. Deliberate decision-making is activated when monitoring reveals too great a discrepancy between anticipated and actual classroom behavior and is essentially remedial [GOODYEAR-87]. Thus, the main feature of the skilled teachers' knowledge structure is a set of schemata for teaching activities [LEINHARDT-86]. As schema can be named a complex knowledge structure composed of interrelated sets of organized actions.

The main schemata for the teaching activities, which we also build in our system, are:

a. Exploration Schema: It aims to explore the parameters collected in the student model and by using the tutoring rules described above to plan the next task.
b. Task Presentation Schema: This schema is activated when the system/tutor wants to present the material to the student.
c. Evaluation Schema: This schema is activated after the student's response in order to evaluate his answer.
d. Remediation Schema: Is responsible to assist to the therapy of the student's misconceptions, by activating the appropriate teaching strategies [WOOLF-87].

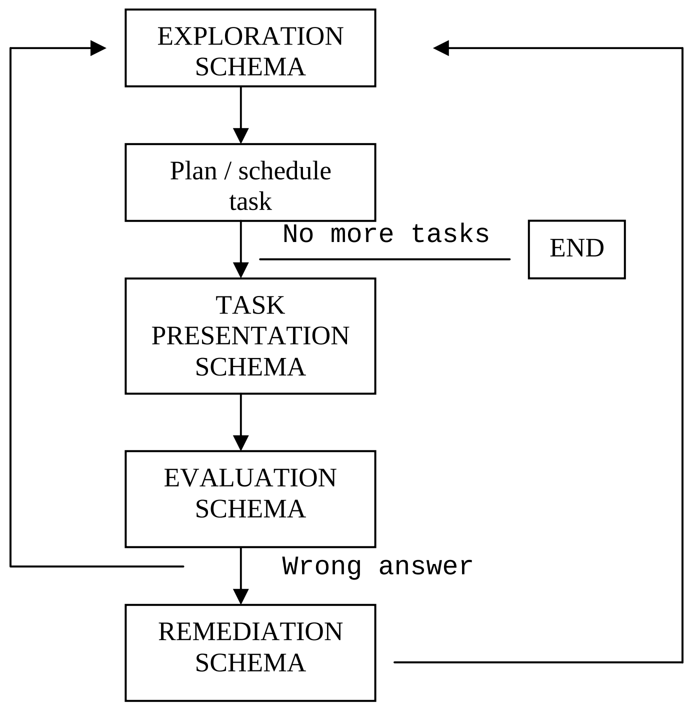

Figure 1

In TEACH the function of the above described schemata is implemented according to the following:
The exploration schema is responsible to determine what the student knows or does not know. According to this information, the system plans the next task to be presented. In case there are no more tasks, the tutoring session ends, otherwise the task presentation schema is activated and an exercise is presented to the student. The evaluation schema is activated next, to analyze the student`s response and update the student model accordingly. This is fulfilled by using the diagnostic rules and the solver in order to identify misconceptions and mistakes in the responses produced by the student. The teaching process continues until the particular type of misconception does no longer exist and only then the session moves to the next difficulty level.

CONCLUSIONS

The above described work explores the possibility of creating a flexible tutoring environment for the student to be taught and for the teacher to modify its components easily, according to the tutoring schemata which express the teachers' classroom decision making. It also explores the generality between types of ICAI programs, in an effort to compile in one program the general principles of the different programs which have been created till now.

TEACH, as a prototype, is a first attempt to implement the components of an ICAI program. Further work is in progress to expand its components, producing a detailed instructional sequence, for the needs of the individual student.

ACKNOWLEDGEMENTS

The authors would like to thank Professor G. Philokyprou for his guidance, as well as, the research team of the University of Athens for their cooperation during this project.

The research reported in this paper was partially sponsored by the General Secreteriat of Research and Technology.

REFERENCES

[PARK-87]: O. Park, R. Perez, R. Seidel : Intelligent CAI: Old Wine in New Bottles, or a New Vintage?, in "AI and Instruction" edited by G. Kearsley, Addison-Wesley Publ., 1987.

[WALLACH-87]: B. Wallach : Developing Strategies for ICAI on Small Computers, in "AI and Instruction", edited by G. Kearsley, Addison-Wesley PubI., 1987.

[GOODYEAR-87]: P. Goodyear : Approaches to the Empirical Derivation of Teaching Knowledge for ITS. European Seminar on ITS, October 1987, Tubingen.

[LEINHARDT-86]: G. Leinhardt, J. Greeno : The Cognitive Skill of Teaching, in "Journal of Educational Psychology", vol.76, 1986

[WOOLF-87]: B. Woolf, T. Murray : A Framework for Representing Tutorial Discourse, University of Massachusetts, 1987.