Example 1
- **Demonstration:** Teacher demonstrates features of a software program to the whole class. (Wall charts summarising key steps may be displayed beside the computer.)
- **Application:** Students (individually or in small groups) practise using the software and reinforce the understanding of its features by using it to complete a task (set by teacher or negotiated with students).
- **Reflection:** Students reflect on their experiences, and discuss successes, problems and strategies.

Example 2
- **Exploration:** Students (individually or in small groups) explore the capabilities of a software program through experimentation and play. The time allocated may be brief (one half hour session) or may extend over a whole term.
- **Reflection:** Students reflect on their discoveries and report to the whole class. The class may create wall charts or booklets of their collective understanding.
- **Application:** Students use the program to complete a task (set by the teacher or initiated by students) or use the program when it is appropriate.

Example 3
- **Exploration:** An individual student or small group of students explore a software program.
- **Demonstration/application:** 'Expert' students tutor other class members in the use of program while working collaboratively on a task provided by the teacher.
- **Reflection:** Students reflect on their experiences and discuss achievements, problems and strategies.

Example 4
- **Exploration:** Students work on a task individually or in small groups.
- **Demonstration:** The teacher introduces the class to a software program which throws new light on the task or allows the task to be completed in a different way.
- **Applications:** The students continue, extend or repeat the task incorporating their skills in the use of the software program.

**Computer-assisted knowledge acquisition:**
*Let's have theory-based instruction*

HELGA A. H. ROWE
Australian Council For Educational Research

This paper argues that computer use in classrooms should be based on pedagogical aims and principles rather than on trial and error, and that it should aim to facilitate the acquisition of knowledge as well as skills. To accomplish this, methods of teaching must focus primarily on the characteristics of the student and his/her individual needs with respect to the knowledge and understanding to be acquired, and only subsequently on the technology.

**SOME PROBLEMS WITH COMPUTER-ASSISTED INSTRUCTION**

Until recently, the way in which computers were utilised in learning and teaching was based totally on the teacher’s expertise in the subject content, supplemented by common sense ideas about computing. Many recommendations have been made as to how computers can be utilised more effectively in schools. Most of these recommendations can be summarised into four broad areas: the school culture and organisational structures should encourage the serious use of computers; sufficient numbers of computers need to be available to assure regular access for teachers and students; ample support for teachers in learning and planning how to use the technology is required; and ongoing technical support is essential. These are important conditions, but in such a complex and multifaceted endeavour of instructional innovation, I believe that there is at least one other factor to be considered: the pedagogical implications of a computer-supported learning environment.

In many classrooms technological issues are considered first and hence command most of the available resources. Currently, the main pedagogical activity in computer-assisted learning is to monitor practice, and the acquisition of some basic skills in the operation of computer hardware and software. To be sure, these are important activities, but it is even more important that teachers facilitate the use of activities which lead to the attainment of the basic educational goals for their students. Much of the theoretical knowledge concerning the principles underlying human cognition, which has emerged over the past three decades, has been influential in the design of methods of learning and teaching throughout the curriculum. What appears to have been overlooked is that theory-based design of instruction is equally important in the hi-tech classroom. The use of computers in classrooms has largely been a matter of trial and error, supplemented by the teacher’s common sense ideas about how
computers might be used. The teacher comes up with a particular design or application and tries it out. If it does not work he/she generates another design etc. Since the set of possible designs is rather large, progress can be expected to be slow.

**DESIGN OF THEORY-BASED INSTRUCTION**

Theory-based instruction begins with a knowledge of the principles underlying the mental processes that are involved in learning a particular content or subject matter. Instruction is then designed to facilitate those processes. Theory-based instruction is based, above all, on the characteristics of the learner rather than on the structure of the subject matter or the technology - even in computer-supported learning environments! If the focus of education is to encourage students to think independently, instructional processes need to be learner-centred. The learner has to be given the opportunity to process information, to ask questions, to solve problems and to make decisions. In computing as in all areas of learning, knowledge is not imparted to the learner, but acquired (i.e. constructed) by the learner through inquiry processes (Rowe, 1993).

Research into learning and teaching has shown that learning is an active, constructive, cumulative and goal-oriented process (Shuell, 1986). Resnick (1987) expects schools to prepare students to be adaptive learners and to focus attention on independent thinking and learning. In addition, learning should be situated so that students can acquire an understanding of important ideas for application to life outside school (Gardner, 1991; Resnick, Levine & Teasley, 1991) To accomplish this, students need to be given sufficient opportunities to think independently, critically and creatively, to construct meaningful knowledge through classroom interaction, and to develop personal appreciation and ownership of the knowledge they have constructed (Brophy & Alleman, 1991).

It is obvious that computers will have a part to play in supporting the above discussed processes. But using computers in classrooms does not necessarily result in the development of these processes, or other long-term cognitive and affective gains for the learners. The use made of computers in the classroom can be conceptualised as a continuum of varying degree of learner control (Lai, 1992). At one end of this continuum computers are perceived as powerful tools (i.e. teaching and learning aids) designed to increase the efficiency and effectiveness, precision and reliability of the instructional process with respect to existing curriculum goals and practices - possibly at the expense of a range of educational goals such as independent thinking, creativity, knowledge construction, and flexibility. (Geiss, 1987). The most common applications of this type of computer-assisted instruction take the form of drill and practice or relate to situations where computers are used as tutors to individualise instruction.

At the other end of the continuum, the emphasis is not so much on what the computer could teach the student, but on what the student could do with the computer. Here learning takes place in wider social, educational and physical contexts (Moore, 1987) where students acquire cognitive, metacognitive and social skills, solve problems, make decisions, develop a commitment to their academic goals, and construct knowledge and meaning for themselves. The use of programmable systems such as Logo (Lai, 1990; Lai, 1993; Papert, 1987; Scott, Cole, & Engel, 1992; Yelland & Masters, 1995) can provide this type of rich and challenging learning environment for students (Blumenfeld, Soloway, Marx, Krajick, Guzdial & Palincsar, 1991).

**CONSIDERING LEARNING PROCESSES**

A major function of instruction is to help the learner to perform a given task and to acquire knowledge and understanding in given domains. Theory-based instruction is guided, above all, by the characteristics of the learner, rather than by the content of the subject matter and the structure of the technology. Designing instruction in this way requires a knowledge of the relevant learning processes for the student. Instruction is then designed to facilitate these processes.

For example, the memory requirements and the limited capacity of human memory influence the processes of learning and understanding. One might consider Kurt VanLehn's research (VanLehn, 1986) into the acquisition of the algorithm for subtraction with regrouping. On the basis of extensive analyses of students' work on subtraction problems, VanLehn found that primary school children perform better when they induce the subtraction procedure from solved examples (rather than, say, construct it on the basis of verbal explanations given by the teacher or in a textbook). Verbal explanations are likely to require more memory, than learning from solved examples. Similarly, the grammar of a language is usually learned by inference from a set of sentences which are grammatically correct, and (optionally) a second set of sentences which are grammatically incorrect. Abstract ideas and general concepts are learned from examples and (optionally) counter examples. Sweller and his colleagues (Paas, 1993; Sweller, 1988; Sweller, 1993; Sweller & Chandler, 1991) obtained substantial improvements in performance by having students work on sample problems which were already partially or totally solved.

Other learning theories have other implications for instructional design. For example, the theories of discrimination learning (Langley, Simon, Bradshaw & Zytkowski, 1987) imply that understanding will be facilitated if the learner is supplied with both negative and positive examples of the rules he/she is supposed to learn. Both the theory hypothesising the efficacy of subgoals (Olsson, 1987) and the theory of chunking proposed by Laird (1986) claim that rule learning occurs when a subgoal is attained; hence, they imply that instruction should supply the learner with well sequenced, easy-to-reach subgoals.

Each type of learning process has particular obstacles associated with it. Instruction should be designed to remove or circumvent those obstacles. Only if the learning mechanism is described in detail, can the obstacles associated with it be identified with enough precision to enable us to design instruction that circumvents them.

**CONCLUSION**

Theory-based designs for computer-based or computer-assisted classroom instruction can be generated quite easily. A stringent test of any instructional design is that it can be implemented on a computer. If the instructional design is clearly stated, then it should be programmable. If it is based on a learning theory and that theory is accurate, then the design should be sufficient to produce learning. Furthermore, strict empirical testing of an instructional design requires that the instruction delivered during the test consistently and precisely adheres to the prescription of that design, a requirement that human teachers might find hard to fulfill, but which is guaranteed to be fulfilled if the instruction is delivered by or with the assistance of a computer. For these reasons, there is a close connection between theory-based design of instruction and the implementation of reliable computer-based and computer-assisted learning programs.
REFERENCES

EKIDS - The new discussion list for K-12 school children

BY CRES THURSBY-PELHAM
Lecturer in Computing Studies, Graduate School of Education, University of Western Australia.
Email: cres@uniwa.uwa.edu.au

INTRODUCTION
The internet is a wonderful way to communicate, but most of this communication goes on between adults who often use the various email discussion lists which are maintained by the list owners of listserv, listproc, majordomo and other list handlers. A new list called EKIDS has recently been created especially for school children from K-12. EKIDS — an acronym for Electronic Kids Internet Discussion Server was announced on Australia Day 26th January 1995. The list is run from the City Beach Senior High School computing centre in Western Australia. City Beach Senior High School is the first school in Western Australia to have become fully affiliated to AARNet and hence to have full internet access available to students and teachers at the school. In fact, the school is believed to be the first in Australia to have developed its own WWW server site which became operational on 15th September 1994. Cres Thursby-Pelham is the part-time computing coordinator at the school and has devoted most of his spare time during the past nine months to setting up, maintaining and developing this unique school internet site. He also lectures in the Graduate School of Education at the University of Western Australia where his research interests are in educational use of the internet in schools.

AN AUSTRALIAN FIRST
This is the first list of its kind for school children in Australia and is believed to be the first open and unmoderated list for K-12 children globally. The list is in fact open to anybody who may be interested in K-12 education including teachers, parents, friends and members of the community. However, the discussion should not be dominated by adult talk and the kids should feel confident that they have a lot to offer their 'friends'.

AUSTRALIAN EDUCATIONAL COMPUTING, MAY 1995