education's goals and methods can only be met by an activist research paradigm which simultaneously creates and studies changes in processes and outcomes of human learning with new cognitive and technological tools.

PART II

ACQUIRING COMPUTER LITERACY
This chapter addresses the general question of what we expect to gain from the introduction of educational computing into the curriculum. In considering alternative instructional uses of this technology we will discuss both current practice and future possibilities, and reflect on how educational philosophies and our perceptions of the human/computer relationship can affect the uses made of hardware and software.

As noted previously, the effects of introducing computers into the classroom are neither predictable nor controllable. The tools which manifest themselves in computing are extremely versatile and can support many educational philosophies and objectives. Educators must reflect actively upon which form of education they are aiming for for their students. What is certain, however, is that the computer literacy requirements for the average person will expand dramatically during the 1990s. The least we should aim for might be to somehow provide everyone with the minimal amount of computer knowledge that would enable them to become computer comfortable. In this context the meaning of 'computer comfortable' is to allow people to be able to interact easily and without fear with a computer at a level appropriate to individual needs.

**EDUCATIONAL GOALS**

Computer software can be described along a continuum of open-endedness. At one end are the single purpose programs which fulfill only one kind of demand, e.g. worksheets, drill and practice programs and many simulations. At the other end are the open-ended tools such as word processors and programming languages, with which any number of different outcomes can be achieved. Another way of describing this continuum is in terms of the purposes of computing, i.e. for simple training and practice or for use as a tool. The training/practice software aims at a known goal or a well defined product, whereas the open-ended computing tools are more like pencils, paint brushes, hammers, chisels, etc., in that they are not designed for the achievement of a particular product or goal. Rather, they are tools with which many different and personal goals can be accomplished, and which can be adapted for many purposes. Training/practice software leads to clearly identified correct uses and
answers, which reflect the acquired knowledge and skills of the user. At the other end of the continuum correctness depends upon what use is made of the tool and on the capabilities of the user.

Different educational goals are inherent in the use of these different types of software. Teachers who are content to use computers as teaching machines, i.e. as providers of automated practice and testing, tend to be identifying themselves with the content-oriented curriculum, and with the view that the aim of learning and teaching is primarily a matter of students' acquiring facts. This approach will lead to the selection of programs which will help students know the facts that teachers and society consider worth knowing. There is strong support for this view in certain influential educational circles, but there are dangers in concentrating only on what? at the expense of how?, i.e. knowing what to do with the facts one has acquired.

There is little merit in educating children to become walking encyclopedias. Rather, our society needs problem solvers who have access to both the information relevant to a problem and the strategies for solving it. Computers can remember (i.e. store and access) facts better than people can, so why not rely upon computerised databases as sources of information and use people's energies for problem solving? More than any other educational innovation, the personal computer is useful for both these purposes. It can store and assimilate, in different ways, vast numbers of facts and rules and it can assist in the development of flexibility of thought. Obviously, to make use of a tool one has to know what to do with it as well as how to do it. The ideal philosophy for learning and teaching with computers is probably somewhere between the two extremes. The computer is capable of stimulating and supporting a great variety of educational goals. What is important is that, before educational goals are established for particular uses of computers, and before software is selected, educators clarify for themselves their own educational philosophies. The choices educators make about the use of computers in their classrooms may have profound effects, not only on the cognitive development of children, but also on the nature of education itself.

GETTING STARTED WITH COMPUTING POLICY

Every school and every classroom needs a written statement, no matter how short or tentative, of what role it intends computing to play in its curriculum. Such a statement should probably be part of the school's long-range plan, but may differ from school to school and within a school, and it will evolve with time. By addressing the following questions, a school can begin to come to terms with how it will view the role of computing:

Is computer literacy important enough to have a place in the curriculum?
Is computer literacy necessary for all students in this school, for selected ability groups or for students in certain subject areas?
Which computer skills must students exhibit in order to be regarded as being able to use the computer as a tool?
Should the ability to write computer programs and to debug computer programs be considered a component of computer literacy?
If the answer to question 4 is yes, which computer language should be used?
Which computer related social and political issues should students be able to analyse and evaluate?
Should computer training for computer literacy constitute a separate area of learning, or should it be integrated into existing subject domains?
What knowledge do incoming students have of computers? How can the school's curricula take differences between students into consideration?
At what stage/level should students start to acquire skills in computing? If these skills are to be used as tools to support learning in other areas, they will need to be acquired early.
When and how will the plan to make computing part of the curriculum be implemented?
How will feedback be secured and evaluated?
When and how will learning and teaching of computing be re-evaluated on the basis of student, teacher and institutional experience with curriculum reforms and in relation to the evolution of new software?

computing policy may be written separately or be written into each document in the different curriculum areas. Whichever option is chosen, the major concern must be that educational computing is perceived as something to assist reaching the goals set out in the school's total curriculum policy. What do we expect to gain from using computers in the school? Do we want students to learn about computers or with computers, or both? To some degree the answer to this question depends on how we define computer literacy.

COMPUTER LITERACY

Much depends on the definition of computer literacy. When an Education Ministry has decided to introduce computers into primary as well as secondary schools, it faces large bills for the hardware, software and other materials. In addition it faces the costs of curriculum development, of professional development and support for teachers, and of the assessment of student achievement. It is obvious that the way computer literacy is defined not only determines the contents of curricula and assessment, but also will have a profound effect on the tax-payer's willingness to support the use of computers in school.
Computer Literacy and Other Literacies

Concern has been expressed as to whether the focus on computer literacy might divert attention from or even conceal the need to concentrate on basic reading, writing and number skills. This concern is likely to be unfounded if computer literacy is viewed as an emerging component of the complex concept literacy as used more generally. There should be no incompatibility, because, as it develops, computer literacy will provide additional support and reinforce the need for the more traditional types of literacy. Indeed, in an information or knowledge society the ability to read, write and calculate will become more and not less important. The introduction of computer competencies into a curriculum cannot supplant the need for the more traditional literacies.

A report commissioned by the Club of Rome (Botkin, Elmandjra & Malitza, 1979) called for global educational reform to foster what it called innovative or anticipatory learning. Such learning is designed to prepare individuals to be able to consider possible contingencies and the long-range implications of various choices. Among learning activities particularly conducive to anticipatory learning are forecasting, simulations and modelling. Their use is made easy and inexpensive by low-cost personal computers. Such techniques emphasise the future orientation of innovative learning and the need for individuals to develop the ability to reflect on the implications of different decisions and to evaluate alternative futures. Anticipatory learning skills should probably become part of the basic skills of our students in the 1990s and beyond. These considerations should strengthen our efforts to bring about curriculum reforms which address computer literacy. They also help link the various dimensions of computer literacy as they develop and grow.

There is an established field of research which compares preliterate and literate school children to assess the postulate that literacy is a prerequisite of logical reasoning. It has been argued (e.g. Olson, 1978) that literacy allows individuals to master the logical functions of language and to separate them from the interpersonal functions of language. Language literacy assists cultures towards the development of formal reasoning systems. Will computer literacy manifest itself in a similar way?

Olson (1988) has argued that computers play three essential roles in students’ learning:

- they can provide rich databases which can be used by individuals as sources of information and for the construction of personal knowledge;
- they permit individuals to organise knowledge in a new way; and
- through interaction with peers about goals, successes and problems in specific computing situations, computing allows for a greater understanding of one’s own information processing and problem solving processes, and those of others. It develops metacognitive knowledge and skills.

Computer literacy is of practical importance for all members of our community, young and old alike. For example, all adults need to know enough about computing situations so as not to be intimidated by an error on a bill resulting from a computerised accounting system. Individuals need sufficient knowledge of how to use computers to be able to decide whether to acquire a machine for home or work, and they need to know how to evaluate when computer applications are helpful and when not. Some of this knowledge can be acquired as a by-product of programming and other computer uses, but most of this type of useful information will not be learnt in this way. Indeed, one might argue that most of what the ordinary citizen needs to know about computers might not be learnt from hands-on computing experience.

Defining Computer Literacy

ERDU’s NEWTECH/HITECH/INFTECH Glossary (Queensland Institute of Technology, 1985), one of the rare dictionaries, glossaries or encyclopedias containing the concept of computer literacy, provides the following comment on the term:

An ill-defined term concerned with computer familiarity. Problems arise in consideration of exactly what skills and knowledge should be possessed by people who, even though they may not actually work with computers, will be living their daily lives in a society dominated by the use of electronic data processing. Opponents of the concept of computer literacy draw analogies with, for example, the automobile: they argue that one may derive all the normal benefits of personal transportation without any knowledge about how it is being done (what is under the hood/bonnet). However, it is not unreasonable to suggest that anyone who draws such simplistic analogies is manifestly overlooking the need for computer literacy. (p. 16)

It is not surprising that the term computer literacy shares the semantic ambiguity of language literacy. The meaning of the latter term is restricted by some to reflect the acquisition of simple reading and writing skills (a narrow definition); others understand literacy to be far more than the acquisition of such basic skills.

Literacy is not the simple ability to read and write: but by possessing and performing these skills we exercise socially approved and approvable talents; in other words, literacy is a socially constructed phenomenon. (Cook-Gumper, 1986, p. 1)

Researchers at the US Literacy Institute have suggested that there are a multitude of literacies, each of which is an ‘integration of ways of thinking, talking, interacting and valuing, in addition to reading and writing’ (Education Development Center, 1988, p. 4). Each literacy is embedded in particular social settings, is shaped by children’s early experiences in the home and community environments, and is modified by different literacies encountered daily in and out of school. In short, in order for children to be successful at school and in society, they need to master a broad range of literacy competences (almost in the sense of...
being multilingual) in order to cope with the diversity they can expect to encounter in written and oral communications across a wide array of contents and contexts.

Widely accepted definitions of computer literacy can be classified into comprehensive and narrow definitions. Comprehensive definitions describe literacy in terms of the comprehensiveness of knowledge and skills which ordinary, educated people need to have in a particular domain (e.g., language, numeracy, health, economics, etc.) in order to function effectively at work and in their private lives in their culture or society for the remainder of this century. Obviously, a literate person can make use of a wider range of intellectual strategies than someone who is not literate. The following examples are of the comprehensive type, defining computer literacy as:

\[ \text{whatever understanding, skills and attitudes one needs to function effectively within a} \]
\[ \text{given social role that directly or indirectly involves computers. (Husen \& Postlethwaite,} \]
\[ \text{1985, p. 997)} \]

\[ \text{whatever a person needs to be able to do with computers and know about computers in} \]
\[ \text{order to function in an information-based society. (Hunter, 1983, p. 9)} \]

\[ \text{that comprehensiveness of knowledge and skills which ordinary educated people need to have} \]
\[ \text{about computers in order to function effectively at work and in their private lives.} \]
\[ \text{(Haigh, 1985, p. 161)} \]

Comprehensive views of computer literacy go beyond the narrow definition relating to a body of basically technical information and include knowledge of how computers work, how they are used, and their impact on society. Luehrman (1980), for example, believes that this body of subject matter should more appropriately be termed computer awareness. He regards computer literacy as a cultural phenomenon which includes the full range of skills, knowledge, understanding, values and relationships necessary to function effectively and comfortably as a member of a computer-based society. He stresses that the computer literacy needs of any one individual will thus vary according to that person’s particular involvement with computers. No single approach to computer literacy can serve all audiences and contexts.

To make this comprehensive notion of computer literacy more operational, Watt (1982) divided it into four distinct but interrelated components:

1. **The ability to control and program a computer to achieve a variety of personal, academic and professional goals:** This includes the ability to write programs in one or more computer languages, read, understand, and modify more complex computer programs, to use a computer as a problem solving tool, and to analyse information generated by a computer program (for example, predictions about economic trends or other futures).

2. **The ability to use a variety of pre-programmed computer applications in personal, academic and professional contexts:** This includes the ability to make informal judgments about the suitability of a particular software tool for a particular purpose, and to understand the assumptions, values, and limitations inherent in a particular piece of software.

3. **The ability to understand the growing economic, social, and psychological impact of computers on individuals, groups and society as a whole:** This includes the recognition that computer applications embody particular social values and have different kinds of impacts on individuals and different segments of society. It includes the understanding necessary to play a serious role in the political process by which large and small scale decisions about computer use are made, and to transcend the dependent roles of consumer or victim (cf. e.g., Turkle, 1984; Welzenbaum, 1976; Wessel, 1974).

Some computer literacy objectives can be integrated into mathematics, social studies, English, science, or other curricula. Others do not fit readily, for example the fundamentals of a computer language and computer programming. Computer literacy objectives which do not conveniently fit into existing disciplines can, however, be taught through specially produced modules of basic computing skills.

It would seem that a comprehensive definition of computer literacy addresses at least three sets of issues:

1. **using the computer as a tool**;
2. **determining the need for and, where appropriate, acquiring programming skills**; and
3. **assessing the personal and societal implications of pervasive computer use**.

As individuals and educational institutions grapple with the implications of computer literacy for the curriculum, each of these three issues requires attention. The weighting of them and details of curriculum contents will depend on the aims of specific courses, and whether computer issues and skills are introduced into a curriculum through separate courses or through modifying existing courses in traditional subject domains so that computing is totally integrated in the subject and becomes a major tool for learning.

The comprehensive view is that computer literacy is an understanding of computers and computing that enables one to evaluate computer applications as
well as to do computing if this is a personal need. This view of computer literacy fits with the long-established tradition of scientific literacy and related formulations such as technological literacy, geographic literacy, and economic literacy, to name only a few. Scientific literacy is generally defined as the knowledge about science which the lay person needs to function effectively. Scientific literacy refers not only to learning scientific facts but also to one’s understanding of the implications of science and science/society issues. Thus, it is not surprising that we often see computer literacy equated with computers in society and courses on the social role of computers.

The comprehensive view of computer literacy is also consistent with the recommendations of the US National Council of Teachers of Mathematics (1980). In An Agenda for Action they recommend that a computer literacy course, familiarising the student with the role and impact of the computer, should be a part of the general education of every student.

The narrow view is that computer literacy is simply a matter of using the computer for particular purposes, i.e. doing computing. The advocates of this view tend to define computer literacy as simply doing computing. They argue that the most basic components of literacy in a language are that the literate person has the ability to read and write, that is to do something with the language, not merely to recognise that language is composed of words, to identify a letter of the alphabet, or to be aware of the pervasive role of language in society. Literacy in mathematics means the ability to add numbers, solve equations, etc., i.e. to do mathematics, is not merely to recognise that numbers are written as sets of digits, to recognise a formula or to be aware of the advantages of being able to do mathematics.

By analogy, computer literacy must also mean the ability to do computing, and not merely to recognise, identify or be aware of alleged facts about computers and computing. Luehm1an (1981) refers to the latter facts as hearsay knowledge. This category of knowledge, i.e. hearsay knowledge, the lowest in Plato’s hierarchy, is essentially verbal. Its acquisition involves the student mainly in encoding information and remembering it when an appropriate stimulus is presented. It is qualitatively different from the knowledge that comes from experience, i.e. doing writing, mathematics, or computing.

The basic flaw in attempting to apply some of the objectives noted in the literature as a standard for what should be taught to achieve computer literacy is that, of the objectives provided, very few actually require the student to be able to do anything. For example, in 1979 the Minnesota Educational Computing Consortium (MECC) was awarded a large grant from the US National Science Foundation and came up with 63 objectives. Only 12 of these 63 objectives require the student to actually compute. Eight of the nine objectives in the Programming and Algorithms section fall into this category, along with three of the thirteen in Software and Data Processing. The other 51 objectives involve nothing more than student acceptance of generally held views or hearsay knowledge about computing, such as might be acquired from a book (cf. Johnson, Anderson, Hansen & Klassen, 1980).

Examples of items testing such generally held views about computing in the MECC (1979) list are:

Identify the five major components of a computer.
[Correct answer: input equipment, memory unit, control unit, arithmetic unit, output equipment.]
In addition to input and output equipment computers contain:
[Correct answer: memory units, control units, arithmetic units.]

Clearly, the student who has read and memorised the classical definition of a computer will score full marks on this item. Yet the problem is that months and years can pass in the life of a computer user or even a professional computer programmer without any need to remember, make use of or even reflect upon the fact that somewhere inside the machine lies an arithmetic unit and a few microns away on the same chip lies a control unit, and that they are, logically at least, distinct. A programmer could create an entire management information system or a data analysis package without ever calling on that piece of knowledge or putting it to use. Except for a few who work at or near the hardware level, people who do computing rarely use such general knowledge about computers.

An even greater concern about the use of objectives like those noted above comes out of direct experience of working with children and adults who are gaining hands-on experience in computing. After only a few hours of such experience, they know enough to score near the top on the handful of test items based on the dozen MECC objectives requiring that the student be able to do computing. Yet at the same time these students tend not to have any idea about what arithmetic units or control units are, and what the difference between them is. However, they do know about input, output and processing because they experience these things and have a need for words to communicate about them. To clarify the difference between this general knowledge and knowledge that comes out of practice, consider the different flavour of the following objectives (which do not relate to general knowledge):

Follow and give correct output for a simple algorithm.
Modify a simple algorithm to accommodate a new, but related task.
Develop an algorithm for solving a specific problem.
Design an elementary data structure for a given application.

Negative Aspects of the Narrow View

Narrow definitions of computer literacy suggest that hands-on computer experience and computer programming are the only important components of computer literacy. Those who promote this philosophy may unwittingly promote...
mindless or meaningless doing in addition to constructive experiences. Without adequate direction, students who are doing computing may actually be practising non-rigorous procedural thinking, acquiring misconceptions, and otherwise wasting valuable learning opportunities. The typical doing computing approach consists of teaching students to write a few programs, usually in BASIC. As Papert (1980) points out, this strategy often results in student learning that stifles creativity and suppresses motivation. It can also lead to awkward and poor algorithmic thinking. Hands-on computer experience does not guarantee computer literacy.

Obviously, those who argue for the narrow view neglect the semantic ancestry of the concept of literacy and preclude a broader understanding of computer technology. There are very good reasons for people to both communicate with computers and be knowledgeable about them. The solution lies in teaching computing not for its own sake but in providing students with constructive experiences in the use of computers to meet the requirements in all subject areas and in their personal lives.

In the brief time which is usually allocated to the running of a typical computer course, the student can learn little about problem solving and algorithms. It is also difficult to provide experiences with a variety of languages, and to build bridges that will allow for the adaptation of what has been learnt to other and real world domains. Students may end up with limited and inefficient strategies for problem solving, and if students only learn about a single computer and software system, they are in danger of developing a narrow conception of computing and have difficulty in transferring what they have learnt even to other computer environments.

**INSTRUCTIONAL OBJECTIVES**

The introduction of personal computers into schools has been justified mainly by two arguments. The first argument relates to the need for societal and vocational relevance of education. Computers play an increasingly important part in our everyday lives, and our children should be educated in their use and in the principles of their operation in preparation for their encounters with them in the workplace and elsewhere. Those who support this view tend to believe that the subject computing should be introduced into schools.

Although the parental lobby strongly supports it, this view of educational computing is limited. It regards hands-on keyboard experience as all-important, stresses the amplification aspects of computers, and also educational uses for drill and practice, but largely disregards the opportunities computers provide for the development of the computer user's improved power of thinking, problem solving and learning. These issues relate to the second argument.

The second argument revolves around the teaching of programming. The assertion is that experience in computer programming will result in new ways of thinking, e.g. that the student will improve his/her power of formal reasoning, will solve problems through heuristics and being placed in a position that encourages recognition of flaws within any suggested solutions will become more reflective and a better decision maker. This is a similar argument to that which has been put forward for the relevance of Latin by educationists in the past. Some argue that programming is the new Latin, but both programming and Latin will only be of use outside their own restricted domains if there is a transfer of skills to other domains of learning.

A review of the literature and discussions with teachers provides some information on how teachers view and use computers. Two distinct instructional uses of computers, which reflect the above arguments for the introduction of computing into schools, became evident: computing as a new subject domain and computing as an instructional tool for use in existing subjects. Obvious differences will result from these uses with respect to instructional goals, learning potential, demands on the teacher, pedagogy and curriculum implications. Contextual factors influencing classroom practice and teachers' personal theories of computers and computing are the major determinants of preferences for how computing should be integrated into the curriculum. The novelty and uncertain status of computing in schools may make it problematic for teachers, particularly those who are not volunteers for the teaching with computers, to decide which position to take.

Both positions will be presented in the next section. Many aspects of the curriculum are similar. Their stated educational goals appear not to differ. Both aim to achieve computer literacy in students, though their respective definitions of computer literacy and criteria for its assessment differ.

**Doing Computing**

Although there are no firm departmental or regional curriculum guidelines for the use of computers in primary schools, junior or middle secondary schools, there is a remarkable consensus among teachers and educational administrators as to what doing computing as a subject means. Like any other school subject it has its own terminology, its own underlying theories, sequence and scope of concepts. The goal is computer awareness or literacy, terms which are sometimes but not always synonymous.

The rationale is that students need to be prepared for the society of the future, i.e. the technological age. The theory of those advocating teaching computing as a subject in schools is that exposure to the machine, software and peripherals increases comfort levels of students until a certain proficiency level (commonly a student's ability to use computer software without the teacher's assistance) is reached. The usual method is hands-on experience.

For the classroom teacher there is an inherent problem with the notion of computing as a subject: it is new and often there is no place for it in the...
timetable. This means that computing has to be run as a concurrent activity with other, more traditional school subjects. In fact, the teacher has to teach two things at once. Doing several things at once may not be new (teachers frequently run multiple activity groups), but lack of familiarity with computers and the fact that there is usually little relationship between what different students or groups of students are doing on their computers, and/or the rest of the class, raises questions of how this can best be managed.

Given the very loose definition of computer awareness and its accompanying rationale, any software that is crash-proof and user-friendly is suitable. This can include utility programs, drill and practice, tutorials, adventure games, word processing, etc. In most classrooms the only criterion is that the software should require minimum support from the teacher who has to give so much individual attention in a class where students learn with computers. This makes programming, including Logo, problematic (cf. Chapter 4). Although programming is often considered to be one of the advanced stages of awareness, it is not self-sustaining. It demands time and attention, and both are at a premium for the teacher in the classroom.

Strategies for teaching students to use computers under these circumstances usually include an initial teacher demonstration of aspects of the software on the computer, allowing students to teach themselves through trial and error, using supporting documentation, and peer tutoring using the few already computer literate students who are found in most classes.

It is time-consuming for teachers to locate and appraise suitable software and other computer-related curriculum materials, to keep up with the field and provide the outside-lessons tutoring and remediation required by some students. Keeping track of how well students are managing on the computer may also be difficult. Ways of monitoring and evaluating the development of students' computing skills are discussed in later chapters. The exact nature of computer-related interruptions also needs to be clarified. The teacher must be in a position to make immediate decisions about whether a situation can be left to resolve itself or whether it requires teacher intervention, whether the problem is mechanical or student-related, whether or not existing rules and procedures cover the situation, whether the situation can be quickly resolved or whether it will require extra time and effort. At issue is the teacher's sense of control in the classroom. Most teachers report that they alter the ways in which they organise thinking skills, and that they do not know how to measure such developments should they occur, but they want to continue teaching with computers in the classroom.

Currently, teachers are trying to cope with computers in the classroom against a background of uncertainty. They would like to see departmental guidelines to classroom. Most teachers report that they alter the ways in which they organise details.

Given the present uncertainty about the place of computers in school, the lack of teacher training and the scarcity of resources, the decision to teach computing as a subject may be inevitable for many schools. Teachers who have taught computing as a subject are pointing out the problems and limitations of this kind of activity in which computer awareness is an end in itself. A certain level of computer awareness is necessary if the computer is to be used as an instructional aid, but if we wish to develop the instructional potential of computers in the classroom we need to look beyond minimal competency or awareness as the major goal. The instrumental or tool potential of computing should be addressed early, when computers are introduced, so that the computer can become a resource for students and teacher rather than an additional obligatory demand in an already crowded curriculum.

In defining computer literacy it is useful to distinguish it from computer science. Computer literacy is not the same as knowledge of computer science. A succinct distinction between the two can be made by suggesting that computer literacy is part of computer science which everyone should know about. As noted above, those taking the comprehensive view commonly define literacies (e.g. language literacy, mathematical literacy, science literacy) in terms of the layperson's perceived needs. In the way computer literacy should be thought of as the knowledge, understanding and skills the average citizen needs to have with respect to computers. This implies that students should be taught more than simply how to operate or program a machine. They also need to know how computers can be used productively and what the major consequences of computerisation are for society. This is why it is so important to encourage computer use in all subjects.

According to the educational computing literature, the objectives of computer literacy curricula range from those aiming to raise general awareness of computing through skilful usage to the possession of broader understandings of the personal, educational, social, economic and political contexts and consequences of that technology in society. The type and amount of knowledge one has about computers determines the potential of that knowledge to be personally, socially and politically empowering.

Computer Awareness

This objective aims to provide the student with some general information about computer hardware, software, vocabulary, uses, history, social, economic and political impact. Students may or may not actually see or use a computer, but if they do, the use is usually limited to demonstrations. Teachers and other instructors rely on a variety of sources of information to create awareness about computing, such as books, films and videos. Technical information and noncontroversial commercial applications are emphasised. Much of the
information which has been found useful in the classroom is produced and provided by the computer industry.

Initially, teaching computing was for the most part teaching computer awareness. This may have been appropriate at the stage when the technology was novel and only few machines were available to be shared by large numbers of students. Many writers concerned with the needs for curriculum aims in computing started off with surveys of what had been taught in the late 1970s and early 1980s. The courses tended mainly to convey information about computers and computing, with little opportunity for hands-on experience. This state of affairs certainly explains the reaction of Luehman and others emphasising the need for a hands-on approach. Even now most of the published educational objectives relate to computer awareness. Computer awareness is not the same as computer literacy, though it might be regarded as an aspect of it.

The reasons for continuing to teach computer awareness are far more compelling than the more formal rationale on which its introduction tends to be based. These reasons often relate to the desire to make the classroom a more interesting and enjoyable place to be in, and for developing new relationships through what is often a shared learning experience between peers, or between teachers and students. The building of these new social relationships in the classroom is often regarded as the reward for the time and effort invested in teaching with computers. Teaching computing thus takes on an affective, self-concept building as well as instrumental value. It makes a statement to students, peers, parents, educational administrators and others about the kinds of instructors their teachers wish to be, the kind of relationship they wish to have with their students, the classroom atmosphere they wish to establish, and their interests. At the classroom level the specification of instructional objectives will certainly be influenced by the decision made with respect to teaching computing as a subject or teaching computing as a tool to be integrated into all areas of learning.

**Ability to Use Computers**

Another objective of computer literacy emphasises the ability to use computers, i.e. programming and applications dominate the curriculum. The hands-on approach is stressed, because the aim is to train students to control the computer. Programming, word processing and spreadsheets are introduced in the elementary grades and more systematically elaborated in the middle school and high school mathematics and business education departments.

The hands-on approach to teaching computing can enable students to develop considerable computing skills, including the ability to use graphics and word processing. Students become confident in their interaction with the computer. But demystifying what is generally a user-friendly personal computer contributes little to understanding the importance and potential of the technology for the individual. In fact, as noted in Chapter 1, a narrow focus on the technical skills of using computers may lead the students to a false sense of empowerment.

The technical focus shifts attention away from social questions and portrays computers as something to learn rather than as something to think about. The computer is portrayed as friendly and accessible, and the user is encouraged to think that all computers, even those in large systems, are friendly and accessible. In this manner, computers are further mystified in the very act of demystification. (Noble, 1985, p. 72)

**Making the Computer Part of Oneself**

A third type of instructional objective is one which aims to have students make the computer part of themselves, in their school and personal work as well as social environments. The attainment of this objective is potentially most empowering, yet it appears to be the least discussed in curriculum documents and least experienced in schools. It introduces computing skills and knowledge within a broader social context, stresses the implications of computer technology and the empowering effects of such knowledge. In addition to some technical knowledge, students need understanding and knowledge of who controls the direction of computing, for what purposes, for whose benefit and whose loss.

One might argue that it is not easy, and in certain cases intellectually improper, to inculcate beliefs and values about a subject which do not arise from the student's direct experience with the content of that subject. If one were writing about mathematics, reading or writing, there would be little disagreement about this point. For example, parents would be properly outraged if their children were asked to spend four out of five days working on the beliefs and values about the subject of mathematics, and to spend only one day on learning to do mathematics. However much we want our students to remember facts about mathematics, and feel good about it, we know that these beliefs and values will be short-lived if the students go out into the world with poor ability to do mathematics. The same applies to all other areas of the curriculum, including computing. In the future, most members of our community will have very real practical needs for understanding computers.

Computer literacy will become as important as literacy in language and mathematics. Like reading, writing and arithmetic, computing gives the student a basic intellectual toolbox with innumerable areas of application. Each one of these tools gives the student a distinctive means of thinking about and representing a task, of writing his/her thoughts down, of studying and criticising the thoughts of others, of rethinking and revising ideas, whether they are embodied in a paragraph of English, a set of mathematical equations, the simulation of a social process or the development of a computer program. Students need practice and instruction in all these basic modes of expressing and communicating ideas. Mere awareness of these modes is not enough.
General Objectives

As yet there are no generally agreed upon curriculum objectives for learning with computers. As noted previously, the US literature contains various lists of items based on objectives for computer literacy (e.g. MECC, 1979; Johnson, Anderson, Hansen & Klassen, 1980; Luehrman, 1981), often without any specification of what the specifics of instruction actually are. Most of the objectives to be found in the published literature are representative of the lower levels of cognitive skills. Objectives covering deeper levels of knowledge and understanding are not sufficiently developed, hence ideas of minimum competency cannot be based on them.

The US National Commission on Excellence in Education (1983) recommended that high school students should complete, among other things, one half year of computer science. The Commission suggested that, as a result of taking such a course, high school students should be able to:

Understand the computer as an information, computation and communication device;
Use the computer in the study of other basics and for personal and work related purposes; and
Understand the world of computers, electronics and related technologies.

Some aspects of computer literacy might be achieved by requiring students to take an introductory computer course and/or programming courses. For general educational purposes it might be better to begin with the broader curriculum aiming at general literacy. Such a curriculum will put primary emphasis on the direct interaction between the student and the technology, with a goal to help students not just to cope with the hardware and software but to master wholly new analytic, expressive and problem solving skills.

The comprehensive approach to computer literacy requires that many domains be included in a curriculum dealing with computing. Not all of them will be given equal weight by all teachers -- rather, the teacher will pick and choose appropriate goals for particular students.

At the present time and in most contexts, any list of curriculum objectives will be an evolving conceptual structure. Initial lists may be a smorgasbord of objectives for computer literacy which requires ongoing refinement and updating. In any case, the objectives will require ongoing revision to take account of changes in the technology itself. Whatever list is constructed, it must provide guidance not only to curriculum but to those who wish to assess progress. More specific learning objectives and suggestions for assessment are provided in Chapter 5.

COMPUTER PROGRAMMING AND COMPUTER LITERACY

Programming

Programming is the process by which a computer is made to perform a particular task. Programming involves the creation of a formalised sequence of instructions which can be recognised and implemented by the machine. These instructions, i.e. the program, are in themselves a static entity, but when executed, they result in a useful means of information processing. All programs are concerned, either directly or indirectly, with the flow of information. Data, whether stated explicitly or made an intrinsic component of the program, are used as an input which is then processed or computed, to generate an output. All of the functions performed by a computer depend, at some stage, upon a program.

The instructions are encoded into a specific programming language. Different languages vary both in structure and in syntax. The choice of language depends on both the application and the computer for which it is intended.

Broadly speaking, the components of a computer program can be categorised into three kinds:

1 Commands which are responsible for the manipulation of data within a system. They perform what is referred to as the actual computation and include reading in values from the external environment, assigning values to variables and sending data to the output device.
2 Commands governing the flow of control through a program. While their syntax may vary, virtually all programming languages embody four basic structures: sequential commands or statements which are executed consecutively and once only, conditions which are used to select between different sets of commands, depending upon the parameters specified within them and which can alter the sequence of execution, and the repetition of commands. Frequently groups of commands need to be performed many times during the execution of a program. This is achieved by creating a loop within the program which retains control of the machine until a certain condition has been fulfilled.
3 Procedures or sub-routines. These are sets of commands forming part of a program which may be used more than once. A procedure is called into action by using its name, and variable values can be used to show to what the procedure should be applied.

The Value of Learning to Program

As noted previously, an essential question in operationally defining computer literacy is whether one must know how to program in order to be computer literate. For many people the answer may be no. However, this does not dispose
of questions relating to the value of programming as far as general education is concerned. A growing body of opinion suggests that there are substantial intellectual benefits to be derived from learning computer programming. Such basic concepts as iteration (i.e. a process which repeats the same series of processing steps, e.g. repeated application of a self-contained routine) usually programmed as a loop, recursion (i.e. the action of a routine calling itself or being called by another routine, sometimes one that has itself been called... until a predetermined value is reached), and similar systematic procedures are more difficult to introduce to students outside the programming environment.

Problem solving skills are vital to everyone, and schooling at all levels should improve the ability of students to solve problems. Learning to program a computer in a suitable language can develop problem solving skills. Properly done, developing a program is a process of defining a problem so that it can be broken down into discrete components, none of which is too difficult to handle, even though the entire program may be quite complex. Computer programming also introduces students to notions of complexity, interconnectedness, uncertainty and the dynamics of a problem space.

Another useful intellectual by-product of learning to program is developing a technique for debugging, i.e. for detecting and correcting errors in a program solution or program. Because of the nature of the process, learning to program involves making many mistakes and learning to diagnose and correct them. Learning to program thus encourages the development of error detection and correction techniques, and even more importantly, develops what some call a no fault approach to making errors. In our assessment conscious educational environment, individuals are usually embarrassed by errors and often attempt to avoid thinking about them. This is unfortunate because it is important that we develop the ability to learn from our errors. Computer programming is an activity which makes a strong positive contribution towards the development of this ability.

Unfortunately, many of the general educational by-products of learning to program are still very much hypothetical. Although many people who have learnt to program can testify to the value of computer programming, systematic research to rigorously evaluate these hypotheses is still lacking. There is, however, considerable support for the view that children who are learning to program establish a different kind of task oriented pattern in their personal problem solving endeavours and in the interaction with their peers when they work together in solving programming problems. Research has shown that these patterns of cognitive processing can carry over into other classes, and suggests that dialogue about problem solving and learning processes can transfer to other areas of learning. This is likely to have significant educational value. Indeed, it may be that it is in communication, rather than (as often presumed) in mathematics, that the computer may eventually make its most important contribution. The connection between computing and communication is very strong. More than a decade ago Seymour Papert suggested that the term computer "science" is a misnomer because "most of it is not about the science of computers, but the science of descriptions and descriptive languages" (Papert, 1980, p. 100).

Choice of Language

Assuming that learning to program has educational value, what languages should we teach? Many would suggest BASIC. However, BASIC is not really useful for complex programs, nor does it have the characteristics that help develop problem solving skills. Indeed, the convoluted logic usually required to work with BASIC requires users to accommodate the requirements of machines and, in doing so, might discourage the development of desired problem solving skills.

BASIC has become popular for two reasons: it does not require the user to learn a large vocabulary of computer terms and it runs on computers with limited amounts of memory. Because the memory capacity of personal computers is increasing substantially without much additional cost, limited memory no longer serves as a limitation. Furthermore, the greater power of other languages rewards us for mastering their somewhat larger vocabularies. If not BASIC, what language then? There are several candidates, among them Pascal, C, LISP, and Logo. Of these four, Logo might be the best option for schools. Logo dialects are now available for a wide number of personal computers.

The strongest assets of Logo might be that it was actually created in an environment of Piagetian developmental psychology and that it functions as an interpreter rather than as a compiler (i.e. it gives the user immediate feedback, as BASIC does, in contrast to Pascal, which is more complicated to use). Logo is sometimes mistakenly described as a language for children, because it has been used successfully with children. However, it is a fully developed language and a powerful tool. It is discussed further in Chapter 4.

Interactive Software

Although the computer is not always thought of as a language tool, it might actually be one of the most useful recent innovations for developing language. Using interactive software enables students to discuss, hypothesise, predict, debate, test ideas and develop thinking skills in a medium which they find highly motivating.

A few years ago one would have suggested that anyone wishing to use the computer as a tool would have to learn to program, but this may not be true any longer. Today a variety of software packages exist, including hypercard, which are powerful, easy to use, require no programming skill and run on small inexpensive laptop and desktop computers. These include adventure games, simulations, word processing, spreadsheets, graphics packages and other information retrieval and exchange devices.
Adventure Games  These are computer programs which are, in some ways, like fiction books. However, an adventure game is planned in such a way that it enables the user to interact with the story. There are many variations of games which differ not only in complexity but in degree of difficulty. Adventure games can basically be categorised into three difficulty levels:

Highly structured games: Here the students have a limited number of choices to make at different stages of the game. Usually quite young children can operate the game because the goals are basic with limited options to choose from. These programs are suited to the lower primary level but are also useful for older students who have had no experience with adventure games and/or computers. Because a limited number of choices are required, these games do not lead to much discussion among students, or to critical thinking.

Partly structured games: The students are able to use a wider range or more open-ended options to reach their solutions to various problems in the game. They are suitable for middle and upper primary grades. These games are more open-ended thus offer more opportunity for discussion than the highly structured games. They require the users to be more self-reliant.

Unstructured games: In these games the students themselves are required to work out all aspects of the game. For example, in some games they may be required to work out the rules and objectives, or how to solve the puzzles without added information. Therefore, this type of game offers an excellent opportunity for promoting discussion, especially for the older, more experienced student.

Generally, the game provides the user with a scenario and then offers a choice of actions. The resultant choice then sets the user on a course with many further choices. More sophisticated games do not even offer choices but rely on the user providing all the input.

Simulations  A simulation is a computer program which simulates a real life problem or situation. It enables the user to interact in various circumstances throughout the program. The user's actions have an impact on what occurs during the simulation. These programs enable students to explore real world situations which may be impossible to explore in any other way. Obviously, such programs should not replace excursions, visits to museums and to industry, but be used in conjunction with them where possible. For example a simulation based on finding and feeding animals at the zoo could be used as preparation and follow up to a visit to the zoo.

Word Processing  Word processing is a term invented by IBM to refer to the process of creating and editing text electronically. Word processors enable the user to type text and then manipulate it. The ability to easily change text formats and styles makes this type of program attractive to all computer users.

Many people equate word processing with typing on an advanced electronic typewriter, but this is actually a poor metaphor. Both word processor and typewriter use a keyboard to enter information, but after that there are only superficial similarities between the two. The word processor allows the addition, deletion and movement of words, phrases and whole blocks of text. Because it is so easy to restructure text it is preferable to think of text processing rather than word processing. A word processor could be regarded as more analogous to a cassette recorder than to a typewriter because they share the facilities of fast forward and fast backward searches, easy editing, and cutting and pasting. Essentially, they are both tools to manipulate ideas, operating in two-dimensional space, whereas the typewriter works in a linear mode only. The word processor is a tool of considerable power and there are a number of valuable learning activities which can be performed even with the least sophisticated word processing package.

Because students in virtually any subject area or discipline need to learn to write, word processing skills can be of great value to nearly every student. Furthermore, word processing software can support the process of teaching English composition by simplifying the mechanical tasks of correcting errors and rearranging text, thus allowing students to concentrate on the more complex aspects of writing. Used in this way, word processing can effectively introduce students to the computer as a tool and can become a first step towards achieving wider computer literacy.

Spreadsheets  Spreadsheet software automates the process of producing reports with columns and rows of information. Almost any application which displays numeric information in a table, and needs row and column totals and/or percentages, may be prepared by spreadsheet software packages. Spreadsheet software is valuable particularly when such tables need to be updated regularly, because totals and percentages can be recalculated automatically. Budget tables which must be revised frequently are natural applications, as are decision making procedures using what if experiments. Issues relating to the value of this software as a mediator of cognitive development will be discussed later in this chapter.

Graphics Packages  We are living in a world which has become increasingly visually oriented. Diagrams and charts have become common forms for displaying information. In the past, the main limitation to using graphics was a lack of the necessary talent to draw them. Now there are a variety of graphics packages which will create pie charts, bar graphs and scatter plots quite easily on paper as well as on screen. They constitute a growing resource that can make it easy to create high quality graphics without programming skill and at minimal cost. This type of package can be useful in a large variety of domains and courses.
Information Retrieval and Exchange  The addition of a communication 
capability, such as a modem, to an inexpensive personal computer can transform 
it into a powerful device for the exchange and retrieval of information. It can be 
used to retrieve information from information banks available to subscribers, to 
search bibliographies, and to exchange messages such as letters, memos and 
readings, via telephone lines. Information exchanged in such a way can be used 
with word processing systems to provide a very powerful communication 
capacity. Being able to communicate electronically with students in other 
schools, even overseas, has become a reality for students in many Australian 
classrooms.

INTEGRATING COMPUTERS INTO THE EXISTING 
CURRICULUM

In most schools where computers are used, the computers are not themselves 
the primary object of study, but essentially a tool. This raises the question of whether 
there should actually be a curriculum for computing, or whether the curriculum 
objectives in content or subject areas should be amended to allow for computer-
based teaching and learning.

Because computers are regarded as tools rather than as a topic of study in 
themselves, teachers using computers in existing courses are sometimes reluctant 
to devote too much time to the development of students' computing skills, and 
consequently the result could be a cookbook, key-by-key approach to teaching 
the use of computers. Although this approach might, in the end, lead most 
students through the required processes step by step, it involves little more than 
repetition on the part of the students. Used in this way the computer is merely 
a medium of presentation, like a special workbook or overhead projector, though 
perhaps a little more dynamic.

Like all other learning, computing should be intellectually stimulating for the 
students. However, there is still the problem of how to accomplish this, in terms 
of both class time and the availability of computer literate teachers. One solution 
would be for Commonwealth or State Departments of Education to commission 
staff or outside consultants to prepare small modules that could be used in 
conjunction with existing curriculum subjects, thereby avoiding the necessity to 
introduce a special course in the technical aspects of computing.

Minimodule of Foundation Skills in Computing

A module aimed to provide a rapid and effective introduction to computer use in 
say, mathematics, science, social studies, writing, etc. would accomplish the 
basic prerequisite goals in educational computing. Students need to be able to 
work with a computer's operating system to perform file-management tasks, to 
use an editor successfully, and to cope with input and output devices. Teachers 
must not restrict themselves to the use of readily available software that avoids 
the use of these skills. A foundation set of skills needs to be presented in a 
manner that allows the ready transfer of what has been learnt among various 
courses which students might be expected to use in different contexts and in 
an assortment of subjects.

The ultimate goal is to make students self-sufficient in the way they use 
computing tools on a daily basis, equipped to solve problems and to learn how to 
work with new software and computers. The module needs to work for teachers 
who are not computer specialists. In addition, the module should impose a 
minimum of overhead in class time. This might involve some restricted time for 
independent study and practice. Above all, the introductory experience should be 
pleasant and inspire enthusiasm.

The Development of Reading Skills

Personal computers are an enabling technology. They help the user to collect, 
organise, store, retrieve and deliver information. One of the most commonly 
available information handling devices in the classroom is the word processor. As 
noted above, the word processor is more than a device to produce fancy printing. 
Word processing is one of the most open-ended and flexible tools with which 
students can think about the structure and purposes of language. That the word 
processor is a tool for writing is obvious, but it can also be used as a tool for 
developing reading skills. In fact, it can be used totally without writing on the 
part of the user.

One of the views currently held by teachers of reading is that children should 
develop writing skills as a way of developing their reading, in other words 
students should be encouraged to write to read. This statement may appear 
contradictory, but it is based on the sound learning principle of using the child's 
own knowledge as a starting basis for development, and for learning by doing. 
An adult who keys in a story which the child tells him/her, and then produces a 
print-out, produces reading material for the child which is personally relevant. 
Because the poor reader is reading materials in context, and is therefore more 
likely to be successful when the context has saliency, this personal relevance 
encourages the development of reading skills.

When employing this approach to the teaching of reading, the role of the adult 
is critical. If the adult takes an active part in the writing process then the whole 
social balance will have changed and the child defers to the authority and wishes 
of the adult, thus losing the all-important personal involvement and/or the 
motivation for story production. The adult helper is there as a medium through 
which the child achieves his/her story goal, not as a judge or assessor. 
Collaborative writing between adult and child may follow later, when the child 
has sufficient skills to feel confident of sharing the task with the adult.
The use of specific word processing features such as \texttt{FIND} and \texttt{FIND AND REPLACE} offer opportunities to practise \textit{reading for meaning} and for more general vocabulary development. These commands allow students to reconstruct the text.

The role of the word processor in developing \textit{information handling skills} and as an introduction to information handling packages may not be immediately obvious. At all levels of education, and in many work situations, there is a frequent need to summarise a text. One interesting reversal of this is to provide students with a summary of text, or with a piece containing only key content words, and asking the students to reconstruct the passage in their own words. The texts written by different students can then be compared or related to the original. The latter activities can encourage students to focus on the structure of text, and to consider what is or is not redundant. Active language games which require rethinking and editing as the writer goes backwards and forwards in the text provide useful experiences in the manipulation of language. The word processor allows for repeated revision without penalty. The non-linearity of such exercises benefits students' language facility, their ability to comprehend and create text, and to generally think more flexibly.

The creative teacher can thus use the word processor in many different ways: as an electronic worksheet for basic skills practice, for the development of advanced reading skills, for the decoding of text, to develop an understanding of story structure, and to encourage planning and self-monitoring. The word processor can support reading and writing, of course, but it is also an organiser of thought, a notebook, and a trial ground for the exploration of ideas.

The Development of Writing Skills

Writing with the computer is a valuable activity for individuals of all ages. Papert's description of an alienated writer who moved from total rejection of writing to intense involvement (accompanied by rapid improvement of quality) within a few weeks of beginning to write on a computer makes convincing reading (Papert, 1980, p. 30). Sharples (1985) goes further, offering an explanation of children's use of few of the high-level skills of editing and revision when writing with pencil and paper:

To be a writer is empowering yet every word that a child forms on paper is confirmation of inferiority. However carefully and neatly a child may write, the result is a poor substitute for adult typeface. If we want children to become adult writers, we should equip them with adult writing tools. (Sharples, 1985, p. 10)

This statement begs the question of how any child ever learnt to write, but it does make a useful point. Children tend to be involved in the technical details of text production, which leaves limited cognitive capacity for the higher level creative, organisational and other more demanding activities of writing. If our minds are trapped by decisions about spellings or grammar, they cannot be used for the consideration of story, plot or even idiom.

Using the word processor allows the writer to concentrate on the process of writing, i.e. the acts of drafting, editing and revising the text. Daitre (1985) suggests that writing on a word processor is more like talking than writing; in other words, it is an interactive exchange between writer and the tool for writing. Text is entered into the machine, which then displays the writer's ideas for consideration. If, on reflection, things do not look right, modifications can be made. The tool responds and a new, clean version of the student's thoughts appears on the screen.

Writing is a multilevel complex process, in which the student writer's attention tends to be directed to the lowest skill components which pose performance difficulties. The notion that novices attend to different aspects of performance in problem solving and other cognitive activities than experts is well recognised:

Writing is a communicative action that results from multiple cognitive processes that operate simultaneously, producing text through their interaction. For example, there are processes that draw a letter on paper, and those that select and organise ideas. While an expert writer can operate competently on these many levels, a novice tends to become locked into more local levels, a phenomenon called downsizing . . . (Levin, Boruta & Vasconcellos, 1983, p. 220)

One area of concern relates to transfer of training, the exchange of skills between the two media for writing, from pen to machine and back again. Salomon (1988) has shown that children who have been taught writing skills, such as redrafting of text using the word processor, do transfer those skills to pen and paper production. Our experiences in the SUNRISE classrooms at Coombabah support this finding. When asked to draft and write an essay using paper and pencil, the students proceeded in similar ways as they did in computing, and the products were certainly comparable with those of students who do not learn with computers.

Finally, it must be recognised that all technological innovations evolve over time to reflect the needs of users as well as new developments. Over time technological innovation is cumulative. One change does not always replace another. Rather, innovations overlap to form the complex combinations of old and new technologies so often encountered in modern society. Many existing technological innovations are under review to revise more suitable and improved applications. This state of flux requires constant monitoring and adaptation of curricula relating to the use of technology.
TECHNOLOGY AS A MEDIATOR OF COGNITIVE DEVELOPMENT

The theoretical framework for this endeavour stems from the work of Vygotsky, who first introduced the concept of mediated activity into the psychology of thought and language (Vygotsky, 1978). Unlike other approaches to mental functioning, Vygotsky's activity theory views cognitive and motivational processes as embedded within larger activity structures whose goals they serve (for expositions of activity theory see Kozulin, 1986; Wertsch, 1985). Activity structures involve mediators, i.e. tools and symbol systems which have deep implications for the way in which intellectual tasks are accomplished. Thus, this theory suggests that the introduction of new systems and tools into learning (or work activities) can be expected to change the intellectual aspects of these activities. According to the theory, however, the nature of these new intellectual demands cannot simply be projected from a study of the tools themselves. The demand characteristics of the tools are not all built into the tools themselves. Many of them stem from the way the new tools are utilised, i.e. the functional purposes they fulfill and the way tasks involving them are structured and socially distributed. A cognitive analysis of the impact of new technologies, therefore, must be concerned with the varieties of ways with which such technologies are drawn into ongoing activities.

A concrete example relating to literacy may help to illustrate the difference between more standard approaches to cognitive development and the activity approach. A writing system is readily recognised as a form of intellectual tool which mediates a multitude of social practices in a culture, including educational activities, work activities, recreational activities, and the like. Here is one technology that scholars have long agreed has cognitive implications. A long tradition within scholarly disciplines, and more recently in anthropology and psychology, has sought to derive these implications from studies of the properties of writing, such as the fact that writing objectifies language, is composed of units that are not marked off in speech, etc. (e.g. Goody, 1987). This school of thought has put forward claims that the intrinsic properties of writing systems, especially alphabetic scripts, promote abstract and logical thinking among those who master them. Literacy programs have often been built around the understanding that literacy both requires and fosters specialised higher order ways of thinking.

Empirical work, much of which was conducted within the Vygotskian framework (e.g. Scribner & Cole, 1981), disputes this postulate. Studies of literacy in various cultural and community settings demonstrate that there is no hard and fast relationship between literacy and cognitive implications. Intellectual implications of literacy are variable, and often contingent on the functions which are being served by writing. If literacy consisted only in rote memorisation of a sacred text, its intellectual consequences would appear to be limited to specific rote memorisation skills (e.g. Schieffelin & Gilmore, 1986).

On the other hand, literacy which serves multiple communicative purposes appears to foster skills in organising and expressing complex information in instructional situations, although it has little impact on memory skills.

The proposition that introducing a writing system into a society has a fixed set of cognitive consequences in all places at all times is an argument in the technological determinist vein. In contrast, the activity theory approach assigns a leading role to organisational structure of the society adopting writing (is writing the prerogative of a priestly class or available to people in many social groups?), specific practices in which writing is introduced as a mediator (is it confined to private uses or does it figure in trade, government, and everyday life?), the individuals who are recruited to literacy in the conduct of these practices (do all participants become literate or do some serve as representative scribes?), and the conditions under which they use it (is text easily produced?).

Since in the modern world there is considerable communality in the functions of literacy across various societies, we might expect such communality to be reflected in like cognitive correlates. It would be misleading, however, to argue backward from discovered similarities in the consequences of literacy to conclude that these are all inherent in the properties of the writing systems. The properties of writing systems have certain potential effects on social and psychological processes, but the realisation of those effects in turn depends on existing, historically created social and psychological factors. The relationship is reciprocal and not one way.

The above framework, illustrated with respect to traditional literacy tools, can guide our approach to new technological systems more generally and to learning with computers in particular. The general message is that the unit of analysis for cognitive studies of computing cannot be restricted to the technology itself, nor to isolated tasks removed from the context of their performance. Such analyses would provide only partial and possibly misleading information for policy makers, curriculum developers and teachers who are concerned with defining educational goals for the future.

Research conducted overseas and in Australia into the effects of the computer on students' cognitive development has too often tended to regard the computer as a single factor of change introduced into a classroom, which is presumed otherwise to remain the same. In other words, the computer is perceived as an independent variable the net effects of which can be controlled and quantified. In reality, there are no net effects. The introduction of computers into a classroom is far more than a treatment. The characteristics and potentialities of the computer become inextricably intertwined, not only with the way students might go about learning and problem solving tasks, but with the tasks themselves and the whole...
context of learning and teaching. It is not the features inherent in the tool but how students and teachers use it that determines the effects of computers in education.

Software Can Restructure Cognitive Processes

How can computer-based technologies fundamentally restructure the way humans think? Three examples will be discussed in some detail, and others mentioned, in which software has qualitatively changed both the content and flow of the cognitive processes engaged in human problem solving. In particular, what the student does and when he/she does it -- in other words, the component mental operations that a person contributes to the computer-aided problem solving efforts -- can undergo substantial change in comparison with the operation of these processes in traditional problem solving environments.

Electronic Spreadsheets A first example of software programs for personal computers which can restructure, and not merely supplement and thus boost, mental functioning is the electronic spreadsheet. The screen image of an electronic spreadsheet physically resembles a ledger sheet, with cells organised in rows and columns. But the resemblance ends there. In an electronic spreadsheet, one can place a numeral, a calculation, or a formula in the formula area of any cell, which can subsequently be edited, copied or moved. The results of calculations in the formula area appear as the content of the cell. The most dramatic difference between electronic spreadsheets and static paper spreadsheets is that one can change cell entries and see the repercussions of that change recalculated immediately throughout the total spreadsheet. Many lines of thought can be simultaneously activated in the form of dynamic living plans, and their outcome compared in terms of crucial variables. This what if property has dramatic consequences for the cognitive activities of, for example, budgeting (and financial modelling) and other forms of hypothesis testing.

Before 1979, in ledger sheets representing financial quantities, formulas relating these quantities and change over time were either recalculated by hand after every change, or modelled with mainframe programs under the control of data processing departments. Executives responsible for financial planning were not directly and personally involved in these operations. Personal computer budgeting has become a highly creative means of generating and testing various scenarios in complex financial situations for what could be, if property has implications for how student users of such programs think mathematically? A student using such a program is likely to spend time primarily on algorithm design and search (i.e. solution path finding) of appropriate operators, rather than engaged in the mechanics for calculating numerical expressions. Consider the task of solving linear equations. Search is not a central concept in algebra instruction today, but a central insight of cognitive science is that hypothesis testing by means of interactive development and testing of different models for budgets. The temporal sequencing of mental operations in the functional system for budgetary thinking has also changed: now the planner can opportunistically and flexibly test hypotheses in the model virtually wherever and whenever he/she wants. For example, any hypothesis on relationships between cells can be tested by modifying formulas and observing the recalculated results.

Beyond the quantitative increase in efficiency (some estimate saving ratios in budgeting to be 80:1) business planners now run vast numbers of complex experiments of hypothesis comparison and they can include many more variables than they could in the past. They also have a better understanding of the interdependencies of the component operations than before this electronic tool was available.

Furthermore, this tool has qualitatively changed the organisation of budgetary justification and argumentation. Electronic spreadsheets are now commonly used, unlike anything before, to quantitatively justify business decisions in group discussions by on-line comparisons with alternatives. The dynamic what if capacities of such a system make it possible to display immediately the consequences of different approaches to a problem that may be suggested during a board meeting of a company.

Finally, at the institutional level, the personal computer electronic spreadsheet has decentralised financial planning. Everyone can play with it. The number of mediating links between planning and testing financial models has been reduced rather than increased by the technology, and executives report feeling more in control of their futures.

Problem Solving in Mathematics Similarly to the spreadsheet users, mathematics educators argue that the use of symbolic manipulation programs such as muMATHS, MACSYMA, REDUCE, MILO and MATHEMATICA for doing algebra leads to a profound shift in the functions and structure of mathematical thinking from mechanical operations to problem solving operations (e.g. Arnold, 1991; Conference Board of Mathematical Sciences, 1983; Fey, 1984; Goldenberg, 1988; Grace & Cassidy, 1990; Heid, 1988, 1989; Maurer, 1984a, 1984b; National Science Board, 1983). These personal computer programs and others can easily accomplish the solving of complex numerical and algebraic equations, factoring of polynomial expressions, evaluation of definite and indefinite integrals, differentiation of elementary functions, solution of equation systems, and simplification of equations, even those with radicals (Kunkle & Burch, 1984; Wilf, 1982). What are the implications for how student users of such programs think mathematically? A student using such a program is likely to spend time primarily on algorithm design and search (i.e. solution path finding) of appropriate operators, rather than engaged in the mechanics for calculating numerical expressions.
The pedagogical flaw in this method is that students do not know when to select most classroom instruction in solving algebra equations, the teacher selects the arithmetic calculations. Students can select the operator and its scope of can also back up a solution path by applying the inverse of a forward operator to solve the equation \(5(2+x)=20\) for \(x\). A series of windows provide the student evaluation.

Operators are also provided for exploring solution paths. There is an \texttt{UNDO} operator which returns the equation to its immediately preceding state and a \texttt{GOTO} operator (not on the menu) which returns to any previous state. Students can also back up a solution path by applying the inverse of a forward operator (e.g. selecting \texttt{DIVIDE} after they have just applied \texttt{MULTIPLY}).

Because the windows show every operator used and every state the equation was transformed into, students have valuable opportunities to learn from specific paths of their problem solving, and they can play with possibilities. They can explore the search paths of their solution space, examine branch points on one stem where an operator was used which led down an unsuccessful path, and on another stem try an operation started down a path toward solution.

The above described learning activities are not possible with traditional methods for learning to solve equations. AlgebraLand and similar software offer new opportunities for different forms and types of learning through problem solving which were not available in static, non-computer-based symbolic technologies for solving equations.

In summary, these types of computer environments emphasize a procedure that is diametrically opposed to the traditional instructional methods. Using this type of software the student chooses when to apply operators, and the computer carries out the mechanical procedures to transform the equation. Students are thus challenged by the problem of search for and discovery of a path of operations that will lead from the initial problem state to the goal of solving for the unknown \(x\).

Learning effective search skills in algebra equation solving is not a trivial task. The cognitive technology of the type of AlgebraLand reorganizes the learning in a way that appears to highlight more fundamental skills to be learned. It introduces students to the functional system of mathematical thinking for the equation solving task. Similar reorientations are evident in artificial intelligence tutors in the programming language LISP (e.g. Anderson & Reiser, 1985) and geometry proofs (e.g. Boyle & Anderson, 1984). The required component operations are redirected. Calculation of arithmetical operations is eliminated, but students can now analyze and learn from an explicit written history of their problem solving moves in searching for the path of operators. This type of software with its focus on problem solving strategies (as the crucial human component in equation solving, finding geometric proofs, etc.) thus provides students with the opportunity to become familiar with the idea of search, to understand the importance of search in a specific case, and to learn how to improve their search strategies.

The consequences for mathematics education, and for what mathematical thought requires, which result from these new cognitive technologies are remarkable: students need to learn, and can learn among other problem solving skills, how to search effectively. And 'although estimation skills are still central, error-free computation of sequences of operations on numbers and formulas is no longer as important as mental activity in mathematical problem solving' (Pea, 1985, p. 173).

Writing with Outliners and Idea Organisers Two dramatically different kinds of computer-based writing technologies will be described. Both of them are designed to better serve the externalization and revision of thinking processes facilitated by written language (cf. Pea & Kurland, 1987a).

The first type of tool is an outlining program. It provides a rich technology for interactively creating and revising a structured top-down plan of a written document. Several commercially available examples for personal computers are Thinktank (Living Videotext) and Framework (Ashton Tate). Their essential property is the capacity they afford the writer of portraying an outline at different levels of detail without revising the content of the document. With this facility one can quickly flip (usually in a keystroke or two) between different perspectives on the document, analyse part-whole relationships, and make and test revisions for their goodness of fit. Teachers using such programs report greater experimentation among students with alternative organizational structures, and vastly more attention during cycles of revision to how the details of the text contribute to the purposes of the whole document.

\texttt{Notecards} (J.S. Brown, 1984a) is a minicomputer tool created at Xerox PARC with a different orientation from the above described outlining program. It encourages bottom-up discovery and definition of relationships among ideas which the writer may have in mind initially only haphazardly, or which do not yield easily to top-down structuring early in the writing process. Through cycles
of shuffling and filing of notecards according to categories the writer can define, one can progressively discover idea structures during writing, which are based on ideas collected from texts and their annotations and linking by various relations (e.g. the rhetorical relations of evidence, comment, argument), which can then be reorganised into a map around which text can be generated. VanLehn (1983) has described in vivid detail his experiences with the powers of Notecards as a tool or organising process for the analysis of a complex text. He describes how, by explicitly tagging the nature of relationships between arguments and evidence with Notecards, he found loopholes in the intricacies of his own competitive argumentation for specific assumptions in his highly complex AI model of learning to subtract (VanLehn, 1983).

In both the above described cases, structurally distinctive features of the writing technologies provide the possibilities for reorganising one’s writing processes and for trying out different cognitive activities during writing. The closing of the temporal gap between thought and action, between hypothesis and experiment, which these technologies facilitate, and the rapid cycles of create-test-revise which they thereby make possible (much like the bases of spreadsheets and mathematics software) appear to have deep qualitative effects on how problem solving in writing is accomplished. Such processes are not anticipated or captured by the amplifier metaphor of computers and computing.

Other Examples Other examples of computer-based technologies which could lead to the reorganisation and not just amplification of human problem solving processes include:

- Complex planning aided by project management software and planning programs.
- Interactive computer programming, particularly in exploratory programming environments, such as InterLisp D (e.g. Shell & Masinter, 1983).
- Using computer databases (including icon-based graphic database systems, e.g. Filevision for Macintosh) and graphing software as tools for exploratory data analysis, for organising data, and for framing and testing conjectures of patterns among variables in the data (e.g. Conference Board of the Mathematical Sciences, 1983; Steen & Albers, 1981; Tuft, 1983; White, 1981), and
- Using simulated microworlds to explore principles of Newtonian mechanics (diSessa, 1983) and systems of mathematics (Abelson & diSessa, 1981) in intuitive rather than formal terms.

Further examples, less accessible today to schools because they tend to run on superminis or minicomputers, but equally dramatic in their cognitive implications for reorganising mental processes are:

- Powerful simulation programs, often incorporating highly realistic graphics, for exploring the workings of complex systems, such as electrical systems (e.g. SOPHIE: J.S. Brown, Burton & deKleer, 1982) or physical plants (e.g. STEAMER: Hollan, Hutchins & Weitzman, 1984), and
- AI programs such as expert systems and knowledge-based intelligent tutors.

Expert systems (e.g. Davis & Lenat, 1981; Feigenbaum & McCorduck, 1983; Hayes-Roth, Waterman & Lenat, 1984) are programs which emulate reasoning processes of experts in the field, and are used to support and guide complex problem solving. For example, they can dovetail the decision making processes of humans in medical diagnosis, design of new chemicals, computer-assisted design and manufacturing, automated factories, industrial scheduling, etc.

Knowledge-based intelligent tutors (e.g. Sleeman & Brown, 1982) build detailed models of student understanding and embody in their interactions a theory of tutoring. Issues concerning the broader relevance of these types of cognitive technologies for the future of human learning and development are discussed by Pea (1985).

In the above described examples, computer technology has come to provide cognitive power tools which can improve certain cognitive processes in such significant ways that, once the tool is understood and used regularly, the user feels better if it is not available. The computer has opened up new possibilities of thought and action without which one comes to feel at a disadvantage. For an increasing number of people computing has become an indispensable instrument of cognitive activity, and not merely an occasional tool (cf. Minsky, 1983; Simon, 1977).

Software can offer far more than an enhancement in the efficiency of mental operations or an increase in problem solving skills. The quantifiable products of human problem solving have indeed been enhanced, as even the amplification metaphor would lead us to observe, but the software has also restructured the thinking activities involved in such a major way that computer users come to develop new methods of thinking about their mental tasks and discover not previously thought of ways of using the computational tools. Thus, there are emergent properties of computer-aided thought that are unrecognised if one only subscribes to the amplification metaphor of computing.

Deeper Consequences of Cognitive Restructuring through Computing

On the whole, education has not accommodated itself to the strong benefits of these latest technologies. Instead, it has tended to assimilate the computer to its traditional fact-oriented agenda. For the most part, computers are not being used to extend and redefine the student’s powers of thinking and expression. A major reason for the prevalence of fact-oriented computer-assisted instruction in schools...
today is probably a commitment of the majority of educationists to the amplification metaphor of computing. Where efficiency and speed in achieving already defined and easily measurable educational objectives are the goal, drill and practice software (offering more exercises in less time) is a logical choice. Although a number of educators have begun work aimed to remediate this situation, less effort is being devoted to thinking about the ways in which computers can help serve as cognitive technologies to restructure both the cognitive processes of students (and teachers) and the broader context of the educational environment. Many schools now aim towards computer literacy of their students, but learning and teaching is often about computers rather than with the computer.

What alternatives are there? Before attempting to find answers to this question, the present context must be examined briefly. The restructuring perspective of educational computing, unlike the amplification perspective, is non-committal with respect to whether the consequences of restructuring of mental activities are positive or negative, developmental or regressive. Here, as in the study of child development, developmental progress is separate from the march of time. Development is an evaluative concept, not a descriptive one. In contrast, the amplification metaphor seems to carry with it the idea that faster and more efficient is better, i.e. the technology offers a means which is regarded as being more adequate to the task in hand.

The restructuring perspective is more problematic. How do we want the effects of computing to manifest themselves? What shapes do we want these effects to take? Television, for example, has opened up new global channels of visual communication and tremendous educational potentials. At the same time, some believe that this medium has hampered written language literacy because so much of the children’s time is spent listening and viewing rather than engaging in other literacy activities. Similarly, Plato’s familiar critique of written language in the Phaedrus, which suggests that the technology of writing will weaken people’s memories, makes clear the dark side even of a most important technological advancement. Both the positive and negative outcomes of a new technology must be considered.

Thus, it is important to go beyond the recognition that cognitive technologies can restructure mental functioning to arguments supporting specific ways in which they should do so. Such arguments must be theoretically and empirically grounded in our best guesses and our best psychological analyses about what our students will need to know about and do with computers over the next two decades, and ideally during their lifetimes.

Education, whether formally or informally acquired, is by its very nature a moral activity, in which choices are made to direct the paths of learning to socially valued goals. What should be the deeper aims of learning and development in computing, and how can education support these processes?

Which of our current learning objectives (many of them historical remnants of curricula defined in the 19th century) are still valuable and which ones are not?

There are some aspects of our students’ world that demand our attention, and that appear to warrant a novel approach.

Looking to the Information Society What should education be like in the information age? We and our students now live in a society which is increasingly dependent on computer-stored information and knowledge, on the use of computational tools for transactions with that information, and the requirement to understand and manage its complexities. A defining feature of this new society has been the information explosion. Knowledge obsolescence is a problem in most fields, and government and private industry need to spend millions of dollars to re-educate employees. Herbert Simon (1987) has pressed the point that in this information age knowing has become redefined as a verb describing access to knowledge rather than of possession of information. To know is no longer to have knowledge in one’s own memory, but to be able to effectively search for, find, and use the information one needs for particular purposes.

This paradigm shift has profound consequences for the goals of schooling, for the emphases of curricula, and particularly for the creation of appropriate roles for computer-based cognitive technologies in learning and teaching. Although the current uses of computers in education are leading to documentable restructuring of both mental activities and the contexts of learning, they are often unproductive when measured against the criterion of helping students acquire transferable knowledge and skills which will be useful in different contexts and/or over a long period of time. This is why we need to analyse our values with respect to education. Which explicit educational goals are most central, and with respect to which purposes? Answers to such questions are necessary to inform the choice and design of cognitive technologies for education.

With our predominantly fact-oriented curricula, we are hardly preparing our children for the life-long learning the information age requires. Regardless of our media, our aim should no longer be the hopeless task of pouring streams of facts through a straw into the child’s memory well, in the hope that the well-bucket will come up full with what is needed. Instead, we can work to help students learn for themselves how, where and when to seek out, organise and use information for different purposes. With this orientation, education becomes a process of enabling independent, critical and unique thinkers to take initiatives individually and collaboratively to pose and solve problems, and to apply and develop their learning and thinking skills while accomplishing required tasks. What is required is that we assemble a new vision for education in an age of technology that recognises and takes account of the causal powers of the individual (cf. Harre, 1984; Harre & Madden, 1975). It appears that knowledge of facts will still be useful, but as usable materials about events and problems and to help guide actions, not as ends in themselves nor as inert memory entries to be accessed at the time of assessment and then forgotten.
Emphasis on Cognitive Skills  An explicit cognitive skills emphasis is central to the issues considered above. For the reasons described, it seems that a productive approach for cognitive technologies in education will begin to

1 define the cognitive skills children will require, so that they can begin to be in control of their own learning and information management, and
2 design and create new technologies to help support the attainment and use of these skills.

The learning of such skills would thus become more explicit rather than a tacit objective of education, as many ideas in educational computing have been in the past. Among other aims which I see as central in the forms of information literacy (rather than restricted to computer literacy) called for today are:

A strong emphasis on cognitive skills relating to information management, rather than acquisition (e.g. Hawkins, Mawby & Ghitman, 1987), including the formulation of questions and the posing of problems, flexible strategies for information retrieval, information schematisation and inference, information synthesis and integration.

A renewed emphasis on written communication and critical inquiry skills, including the evaluation of arguments as well of sources of information and claims to knowledge.

Metacognitive and self-regulatory skills such as planning ahead, comprehension monitoring and evaluation, cognitive resource management or control (e.g. Schoenfeld, 1985), and learning how to learn (e.g. Dansereau, 1985; Weinstein & Underwood, 1985).

Strategies for creative thinking and inventive problem solving (e.g. brainstorming, problem decomposition, hypothesis formation and testing, and debugging approaches to a task) and systematic decision making methods (e.g. compositional approaches to comparing utilities of choices, such as cost-benefit analysis) which can crosscut knowledge domains.

Peer teaching and cooperative group problem solving, and the practice of negotiation skills.

Why are these types of skills important? They are important because they appear to characterise the cognitive performances of expert problem solvers in many disciplines, as the AI and cognitive science literature attests (e.g. Barr & Feigenbaum, 1982; Brown, Bransford, Ferrara & Campione, 1983; Greco & Simon, 1988) and because they are high-yield skills which can be expected to be useful throughout the life span, unlike the traditional fact-oriented curriculum. These broad sets of skills can also crosscut the too often segregated domains of the traditional curriculum, and one would hope that new cognitive technologies developed to support them could be used throughout schooling.

Software to Promote Transferable Cognitive Skills  Many forward-looking educators and schools have begun to help students acquire the thinking tools used by adults to solve problems in such disciplines as business, history, mathematics, and science, e.g. software for graphing, database management, word processing, and spreadsheet software. The difficulties of integrating adult versions of these tools (i.e. programs designed for different users and different purposes) into the curriculum have become obvious. Versions of these tools which are specifically designed for children have begun to appear during the 1980s, including the widely used Bank Street Writer (Kurland, 1987) and the Quill writing system (Rubin & Bruce, 1988).

For example, in school studies conducted by Char and colleagues from Bank Street College, New York (Char, Freeman & Hawkins, 1985; Hawkins, Char & Freeman, 1984), it has been found that the powerful information handling tools provided by database management programs require new skills (in problem definition, planning for searches of the databases, etc.) which many middle school students have not yet acquired, and that even some highly creative teachers who deeply value critical inquiry and information literacy are unsure how to teach these skills. How can technologies for education serve not only as tools for thinking, but as tools for thinking skills to develop?

Currently, there are no computer programs available which explicitly aim to tutor the development of thinking and metacognitive skills which are so important for life-long learning and problem solving. Although curricula for thinking and problem solving skills, such as those of Venezuela's Project...
Intelligence (Herrnstein, Nickerson, de Sanchez & Swets, 1983), which was developed with the assistance of Harvard University and Bolt, Beranek & Newman, have proliferated in the 1980s (see reviews in Nickerson, Perkins & Smith, 1983; Segal, Chipman & Glaser, 1985), we find no computer-based system for achieving these aims.

Several projects under way at Bank Street College, New York, may contribute to visions of what is possible. In one, Pea and colleagues are building and testing software tools for helping children engage in critical inquiry and construct a personal perspective about various topics, particularly in science, throughout the curriculum. In a second project the same group is building and testing a software environment to encourage the development and use of systematic decision-making skills, including problem definition, analysis of alternatives, evaluating attributes of alternatives, and various heuristics for comparing choices. Paramount in each case is the creation of both effective and enjoyable tools for learning by doing and student understanding of how to proceed, which will transcend the specific problem domain under study. The belief of Pea and his group is that if they create useful tools for thinking in these ways, the new visions of education described earlier will at least become possible because they are technically feasible.

We require cognitive technologies for education which embody an explicit knowledge transfer architecture, i.e. transfer activities are part of their very structure. Pea and others are exploring this approach to instructional design in a current research and development project on cognitive skills. In the design of IDEA (Integrated Decision Envisioning Aid), a specific domain of decision making -- family planning -- is used to introduce generalisable aspects of systematic decision making skills (e.g. goal monitoring, constraint planning, defining the space of alternative choices, analysis of attributes of alternatives, plan evaluation and monitoring). Multiple examples of the application of each targeted general decision making method are provided by the software. In this way the learner can at any time explore or be guided to learning generally useful aspects of methods which he/she is learning to apply in the specific case. One might expect that by combining the functions of a domain-specific problem solving tool with those of a general thinking skills coach, an effective program for learning complex thinking skills will emerge.

CONCLUSION

We need to design and engineer environments for the transferable learning that an information age requires. More specifically, to inform education effectively, theory and practice will need to be unified through the intervention of research-informed electronic learning systems which can be used in educational settings. As Greeno (1985) argued: ‘Important advances in instructional technology and in basic cognitive science will occur as an integrated activity’ (p. 2).

Research and development activities can be united in the creation of educational software prototypes and prototypes of curriculum activities, which are designed and built by interdisciplinary teams of researchers, educators and software developers, and progressively modified in response to formative testing with students. These prototypes can provide sophisticated learning environments for students and simultaneously serve as research tools for determining how skills and knowledge develop with these new cognitive technologies. I would argue that such technologies might serve as the educational infrastructure linking information processing research to educational practice, which Champagne & Chalklin (1985) suggest is necessary for cognitive science studies to have significant classroom applicability.

Some readers may disagree with the emphasis on the positive effects of computers as reorganisers of mental functioning. The reason for highlighting this implicating computing is that I believe that in the absence of prototypes guided by positive visions of what could be, it is unlikely that we will ever learn what education can become. Just as a child needs tools to think (Papert, 1980) as he or she learns to define and solve problems, so do we, as we work to reshape the aims and methods of learning and teaching with computers, in response to the challenges of an information society. We need to create a plurality of prototypes of electronic learning environments to work with, whose effects, positive and negative, can be empirically examined, reshaped, reassessed and debated, rather than the armchair inspired critiques of computers in education that have tended to overemphasise the long-term benefits of currently available software.

John Dewey (1915) criticised American education at a stage when it had yet to adapt to the changes brought about by the Industrial Revolution: ‘The primary waste is not money or resources but human life, the life of the children while they are at school, and afterward because of inadequate and perverted preparation’ (p. 59).

As in Dewey’s days, we are now in need of fundamental change, guided by research on student learning with emerging cognitive technologies and by communal dialogues about redefining educational aims. Everyone is a stakeholder in this enterprise of reform. Students, teachers, parents, researchers, industry and business, and policy makers all stand to gain or to lose. Working together to shape the technologies which will reorganise human thinking, we may be able to create a new system of education which addresses and fosters the creative spirit and flexibility of the human intellect, that builds on and discovers new worlds of cognition, action and play, made possible by the remarkable symbolic powers of computers, and that yields resilient adults who are ready to meet future worlds more radically different than we can even begin to imagine at this stage.
Teaching and Learning

What type of learning environment does one create to maximise the opportunities for learning and personal development of students who have their own laptops or have ready access to desktop computers? It may seem somewhat impertinent to include a chapter on aspects of teaching in a book which is likely to be read by experienced teachers, but discussions with teachers from schools in a number of states have shown that working with computers forces teachers as well as students to reconsider many of their existing ideas about learning and teaching, as well as about the relationship between students and teachers.

When personal computers are introduced into a classroom, they are viewed as a tool for learning. It is expected that learning with computers will allow students to take more responsibility for what and how they learn. Some people might interpret this as an extreme form of discovery learning. They believe that it is the teacher's job to set up the hardware and provide some occasional maintenance of the equipment, give students basic instructions on the use of their laptops, and that from then on learning will take care of itself. It is not difficult to see why this view prevailed. As was noted in previous chapters of this book, the philosophy of Logo is child centred. It emphasises learning much more than teaching. Students are expected to work on their projects and teachers are advised to consider their interventions carefully. But does not mean that teachers should not intervene at all, or that the teacher is unimportant even in the Logo classroom. On the contrary, the teacher's role is vital, and it involves teaching and not just managing the classroom. It is certainly appropriate to give information or to suggest a particular action for some students. Different approaches will suit different needs and situations, so what follows will not suggest hard and fast rules. What is clear is that 'computerised information storage and retrieval is capable of offering liberation from cluttered brains and thus giving freedom to concentrate on the development of flexible thinking skills' (Chandler, 1984, p. 56).

The teacher plays the strongest part in the creation of a computing culture for his/her classroom. This culture needs to comfortably support all the students and the teacher. Teachers who are planning to teach students with computers need to examine their own attitudes about computers in society, personal computing and computing in the classroom, acknowledge for themselves the areas creating distress and identify the areas which they believe can be addressed with optimism. Major prerequisites for the teacher are enthusiasm, knowledge, some experience, hardware and, most importantly, time.

One theme which will run through much of this chapter is that of control. Traditionally teachers have control over almost everything that happens in the classroom, from deciding what is learnt, and in what order, to organising seating arrangements and rationing paper and pencils. Using computers provides an opportunity to encourage children to take responsibility for their work. Handing over, sharing and accepting control can be difficult for teachers and students. In this chapter an attempt is made to highlight some of the special features of the teacher's role in teaching children with computers which can help the process of sharing control. Some of the suggestions made may already be part of the reader's teaching style, others may be less familiar. We shall first consider some philosophical points, briefly introduce Logo and then turn to issues relating to classroom organisation and the interaction with the students.

PHILOSOPHICAL ISSUES

Piaget (e.g., 1952, 1954, 1973) stressed that the principal goal of education is to create men and women who are capable of doing new things, not simply repeating what other generations have done: men and women who are creative, inventive and discoverers. The second goal, he suggests, is to form minds which can be critical, which verify, evaluate and not just accept everything they are offered. In other words, Piaget asks that education should produce independent thinkers and learners.

What Is Independent or Self-regulated Learning?

The terms independent learning and self-regulated learning are regarded as synonymous here. Both imply that students themselves take charge of their cognitive efforts, and that to a large extent they manage their cognitive skills, abilities and motivation. Independent learners are motivated to succeed and/or to avoid failure. Independent learning combines cognitive strategies, knowledge, skills and motivational states in ways that develop coping tactics and thus preserves feelings of self-worth in students. As they become more independent, learners develop confidence in their learning and problem solving abilities. Confidence in the ability to regulate their own learning enables individuals to attack challenging tasks and to persist in the face of difficulties. This confidence distinguishes mastery-oriented students from students who avoid failure by being passive or defensive about learning tasks; it also helps to establish motivation, to take risks in problem solving, and to expend the effort and perseverance necessary for difficult tasks.
Independent learners approach tasks strategically. From a repertoire of previously acquired knowledge and learning strategies they have learnt to select those which are likely to be appropriate to a particular task and situation. They can evaluate tasks, plan various options and modify their cognitive strategies. In sum, independent learners are aware of effective learning and problem solving procedures (because they have prior experience in their use) and are able to take control of their actions.

Another way of describing the characteristics of independent thinking and learning is as a set of cognitive preferences, or characteristic ways in which an individual conceptualises and deals with his/her environment. Independent learning is a way of approaching tasks, issues, etc. - a preparedness to organise information and experience for oneself, and a conviction that certain strategies are important, effective, efficient, worth some extra effort and instrumental to success.

Conceived in this way, independent thinking and learning become a set of information processing habits. Not simply habits in the technical sense of learning theory (as they are not directly responsive to behaviourist principles of acquisition and extinction); instead they are more generalised habits of thought; not just a tendency to use specific behaviours that have become relatively enduring or automatic through repeated performance, but rather the enduring structural and functional bases for such behaviours. Defined in this way, independent or self-regulated learning involves both dispositions and abilities/skills, which teachers encourage and facilitate in their students.

It is now generally accepted among educationists that learning is a continuing process of information acquisition, transformation, association, storage, retrieval and evaluation. Continual interactions and restructuring of perceptions, knowledge, experience, emotions, motivations and interests take place within the individual as he or she adapts to the demands of the environment or adapts the environment to his or her needs.

All human beings have the power to further develop their intellectual strengths throughout life. However, there are two prerequisites for this:

1. The individual must be motivated to continue learning and to exercise his or her intellectual abilities, and
2. The individual needs to have the basic knowledge and practical skills in how to go about these cognitive activities.

Teaching is one of the most powerful mechanisms for facilitating and developing independence and self-regulation in cognitive behaviour. The meaningful communication with adults, peers and the broader environment is as essential for the development of both these prerequisites as for general intellectual growth.

Goals and Assumptions

Teachers foster independent thinking and learning by encouraging students to participate more actively in the communication process of learning/teaching and by guiding them towards assuming control of how they (the students) process information.

The goal of such teaching is based on assumptions which differ somewhat from the assumptions which have traditionally provided the framework for the teaching of subject content. Here are examples of the assumptions made by teachers who encourage independent thinking and self-regulated learning in their students:

There are large individual differences in abilities, skills, interests, motivations, thinking and learning strategies in the students in the class; what works for one student may not work for another.

Students may ultimately learn to learn and think independently, but not merely because the teacher taught them. In a very real sense, students must teach themselves, i.e. they must find out for themselves what method of problem finding, problem solving and learning work for them. All the teacher can do is to provide every possible means to facilitate the construction of knowledge by the students for themselves.

Some teachers are so preoccupied with the evaluation of learning outcomes (i.e. the correct answer attitude) that they ignore the processes which are taking place during problem solving and learning. In independent learning it is the way the student goes about a task and the related thought processes that count. Very often there are no immediately scorable answers. Ultimately, students who think and plan well will be in a position to generate good answers. However, good answers are not necessarily the result of good and/or independent thinking.

For centuries thought was regarded as something that originates in the individual’s mind, i.e. inside the individual, and is then expressed socially. More recently we have come to recognise the importance of social interaction in the formation of personal ideas. Thought emerges, to a large extent, as a social process and is internalised by the individual only after it has been expressed socially. A substantial portion of our ability to think thus originates outside ourselves. Interaction and interchange of ideas in discussions with others is essential. This is true at all levels of development. Learning is about communicating ideas and restructuring personal knowledge, attitudes, etc., as a result of such communication.

To enhance independent thinking and self-regulated learning we need to serve not strictly as teachers, but as models and facilitators, or as communicative learners (Black, 1988). We must recognise that we too are learners, and that we encourage independence in our students by fostering and taking part in collective efforts to identify questions and possible answers.
Class discussion is more than a peripheral part of the curriculum. To the contrary, class discussion is a legitimate and integral end in itself, because it is in such discussion that ideas are produced, shared, reflected upon and internalised. The intellectual development of the individual is enhanced as a result of the group’s producing a best possible collective product. How often do we become aware of how difficult it is for members of a group to identify who first had which idea?

Hence, to accomplish the goal of enhancing independent thinking and self-regulated learning in our students we need to abandon some of the assumptions of traditional instructional design, which is rooted largely in behavioural theory, in favour of new designs, which tend to be rooted in cognitive theory. We abandon assumptions such as:

- The teacher is the trainer and the student is the learner.
- Learning is a task for the student, and (in school) only for the student.
- The only thing that counts is the correct answer.
- Class discussion is primarily a means to an end.

In stressing the social and communication aspects of learning, learning to learn and training for cognitive independence, I do not wish to give the impression that the teacher’s role is in any way diminished. The aims may have changed, but if anything, more is required of the teacher.

A dominant focus in learning and teaching is on changing students’ perceptions of aspects of the world around them. The way in which they think about particular phenomena is at the core of education. However, learning in any discipline involves more than the student making sense of his or her personal perceptions and experiences; it also involves being initiated into ways of seeing which have been established and found to be fruitful by the cultural, social, academic or scientific community. Such ways of seeing cannot really be discovered by the learner -- and if he/she happens to come across or hit upon such consensus viewpoints he/she would be quite unaware of the status of the ideas. It is the teacher who introduces ideas and viewpoints from the outside world of learning into the class discussion. Even more importantly, in most situations only the teacher can provide feedback as to the consensus and status of these ideas.

**Meaningful Tasks**

A central theme in Dewey’s writings (e.g. 1902, 1938) on education is the notion that classroom activities must be related to the child’s experiences, interests and goals. This was a radical view for an era in which didactic teaching was the most acceptable method of instruction. Although the general notion expressed by Dewey has found wide acceptance in Australia and other Western countries in recent decades, many teachers are experiencing difficulties in implementing it, because of limited resources, materials and training. It is an expectation for many people in the field of educational computing that the personal computer can be a resource for engaging children’s interest and fostering more active, creative and independent learning. This expectation is based on two assumptions, namely that children are intrinsically motivated to work on tasks which are meaningful to them, and that the most effective educational environment is one which provides meaningful tasks, that is, tasks which embody a function or purpose that is understood by the students.

While some children enjoy learning about a particular topic for its own sake, in most cases facts and skills are best learnt in connection with wider concepts and ideas which give them meaning and significance. In this way, not only are students motivated to master the facts and skills, but they have a framework in which to understand the logical, scientific, technological, social or cultural significance of the facts and their relationships to other facts.

The above assumptions leave two fundamental questions unanswered:

1. Where do the goals which interest the students come from? Are they inventions of the students, or are they imposed by the teacher?
2. What is the relationship between the goals which students work towards in the classroom and the tasks with which they will be confronted in the world outside school?

Dewey regarded the extremely child-oriented approach as as unsuitable as the traditional view that the teacher must impose the classroom tasks. The teacher has very important responsibilities, which include suggesting tasks and presenting to the students alternative interpretations of problems. In many respects Dewey’s approach is more consistent with the socio-historical approach to child development described in the writings of Vygotsky (1978) and Leont’ev (1981), in which the importance of the teacher/student interaction is emphasised, than with the universalist approach of Piaget which de-emphasises the cultural context.

Meaningful tasks may come from a variety of sources. One source is the pool of spontaneous ideas children themselves have. Most children have one or more topics which they simply like. However, for most topics in classroom learning, this source may not be the most important. Teachers can make classroom tasks meaningful by showing students their significance in terms of a variety of uses for the skills involved. The functional learning environment created in this way can be a simulation of the real problem (e.g. role-playing a business transaction as a context for doing arithmetical calculations), or it can be a real problem (e.g. actually having a stall at the school fete to raise money for another computer). The functional learning environment can also be of a more abstract nature (e.g. a geometry problem can provide a meaningful context for calculating the size of an angle). A teacher can create interesting functional learning environments by
crossing traditional discipline borders (e.g. by showing how geometric concepts such as triangles can be used in geography to solve navigation problems).

An approach to the second issue, i.e. the relationship between classroom and real world goals, is closely related to the first. It would be reasonable to suspect that transfers of learning from one domain to another and the usability of school learning in later life are inseparable from the variety of functional learning environments in which they are embedded. Being able to see the same fact from multiple perspectives (e.g. recognising the different uses that can be made of a tool) engenders a flexible approach to acquiring knowledge that would otherwise be absent. This flexibility makes it possible to adapt knowledge to new functional environments that cannot be specifically anticipated in the classroom.

Personal computers can play a useful role in functional learning environments because of their capacity for simulation and because they themselves are important tools for the solution of a variety of interesting real world problems. Computers do not function on their own. A teacher must build the bridges between the tool, the school task, the thinking skills, and their functional significance for the culture beyond the classroom.

The teacher's role in educational computing is to provide an environment which is in sympathy with the child's level of development in order that appropriate intellectual leaps can be made as efficiently as possible. The teacher has a far more active role in classrooms operating according to the principles advocated by Piaget (1973), Vygotsky (1978), Leont'ev (1981), and Bruner (1983) than in the traditional classroom. For example, the teacher's role might be considered as one designed to provide a scaffold for children's problem solving (Wood, Bruner & Ross, 1976). Here the teacher is a key interventionist, providing help when the student is in difficulty, standing aside when he/she succeeds, and generally supporting such abilities as to select, remember and plan which might as yet be underdeveloped in the students.

These ideas have had profound influences on the organisation of modern classrooms, and they also provide the basis for an equally dramatic reorganisation of thinking about the use of computers in classrooms. In place of drill and practice, Papert (1980) offers a Piagetian vision of cognitive development driven by interaction not so much with the total world itself, but with a simulated microworld accessed through the Logo programming language.

**Constructing Knowledge**

It has long been recognised that learning (with understanding) involves the structured organisation of a knowledge system in which concepts take their meaning from the theories in which they are embedded. Central to this perspective is the historically important view that learning comes about through the learner's active involvement in knowledge construction. Within this broadly constructivist perspective, learners are thought of as building mental representations of the world around them which they use to interpret new situations and to select actions within them. These mental representations or conceptual schemes are in turn revised in the light of their fit with experience. Learning is thus seen as an adaptive process, in which the learner's conceptual schemes are progressively reconstructed so that they are in keeping with a continually growing range of experiences and ideas. It is an active process of sense making over which the learner has some control.

In so far as it views learners as architects of their own learning through a process of equilibration between knowledge schemes and new experiences, this perspective reflects and builds on Piagetian views. It differs from Piaget, however, in two significant ways. Instead of focusing on the development of general logical capabilities, this theoretical position emphasises the development of domain specific knowledge structures. In addition, whereas the emphasis in Piagetian theory has been on the personal construction of knowledge through an individual's interaction with the physical environment, the current constructivist perspective also acknowledges to a greater extent the social processes in knowledge construction both at the level of the individual and within the community of experts. The writings of Vygotsky have been increasingly influential in shaping thinking about these social and cultural influences. What is internalised by the child during learning is not what the experts say, but a version of the interactions that constitute the joint activity. Thus, without coercion, these interactions guide children towards the cultural interpretation and significance of the tasks in which they are engaged (Newman, Riel & Martin, 1983).

The essence of this school of thought is that the human intellect develops naturally through interaction with the environment. Through this interaction the child discovers the properties of the world and the characteristics of his/her own relationship with the world. The contention is that interaction is of prime importance, because it is the only way in which we can come to understand our personal world and learn how to operate within it and upon it. The reality which we come to understand is, under this theory, a personal construction, and the process of construction is fostered by interaction experience. Cognitive development is seen not as the product of an accumulation of facts, but as being driven by the individual's interactions with the physical and social environment. According to Piaget's theory, learning is under the control of the learner and knowledge and skills cannot be taught directly. Self-directed thinking, learning and problem solving, through actions in the world which teach us by means of feedback which they generate, is the essence of this view of development. It is a view which transfers readily when we consider the impact of computers in the classroom, for here we can provide children with a rich microworld which they themselves can explore with ease and little risk.

If education is to be effective, it must take into account the student's contribution to the learning process. Educators must consider how learners interpret the accepted body of knowledge, including both content and technique. A fundamental principle of cognition is that learning requires knowledge. Yet,
cognitive research also shows that knowledge cannot be directly given to students. Before knowledge becomes truly generative (i.e., knowledge that can be used to interpret new situations, to solve problems, to think and reason, to learn) students must elaborate and question what they are told, examine the new information in relation to other information, and build new knowledge structures. Educators are thus faced with a central problem: how to help students start developing their base of generative knowledge so that they can learn easily and independently. Teachers and curriculum developers will have to learn more about what students understand, and then apply what they find out, to improve teaching.

Problems to be solved have to become the students' own. How individuals perceive tasks is influenced by their own generalisations and extensions of the information they are given by others. By the time this point is reached the students are no longer working on the teacher's problem; rather they are exploring their own. In short, they are doing mathematics, writing, logic, literary interpretation, etc. The task as given by the teacher may be seen as a springboard for discussion (including amplification) of ideas such as establishing subgoals, working backwards, assuming you have a solution and determining its properties, exploiting extreme cases, solving the problem in more than one way, generalising, and creating one's own problem. It is up to teachers to:

- help students understand that a problem is not a problem until one wants to solve it;
- build a supportive classroom atmosphere in which students will be prepared to tackle the unfamiliar and not feel threatened when they experience difficulties;
- allow students to pursue their own paths towards solution and assistance when necessary, without giving the answers away;
- provide a framework within which students can reflect on (i.e., think about, discuss and write about) the processes involved and thereby learn from experience;
- talk to the students about the processes involved in doing and using mathematics, science, geography, writing etc., so that they can build up a vocabulary for thinking and learning about it. Students learn much more effectively when the teacher draws their attention explicitly to the strategies and processes involved.

To attain this goal teachers need not only a clear conception of what is to be learnt but also an ability to see this knowledge through their students' eyes. More specifically, this ability includes the following:

- Knowledge of students' typical interpretations of questions, instructions, procedures, and vocabulary at given ages and levels of achievement.
- Knowledge of individual children's unique interpretations of these same topics.
- Knowledge about how to introduce formal domain knowledge by building on students' existing abilities, by helping them to generalise informal knowledge to new and abstract situations, and by encouraging the formation of connections between what the student knows and the abstract representations of, say, mathematics.

Constructivism has multiple roots in the psychology and the philosophy of this century. These include the developmental perspective of Jean Piaget, the emergence of cognitive psychology under the guidance of such theorists as Jerome Bruner and Ulric Neisser, the constructivist perspective of philosophers such as Nelson Goodman. Central to the vision of constructivism is the notion of the organism as an active agent, who does not merely react or respond to stimuli as in the behaviourist view, but engages and grapples with his/her context, and seeks to make sense of things.

In particular, learners do not just take in and store up given information. They make tentative interpretations of experience and go on to elaborate and test these interpretations. Even when the learning process appears to be relatively straightforward, for example when learning a short poem or a new word in a foreign language, constructive processes operate. Approximate mental structures are formed, elaborated and nested, until a satisfactory structure emerges.

The main thrust of Piaget's work has been to map out distinct stages of development and to describe ways in which the child constructs a view of the world which either accommodates existing cognitive structures so that they fit with new knowledge, or assimilates incoming information so that it fits existing structures. The view is strongly individualistic and constructivist; the impact of this theorising upon educational practices is best exemplified by individuals engaged in discovery learning. Piaget makes some reference to the significance of collaborative work and the importance of cultural contexts, but these themes were not part of his research. In contrast, Vygotsky (1978) has argued that learning and cognitive development result from a process which is essentially social rather than individually based. The nature of education is to share meanings and interpretations of what happens in the world by an elaborate communication process. Essentially the skilful teacher provides tasks which lie within the learner's zone of proximal development and provides enough support to allow the learner to succeed. As a result of assistance received on tasks which lie within the zone of proximal development, the child learns to internalise the processes offered by the teacher, so that the nature of what is learnt, and the cognitive development which results, will be determined by the environment in which learning takes place. Vygotsky also talks about the zone of proximal development in interactions between a child and more able peers.

Vygotsky's approach has been contrasted both with approaches in cognitive science (e.g., Edwards, 1990) and with Piagetian theory (e.g., Smith, 1989). Piaget is seen to offer a biological view of development, and Vygotsky a social view. One might view a Vygotskian perspective as one where the learner is led towards
All these approaches share a view that knowledge is socially constructed, and all are consistent with general constructivist views (e.g. Neisser, 1976; Berger & Luckman, 1967) that humans interpret the world around them and build theories (though often implicitly) about all aspects of their lives. At any time these implicit theories shape the way the world is viewed and the way events are interpreted. Constructs about events or people can be changed by evidence, discussion, reflection or direct teaching, and it is almost certain that no two people will see the world identically. Failure to take account of current constructions, and the way they might be modified, is likely to lead to a failure to modify these constructions at all.

If learning has this constructive character inherently, it follows that teaching practices need to be supportive of constructions that occur. The critique by constructivists of conventional teaching practice is that it is not supportive enough of the constructive processes which need to take place in the minds of the learners.

A considerable amount of computer-based instructional material is currently available to teachers, much of it of either the tutorial Computer Aided Instruction (CAI) or Intelligent Computer Aided Instruction (ICAI) kinds. These materials implicitly build on the view that knowledge exists in some absolute sense, that the structures imposed upon the world by current educational frameworks are correct, that knowledge is acquired by individuals, who accept the structures within which it is presented, that the endpoint is acceptance of some abstract intellectual structure which will be similar across learners, no matter what the original context of learning, and which can be applied to a range of domains. This view conflicts with constructivist ideas and with evidence from cognitive psychology. Ridgway (1988) discusses the associated pedagogic problems.

**LOGO**

The computing language used in the SUNRISE classrooms at Coombabah is Logo. A strong view exists in the educational community in Australia and overseas which suggests that Logo creates a good environment for programming, and encourages good programming techniques. However, BASIC shares these as well as other characteristics of LOGO, including its versatility and availability. An initial aim of the SUNRISE project appears to have been to answer some questions about the cognitive and social impact of Logo in a Year 6 and Year 7 classroom. An interwoven theme was how student and teacher assumptions and understanding concerning the nature of programming and its requirements changed as they became increasingly familiar with the programming culture emerging in the classroom. In this section we reflect on how our observations enabled us to look more closely at the distinction between the cognitive skills that might be practised through some uses of formally elegant symbol systems such as Logo and the way in which it evoked particular practices in the classroom.

During the first year of the SUNRISE project at Coombabah (cf. Ryan, 1991), teachers intended the computer activities to be largely child-initiated, so as to encourage the child-centred, Piagetian learning without curriculum advocated for Logo (Papert, 1980). While the teachers in the first year gave students some simple instruction in Logo during the first weeks and occasionally held group sessions to introduce new aspects of Logo during the year, their self-defined role was principally that of constructively responding to students’ questions and problems as they arose. Students’ primary activities were the creation and development of their own computer programming projects.

The second year appeared to differ from the first in that at least two of the teachers decided to take a more directive role in guiding their students’ explorations of Logo. These teachers gave more regular group instruction to introduce key computational techniques, and to demonstrate how they work in procedures. Students were required to complete specific assignments which required familiarity with Logo concepts and basic programming skills.

Many educators have been focusing on the use of computers for drill and programmed instruction -- to provide individualised practice and instruction in usual curriculum areas. In the SUNRISE classrooms at Coombabah the teachers have agreed, informally among themselves, on additional aims which involve making use of computers:

- to provide an environment in which learning can be intrinsically motivating and fun;
- to allow students to discover, explore and create knowledge;
- to help develop skills of thinking and problem solving;
- to make some of the most powerful ideas of the developing computer culture accessible and tangible to students at an early stage of their schooling.

A most striking impression the visitor to the SUNRISE classrooms gains is that of the powerful motivation which the computer displays, especially the graphics can create. Every bit of the student’s attention is focused on the screen. And this powerful motivation is waiting to be harnessed towards intellectual growth and learning.

**What Exactly Is Logo?**

Logo is a computer language which was developed to provide an environment which allows learning to take place as naturally as possible. Seymour Papert and his colleagues Bolt Beraneck, and Newman, and later at MIT, set out to create a computer language which would combine the capabilities of artificial intelligence
with the theories of Jean Piaget in order to allow a learner to build his own intellectual structures through estimation, interaction, experience and revision.

Logo is one of the most powerful of computer languages available for personal computers today. The power of a computer language does not come from what you can do with it. Any program you can write in Logo you can also write in BASIC, Pascal or FORTRAN. Rather the power of a computer language is related to what you think with it. Less powerful languages, like BASIC and FORTRAN, force you to attend to lots of details, such as where you must put a semicolon or how long a word can be. Logo has a few simple rules of syntax which are applied uniformly, which makes it easier to focus on the task at hand. (Friendly, 1988, p. viii)

Logo is a list processing language which can be used to achieve a number of purposes, for example text processing, interactive simulation and music production. The language is probably best known for its graphic capabilities. Logo graphics have been used for many mathematical purposes including the acquisition of geometry and mental arithmetic skills, as well as the appreciation of general heuristics for problem solving such as breaking problems down into subproblems. Different versions of Logo graphics are available, and have been extended beyond the production of graphical displays on the screen to more concrete manifestations as turtle graphies. For example, a robot, which can look remarkably like a turtle, drags a pen around a piece of paper on the floor. Instructions to move the pen are given through the computer in the usual way, with the theories of Jean Piaget in order to allow a learner to build his own intellectual structures through estimation, interaction, experience and revision.

According to Papert (1980) Logo is an environment in which children can learn fundamental mathematical concepts and powerful problem solving methods without the intervention of teachers. Papert takes his inspiration from Piaget, who has argued forcefully that ‘each time one prematurely teaches a child something he could have discovered for himself, that child is kept from inventing it and consequently from understanding it completely’ (Piaget, 1970, p. 175).

The Logo language is designed to provide an environment in which self-directed and independent learning are encouraged. The learners themselves should be in charge of

- seeing a problem to solve,
- making choices,
- playing with the problem, experimenting and trying out solutions,
- building on what he has already done to do something more. (Harper, 1989, p. 1)

One of the most popular aspects of Logo is that it allows for the creation of graphics effects with repeated sections, such as the petals on a flower, or trees in a forest. To produce a shape, the turtle follows commands to move forward by a stated distance, and to turn right or left a stated number of degrees. Combinations of commands can be given names, and then be used as procedures within other parts of the program by reference to these names. So, if the student wants to draw a flower with a number of petals, the instruction for a single petal would be given a single name, together with an instruction for moving to the starting point for the next petal, and then in order to draw all the petals with one instruction the student would give a repetition instruction. Once the flower is complete, then all of the instructions necessary for the flower could be given one name, and this single instruction would produce a complete flower. Similarly, a patch of flowers could also be drawn by a single instruction which called upon the procedure for each flower, and so on. The claim is made by Logo theorists that the experience of debugging is of particular benefit for the development of more general problem solving skills.

**Evaluation of Logo Effects**

Papert and his colleagues claimed that experience with Logo benefits children’s cognitive development. As was noted in Chapter 2, attempts to evaluate this claim have brought mixed results. Strong support for the assertion was provided by Robert Lawler (1985) whose book *Computer Experience and Cognitive Development* describes the extensive case study conducted on his six-year-old daughter over a period of six months. Lawler himself acted both as personal tutor and evaluator. He concluded that the effect of this experience with Logo allowed his daughter to demonstrate behaviour typical of a child in Piaget’s stage of formal operations, i.e. far beyond the expected attainment of an average six-year-old. The examples of her problem solving, planning and debugging activities which he presents are certainly impressive.

Early evaluations related principally to the Brookline project (Papert, Watt, diSessa & Weir, 1979) and the Bank Street studies (Pea & Kurland, 1983; Pea & Shengold, 1987). The Brookline project report contains positive evaluations which are themselves difficult to assess, but the Bank Street research found no differences between a Logo group and a control group on a non-programming planning task. The failure to find improvements in planning is important, because this is one of the few direct tests of the claims regularly made for the benefit of learning to program. Finlayson (1984) and Clements & Gullo (1984), discussed in Chapter 2, described clear benefits of Logo experience for the development of mathematical thinking skills. More recent studies have confirmed the positive effects of Logo programming for the early development of mathematical concepts (e.g. Hughes & Macleod, 1986; Robinson & Uhlig, 1988).

Not all evaluations of Logo have found positive effects. Pea & Kurland’s review (1984) comes to the conclusion that the idea that programming experience can transform children’s minds is itself a form of *naive techno-romanticism*, and after reviewing a number of Logo evaluations Simon (1987) agrees. Whereas...
most recent reports (e.g., Underwood & Underwood, 1990) do not support such a
scathing dismissal, some caution in the acceptance of all claims by the
proponents of Logo is warranted. Certain benefits can be observed in children.
These include students' ability to generate creative ideas, the development of
spatial skills and improved numeracy. There are studies which have shown
positive transfer from Logo debugging to other debugging tasks (e.g., Lawler,
1985; Klahr & Carver, 1988), and also studies which show improvements in
specific mathematical and spatial abilities after learning to program (e.g.,
Clements & Gullo, 1984; Finlayson, 1984; Hughes & Macleod, 1986; Robinson
& Uhlig, 1988).

As with many applications of computer-based learning, one of the greatest
attractions of Logo is the motivation that it generates in the children using it
(Lepper, 1985; Hughes & Macleod, 1986). Mostly this is measured by time-on-
task; Obviously, claims for educational benefits must be based on measures that
are more profound than the latter if we are to improve the quality of students'
cognitive skills and not only their powers of concentration.

A review of the literature relating to uses of personal computers as aids to the
development of children's thinking revealed that learning to program with Logo,
using databases (e.g., Underwood, 1986; Underwood, 1989; Underwood &
Underwood, 1990), and using problem solving games and simulations, can each
be seen to produce changes in the ways in which users think about their worlds.
There is no curriculum as such in the use of these programs: these applications
are educational tools with open-ended uses. Irrespective of the specific
educational goal, what is acquired by the student is procedural knowledge. In the
case of database and simulation activities some investigators have found sudden
and strong developments in hypothesis testing, categorisation and questioning
skills of students (Underwood & Underwood, 1990). Most of the gains noted
here and in Chapter 2 were observed after only a short period of computer use.
Education is a long-term activity, but educational research projects relating to the
use of computers in schools tend to look for changes after a few months of
experience with the computer. Unfortunately, our empirical study in the
SUNRISE classrooms at Coombabah was no exception to this tendency. As
Snow & Yallow (1982) have shown, the impact of any one educational treatment
may not manifest itself for several years, and equally may continue to show an
effect when students have moved from one school to another. It is impossible to
decide whether any of the studies reported in the literature would have come up
with any long-term changes in the cognitive development of the children who
participated. The measures just were not taken. It is unfortunate that so few
research projects look for changes over the course of several years rather than
weeks or months.

CLASSROOM ISSUES

Teachers faced with the task of integrating computers into learning and teaching
for the first time will find themselves in diverse and perhaps initially
uncomfortable roles. In addition to more traditional activities, such as being an
instructor, demonstrator, evaluator, etc., they will take on such jobs as technician,
timekeeper, solver of management tasks, observer, collaborator and model
learner.

The role of the teacher . . . is facilitator and co-learner, rather than the source of all
knowledge. There are a number of ways to organise a classroom [with computers] to support
this role and produce a suitable learning environment. These ways include demonstrations,
peer tutoring and group work. (Queensland Department of Education, 1986. p. 3)

Teachers share in the process of learning with their students. They seek
information from their students, observe and document observations, and extend
ideas. Two of the teachers at Coombabah have recently reread Papert's work, and
they read some journals relating to educational computing, particularly on the use
of Logo. On the whole, teachers introducing computing into their classrooms for
the first time are faced with enormous time pressures and are finding that the best
they can do is just to implement what they know.

Hardware

An obvious aspect of the teacher's role in a computer-rich classroom is that of
managing the hardware, i.e. the computers, printers, disk drives, floor turtle, etc.
It is quite natural for teachers who are unfamiliar with the machinery to be
nervous, and it is important that they allow themselves enough time to become
confident. It also helps to show the students how to use all the facilities they
need. Technical assistance should be available to teachers at relatively short
notice, otherwise much teaching time will be wasted by the teacher in the role of
novice technician.

Students will need time to get over the novelty of using the computer, printers,
etc., particularly when they are also getting used to the freedom to choose what
they will work on. Time will have to be apportioned for this purpose. The printer
is a valuable device and it is good to encourage students to print out their
procedures, so that they can study them more intensively than they could if they
were displayed on the screen. Some teachers report that initially, whenever a
student wanted to print out a procedure, the printer was already occupied by other
groups using it to get (several) copies of pictures they had drawn. It is perfectly
reasonable for students to take away a copy of a screen picture they have
designed, but this is very time-consuming. It might be tempting to limit this
particular use of the machine, but planning might make it possible to allow things
to progress more naturally. Gradually students will become tired of printing out everything. They will learn to discriminate.

Resources

Among the resources in the classroom should probably be a handbook on Logo, BASIC or whatever computing language is being used, which is written in a way that is accessible to the students. This is important if we want the students to answer some of their own questions independently. Some manuals start with a tutorial section which the reader has to work through. This is not likely to be helpful in the classroom. The aim should be to provide a resource which could be picked up for just a few minutes and then replaced when the required information has been obtained. A home-made booklet with explanations which are short and to the point might be the best idea. An example of the use of a primitive need only be given, if its sense cannot be conveyed in any other way. The manual is meant to be used as a reference, either to remind the student of the syntax of a particular word or when a new word looks as though it might be appropriate for a particular task in hand.

In the SUNRISE classrooms at Coombabah the students are given page by page explanations of commands with which they make up their own manuals. As a consequence of this, the presentation of the manual might be quite dry and the students are not as involved with it as they would be if it were more contextual. Students need to be taught to use available resources. For this reason, it might be best not to use manuals or handbooks for the first few weeks of the school year. In the early stages students really need only a few commands, and the teachers can provide the format for the use of these by designing a large poster. Booklets and handbooks can be kept in a cupboard and introduced slowly, i.e. whenever it seems appropriate for particular students. Wall posters can be used to illustrate essential early commands and to display students’ work regularly. The latter might include screen dumps together with computer code, to encourage students in yet another way to develop projects.

Curriculum

In planning classroom experiences aimed to develop independent thinking and self-regulated learning with or without computers, it is important to consider the developmental levels of the students, the mode in which information will be presented, and the subject matter that is to be acquired eventually.

As in all instructional planning, learning tasks in computing generally move from those requiring simpler operations to those which are more complex, i.e. from more concrete and observable to more abstract dimensions, and from an emphasis on working with known materials towards creating or inventing new, previously unfamiliar approaches. Some aspects of independence appear to develop slowly and experientially, but their development is facilitated by tuition and practice. The same applies in learning with computers.

Learning activities can be developed which cover a wide variety of areas. For example Friendly (1988) demonstrates a range of educational domains which can be explored concretely in a Logo learning environment. These include generative grammars, physical laws of motion and mechanics, artificial intelligence and robotics, and the ideas of calculus. Some of these seem to be quite difficult. The important characteristic of Logo as a tool for learning is that it allows difficult concepts to be defined as procedures which make the computer actually do the thing which the concept means. The important characteristic of Logo for educational purposes is that it provides the means to create concrete, often graphic, models of learning domains, which can be manipulated by the learner. Such simulations, often referred to as microworlds, show how the rules or laws of a particular system work. Students are encouraged to ask What would happen if I changed the rules? and can thus come to understand ideas and theories which can go far beyond what would traditionally be expected at school level.

Students can learn new ideas from one another either by looking at wall displays or one another’s screens or by listening to peers describing new projects. Sometimes it is necessary, however, for the teacher to introduce a new idea, because none of the students stumbled across it or asked a question which allowed the teacher to introduce it in the context of their work. For example, a teacher who was teaching a group of able 12-year-olds with Logo for three weeks found that none of the students were using variables. The teacher did not want to have a formal class lesson, where the students had to listen before they would do some boring exercises, but he did think it worthwhile to bring the idea of variables to the students’ notice. The teacher’s resolution of the problem was to spend ten minutes without computers, explaining to the class how to write a procedure with inputs. He did not dwell on technicalities at this stage, but stressed how an input made a procedure more flexible than it was before. In other words, the teacher gave the students the ideas which they could use later.

Planning is another activity which is best encouraged away from the machines. Professional programmers probably spend more time with paper and pencil than they do at the keyboard. Students need to be encouraged to plan their work, but there are dangers in over emphasising this. The students are learning to program and so they will need to make mistakes. It is often easier to explore different possibilities at the keyboard. When students learn new techniques they need to practise them before they can use them efficiently, and this is not possible without the computer. We have seen students spending a long time planning a drawing in their exercise books only to reject the plans altogether when things went wrong. However, it is useful for the students to do a quick sketch before they start to type. The teacher should make it clear that the children do not have to be bound by every line of their sketch but that it makes working much easier. It is not uncommon to see two children having a debate because they are working
from different mental images of what it is they want to draw. Their sketch will focus their collaborative activities.

Lesson Format

Organising lessons in which students are working on computers will involve making considerable changes to what might be the normal classroom routine. This is equally true for lessons in a computer room, in a classroom with a limited number of machines or in a situation in which every student has his/her own laptop.

Clearly, learning with personal computers is not compatible with traditional didactic methods of class teaching, but fits naturally into a situation where students are working on their own and/or in groups. Handling a lesson where students are working in this way does present quite different problems to those of talking to the whole class. Many teachers have found that there is a danger of getting caught up with the problems of one group at the beginning of the lesson, so that other groups do not settle down properly. It can be helpful to make a deliberate effort to deal only with immediate problems in, say, the first ten minutes of a lesson, until all student groups are settled and able to get on with their work. Often, the initial problems in any particular lesson will be technical ones which might or might not be dealt with quickly without absorbing too much of the teacher's attention. Getting students settled and working quickly is important in order to establish a good working atmosphere, which will later allow the teacher to spend longer periods giving more concentrated attention to groups who need help and advice. Teachers also find that they need to train students to recognise when the teacher is involved in a discussion with a particular group, so that they do not interrupt as soon as they come across a problem. As the students become more experienced, they realise that they can solve many of their problems without the help of the teacher by talking to other students.

Tracking Student Learning

Much can be gained by the teacher who requires each student to keep a diary of process notes (e.g. Rowe, 1989), questions and descriptions of achievements and problems encountered during computing. The teacher reads these reports regularly and responds. The diaries provide a valuable vehicle for keeping track of progress, and for allowing learning patterns to become visible. Simply describing a problem will often allow a student to understand the task more fully, and thus be able to solve it. Diaries provide direct access to remediation. They enable students to formalise their own thinking, to identify errors and learn from them, and to express their difficulties exactly when they are asking for help.

Student diaries of their computing efforts provide the teacher with a sense of being in charge, of knowing what is going on, and a means of keeping records of student work. They provide a means for maintaining a personal relationship with each student on a daily basis. Since the teachers are usually also still acquiring programming skills, the student diaries can become the source of a sense of comfort. They can show the teacher what he/she needs to learn, i.e. what the teacher's own homework will be.

Beyond this, the student diaries of computing provide both students and the teacher with the assurance that they are part of a collaborative learning experience. The students recognise that the teacher is working with them. Confidence about this alleviates for students and teachers feelings of insecurity or anxiety which might otherwise be present in the initial stages of teaching and learning with computers. Most educators agree that anxiety interferes with learning. As will be discussed in Chapters 6 and 7, feelings of insecurity and computer anxiety were observed even among the more experienced students in the Coombabah project.

Teachers must make sure that back-up copies of all the students' disks are made, to protect the students from work loss due to damage or filing mistakes, and to enable teachers to see the pattern of the work of individual students so they can plan tasks which meet their developmental needs. By examining the work on the back-up disks the teacher can also determine whether a programming problem should be solved with direct assistance by providing a tool which the student may not yet be ready to invent, or whether the student should be encouraged to persevere on his/her own.

Examination of the disks, outside school hours, allows the teacher time to work on programming problems by trying out the programs the student attempted, and trying several plans and solution paths, away from the stress of a class period. Inevitably, examination of the disks forces the teachers to think about their own next learning steps. During 1992 teachers in the Coombabah project examined back-up disks rarely, if at all. They had not set out to evaluate students' progress in computing, but had decided to restrict themselves to assessment of skill and knowledge development in subject domains. As a result of this policy both students and teachers will have missed out on valuable opportunities for learning. Assessment and evaluation are discussed further in Chapter 5.

During the actual class time the teacher walks among the students and observes, admires, comments and answers questions. The teacher can collect small groups of students around a common interest problem. Sometimes he/she will ask a student to share some work with the whole group as a teaching example or model of problem solving.

The words a teacher uses are important. Instead of providing a solution for a student immediately, the teacher might say: Describe the problem. Tell me what happens. What did you want to happen? Try it now and show me, or Teach me what you did. These types of response are important for several reasons. They
give both the teacher and the student time to reflect on the problem, and they require a verbal description on the part of the student. The act of describing something accurately diminishes personal emotion and allows the describer to see more clearly what has actually happened.

If, after the student has described the problem and neither teacher nor student know how to solve it, teacher and student can write a plan together in English words. Such a plan would specify exactly what the student wanted to achieve and perhaps include an example, which, if it is too difficult, can be substituted by a simpler one. Teacher and student together can write a superprocedure before they invent subprocedures, always making very sure that the first step is one the student can solve successfully.

Other tasks for the teacher include collecting, displaying and identifying resources. Many teachers use bulletin boards to stimulate the learning process. The teacher might post a weekly or daily mystery procedure, a new command with definition and examples of use, a challenging programming idea, a template or procedure to copy and try out, or a chart of students’ names indicating their specific areas of expertise in order to make peer tutoring possible for every student. Sometimes the teacher might post an interactive program for students to copy, use, then modify and make their own.

Copying among Students

In the classroom everything students do on their computer is to some extent public. One cannot cover up a screen in the same way as some students hide their written work from one another. When learning with their own computers, it is easy and natural for students to look at, and to comment on, each other’s projects, and so to learn from one another. The teacher can encourage this by giving space for wall displays of children’s work and other stimulus materials. The relationships between students are different from those in a more conventional classroom. The students tend to share ideas and knowledge in an environment which does not continually stress competition.

For some teachers, the thought of students having access to each other’s work in this way may raise the problem of copying. A certain amount of copying, and the right sort of copying, is perfectly healthy. Encouraging students to learn from each other is, in a sense, encouraging them to copy. Some students at Coombabah learnt about variables by copying a program directly from a wall display. Students often copy something that looks attractive, and then make it their own by adding to it or changing it.

Too much copying is unhealthy: it tends to stop students getting involved with their own projects. As with using the printer, students can learn to be more discriminating in the ways in which they use each other’s ideas and procedures. It is important that they are given the time and the guidance to do this constructively. The teacher’s first instinct may be to try to impose his/her own rules about copying, but this will not necessarily achieve the result of helping the students to develop a responsible attitude. Probably the students will still copy, but they will become more devious and copy in less obvious ways.

The fact that the students are permitted to use the computers during breaks and lunch hours, and take them home, brings further benefits but increases the opportunities for copying.

Learning Contracts

This strategy requires students to enter into a contract to carry out certain learning tasks. Some of these tasks might be classroom based, others are carried out outside classroom periods at school, at home or in the community with students performing useful learning and service projects. Learning contracts allow students to become more self-directed and independent in their learning. They can also offer them the chance to explore, learn and practise real life skills in a meaningful context.

The contract approach recognises that there are students with a wide range of both academic and personal/social abilities. By working through individual or group contracts, students are enabled to develop their abilities in relation to their own needs. Negotiation with students about the contents and the types of activities they are going to be involved in is an important aspect of the strategy. Students need to feel responsible for their own learning. There is also an inherent motivation to work when students feel there is an element of choice. There should be negotiation between the teacher and the group, and among individual members of the group. (Queensland Education Department, 1988, p. 8)

In its publication Practical Computer Methods: Guidelines, the Queensland Department of Education (1988) outlines the steps to follow in presenting a contract system to students, and presents sample contracts and projects.

Fostering Problem Solving Skills

When one considers problem solving with and without a computer, it becomes evident that teachers may need to focus on similarities between programming and problem solving without computers. Similar components will have to be made explicit and practised. Those who believe in an isomorphism between programming and general problem solving ability tend to assume that the similarities in processing outweigh the dissimilarities and are sufficient to assure transferability. This assumption is yet to be tested. At present there is little historical, theoretical or empirical support for it.

Some create a one-to-one correspondence between programming and problem solving processes by noting similarities in requirements. For instance, both involve specific directions, planning, hypothesis formation, goal-oriented
behaviour, subgoal decomposition, and means-end analysis, monitoring and evaluation, so that discrepancies between what is obtained and what is intended can be eliminated. Unfortunately, there are also dissimilarities in the two processes which detract from successful transfer. For example, computer programs typically have perfect access to previous information while humans tend to lose information over time. Further, computer programs are rigidly sequenced and, once begun, continue to execute a routine to its conclusion. In contrast, human problem solvers are easily distracted by external stimuli and by ideas unrelated to the problem at hand.

Why are some students unable to cope with the problem solving skills necessary to write Logo programs needing more than three procedures? Burrowes (1985) believes that some students are generally weak in problem solving, have a poor self-image, or perhaps feel defeated by the educational establishment. When attempting to increase students’ problem solving skills, one must also recognise the precursor skill of problem posing. Problem solving emphasises goal directed activity at the expense of exploratory behaviours. Exploratory work with Logo often leads to problem posing, which in turn can result in more goal directed problem solving. Sometimes students move directly into problem solving mode without having really understood what the task might require. Also, cognitive and affective variables cannot be separated. The processes of problem solving, program design and other creative work need like all academic learning to be viewed in the context of the student’s motivation, interests and feelings about computing and the classroom culture.

Collaborative Learning

The term collaborative learning is an umbrella term which covers such activities as learning in pairs or small groups, cooperation and collaboration. Strictly speaking, the meaning of the three terms differs. Small group simply refers to a reduction in the size of groups or dividing the class into groups. Cooperation (an antonym to competition) means to help one another to do whatever is required for the group to succeed (e.g. Slavin, 1985, 1986). Collaboration refers more to the human relationships in the classroom which are expected to help students become more active, autonomous, responsible and self-directed in their learning (Whipple, 1987). Cooperation is, obviously, a prerequisite for collaboration. According to Whipple (1987), Chung (1991) and others, important characteristics of collaborative classroom activities are:

- Teachers and students are active participants in the learning process;
- Collaboration reduces the distance between teacher and students;
- Collaboration creates a sense of community in the classroom;
- Knowledge is created, not transferred;
- Collaboration locates knowledge in the group as well as in individuals.

Johnson & Johnson (1989) point out that the processes of interdependence, interaction and integration must be operating in the classroom if collaborative learning is to be successful. Students must see themselves as positively interdependent so that they take a personal responsibility for their contribution to the achievement of group goals; and they must engage in considerable face-to-face interaction in which they help each other, share resources, give constructive feedback and advice to one other, and be sensitive to feedback from others.

One way in which the teachers at Coombabah foster these processes is through the identification of class experts, who have been coached by the teacher or trained themselves in the use of particular procedures, pieces of software, etc. While this instructional strategy certainly fosters interaction and interdependence between students, it is not well received by all students. Many of them have complained that the same people are selected to be experts, and that most students do not have a hope of becoming experts. These status differences might well be counterproductive for the learning climate of the class as a whole. Collaborative learning and students’ becoming responsible for their own learning are fostered by having students rely more heavily on their peers than on the teacher for solving problems and evaluating outcomes. Students can be encouraged to seek help from peers and only after having done so to ask the teacher for assistance.

During collaborative learning students share, rather than compete for recognition of their efforts. They monitor and evaluate learning processes, rather than hurry to finish tasks quickly. The small group provides safe opportunities for trial and error activities as well as for asking questions and expressing opinions. In small group work more students have a chance to contribute ideas. The group also acts as a motivator and provides students with many opportunities to take on the role of teacher as well as learner. The system of teachers selecting and training experts may not be the best way of fostering truly collaborative learning.

Quite apart from the practicality in situations where there are insufficient machines for each student to have their own, the advantages of computing with one or more partners are great. Discussing ideas with others is an important aspect of any learning situation. Putting ideas into words is not just a prerequisite for such discussions but a valuable way of clarifying what and how one is thinking for oneself. The experience of starting to explain something, only to realise that one has not understood it oneself, is a common one.

Thinking aloud (e.g. Ericsson & Simon, 1984; Rowe, 1985) can be a powerful way of exposing one’s own misunderstandings to oneself and within a small group. It also provides a means of sorting out the confusions by talking them through. This activity is much less threatening, and more meaningful, when the person being talked to is a fellow student, who is not expected to already understand what one is trying to explain, rather than a teacher, who might be expected to know all the answers. Sharing ideas gives the opportunity to learn from others and to see many different views of a problem.
We observed that when students in the Coombabah project were working on Logo, it was natural for them to share their views, problems and achievements with their partners. However, forming stable working relationships is not something that all children take to naturally. The literature suggests that girls may be more comfortable in collaborative work groups than many boys. The literature would suggest that, typically, the girls formed more stable partnerships while the boys tended to change partners more often and were often keen to work on their own. This was found not to be so at Coombabah. Girls did not appear to form more stable partnerships than the boys, nor did boys change their partners more frequently. It was observed, however, that more boys seemed to have a preference for working alone than girls. The class experts often preferred to work on their own until they had worked out a new procedure.

For the teacher to determine the partnerships among students would be missing an opportunity to encourage the students to take control. Sufficient time must be allowed for the students to choose their own partners and to change them as they desire. Also, implicitly teachers might get over the message that working with a partner is useful and important, and occasionally talk more explicitly to students who are having difficulty settling into partnerships.

Inevitably there will be some students who prefer to work on their own some of the time, particularly as they become more experienced as programmers, and it is reasonable for a teacher and peers to respect this. In Coombabah we observed that often pairs of students who are experienced programmers, preferred to work on their own, but chose to be sitting near enough to one another to look at each other’s screens. Although they were working on separate projects, they discussed their work and offered each other help and advice.

**Intervention**

A common misapprehension has grown up, especially in the Logo community, that in order to encourage students to work independently the teacher should not intervene at all. I do not agree with this view, because I regard the teacher’s input as a vital component of the student’s learning. The non-intervention view has resulted from a genuine concern that some teachers might play far too dominant a role in the students’ work, and that the teacher’s intervention can inhibit the students from developing their own understanding. A useful strategy for teachers is to stop and think before intervening, giving the students time to think out their own solution paths and giving the teacher a chance to work out an appropriate reply before jumping in. It is important to spend time just watching and listening when students are working at the keyboard, to find out exactly what they are doing, before deciding whether intervention is necessary or appropriate.

It is often tempting to try to introduce a new command or technique before students are ready for it, or to correct an error rather than suggesting how students could solve the problem for themselves. Actually, students working on their own projects in computing often reject suggestions when they are made, only to come back to them some time later. Students learn new ideas when they are ready for them, and in a large classroom it is hardly possible for the teacher to ascertain the needs of each student at exactly the right time. If the teacher’s interventions are based on the students’ own progress, but without any pressure for ideas to be taken up immediately, then students can make the ideas suggested to them by the teacher their own when they are ready to do so. It is impossible to make hard and fast rules, but the suggestion of never touching the student’s keyboard is a good starting point. Obviously an exception to this would be if there is something wrong with the disk and the teacher wishes to get the student back to work quickly. Even in such a situation the ideal response would be to help the student fix the problem herself/himself by verbal suggestions which the student might try on the keyboard.

**Holding back** from intervention to allow students time to find their own solutions, to develop their own projects and to control their own pace of work is often an important part of the teacher’s role. It takes time for students to become committed to the work they are doing. At the start of a new project students might work at a relatively low level before they are ready to tackle the more difficult aspects. Students with their own laptops and Logo projects are able to learn in a natural way because they have control over what they are working on, and because they can control when they tackle difficult ideas.

**Mistakes.** Another way in which students can take control of their own work in computing is in judging their own success or lack of success. Because students have set their own goals, and usually have instant feedback on their actions from the computer, the need for the teacher to say that something is correct or wrong is much reduced. In fact, a teacher might well be embarrassed when he/she says *That is nice*, commenting on an attractive display on the screen, and is told by the student *No, this isn’t what I wanted to do at all*. Students who are used to having their work *marked* by teachers may take some time to get over regarding errors as wrong and something to feel embarrassed about. They will need encouragement from their teachers to change this feeling. Promoting debugging as a respectable, useful and at least mildly enjoyable activity is one way of developing a positive attitude to making and correcting mistakes. Using the term bugs instead of mistakes is likely to help overcome existing prejudices.

An important feature of Logo and other software is the error messages which appear on the screen if a command is typed which the computer cannot interpret. Unfortunately, these messages are often not read by novices, or they are found difficult to interpret. In the early stages of computer use most error messages tend to reflect typing errors. Learning to read and interpret error messages is an important step for students in learning to program and in taking responsibility for their own work.

**Asking questions.** It is often easier to make interventions in a relaxed way by asking questions. When students are involved in their own projects, it is clear that they themselves know what they are aiming at and that the teacher is on less
familiar ground. This is the reverse of the traditional classroom situation and can lead to noticeable changes in the dialogue between teachers and students.

In traditional classrooms, the purpose of almost all the teachers’ questions is not to get information but to test what students know. Questioning becomes ritualised by both teachers and students and loses its potential as a teaching technique, or even as a normal means of interaction between people. In computing environments the teacher is more often genuinely asking for information (e.g. How did you draw that part? Which procedure draws the eyes? How will you get the turtle back to the right place?) and the students realise this. Questioning becomes the basis of a conversation rather than an interrogation. Asking appropriate questions can be a powerful way in which the teacher encourages students to explore extensions to their projects, and to introduce new challenges.

TEACHING TO FACILITATE INDEPENDENCE

Teachers are able to touch students in many ways. They implement educational policy and curriculum content. Even more importantly, they establish the educational climate, and structure learning experiences. They have almost complete power over the processes which take place in the classroom, and in the final count these processes contribute more to education than does drill and practice. One of the most important goals for the teacher is that of encouraging students to become responsible for their own learning.

Teaching and learning with computers provides the potential for natural human relationships to develop between teacher and students as they collaborate to solve a problem that arises in one of the students’ projects. As noted above, the role of the teacher becomes more like that of a collaborator, and their authority is based on their knowledge and ability to help, rather than on personal and professional status. When the teacher is seen as cooperating in students’ activities, rather than judging them, the relationship between students and teacher can begin to approach the ideal described by Jerome Bruner in Toward a Theory of Instruction:

I would like to suggest that what the teacher must be, to be an effective competence model, is a day-to-day working model with whom to interact. It is not so much that the teacher provides a model to imitate. Rather, it is that the teacher can become part of the student’s internal dialogue - somebody whose respect he wants, someone whose standards he wishes to make his own. (Bruner, 1966, p. 124)

One factor which contributes to this more natural relationship is that the teacher is often talking to students individually, or in small groups, rather than to the class as a whole. This obviously requires a different manner and tone of voice, and it is made easier if the teacher can sit with the students rather than standing over them. Of course this could lead to problems if the teacher’s attention is completely taken up with one group, to the exclusion of the rest of the class. Simple strategies such teachers sitting in such a position that they do not have their back to the rest of the class when talking to a group can help to maintain contact with all students.

What is being said in the classroom and how it is being said, and what students and teachers do in the classroom, greatly affects learning. Certain teacher behaviours have a particularly strong and direct influence not only on student learning and educational achievements, but also on motivation, self-concept, social relationships and how students think about learning itself in and beyond school.

Many of the teacher behaviours which have been shown to invite, enhance and maintain high levels of communication among students can be seen as falling into one or more of the four following categories:

1 Structuring the classroom flexibly to allow for individual, small group and total group interaction as may be required. Managing the resources of time, space, materials, energy, interest and motivation to facilitate participation in the communicative process by all students. Making thinking for oneself and taking charge of one’s own learning an important and acceptable educational objective for each student.

2 Questioning to help students acquire new and access previously acquired information and experiences (input), attaching that information to previously acquired knowledge and restructuring and transforming it into meaningful relationships (processing), and applying what has been acquired, restructured or processed in other ways, in a variety of different, including novel situations (output).

3 Responding to make students aware of their ability to think and learn independently, helping them extend the power of this independence and, above all, maintain their interest in continuing to acquire more effective ways of learning.

4 Modelling the types of behaviours you wish your students to acquire. As part of the day-to-day activities in the classroom, speak about and demonstrate the strategies you yourself use as you go about cognitive tasks, and encourage class discussion about the strategies different people are finding effective (or not effective).

Structuring the Classroom

Structuring the classroom refers to ways in which the physical environment of learning, environmental resources such as time and materials, and human resources such as energy, interest, motivation, etc., are used. Every classroom is structured in one way or another, either consciously or unconsciously, either
directly or indirectly. Even an unstructured classroom imposes a structure to which and within which students and teachers react and interact.

Structuring the classroom for high levels of communication should be conscious, deliberate and clearly based on the desired objectives for the students. Having planned in advance which learning tasks are to be accomplished and what types of interaction are to be achieved, the teacher may wish to state some ground rules, describe and explain the objectives, place limits and constraints, and create an organisational pattern which he/she expects might best accomplish the desired objectives. Research has shown that allowing students to work cooperatively promotes more independent thinking and higher reasoning strategies than do more competitive and individualistic learning situations. In fact, most successful programs designed to develop higher order thinking skills prescribe cooperative learning activities. It was found that the social setting provides occasions for modelling and practice. Skilled thinkers (often the teacher, but sometimes more advanced students) can demonstrate desirable ways of attacking problems, analysing text, or constructing arguments. Students can scaffold complicated performances for each other. Each one does part of the task, and, by working cooperatively, students can arrive at solutions that one student could not manage alone at that particular stage. In addition, mutual criticism during shared work provides the feedback that can help refine or restructure individuals' knowledge and skill.

Teachers who encourage and promote independent thinking and learning tend to provide a classroom climate where:

- the students are able to see themselves as being in the decision making role,
- the students decide on the strategies they will use to accomplish given tasks,
- the students determine the correctness or incorrectness of an answer based on data they themselves are producing and are able to validate,
- the students are involved in setting their own goals and means of assessing the accomplishment of those goals.

Also, the reward system in such classrooms is intrinsic rather than extrinsic (i.e. derived from internal motivation to learn, an intellectual curiosity about phenomena, a striving for competency and accuracy, a sense of responsibility to be a productive member of a community of learners, and a desire to emulate significant respected others).

Questions and Answers

Aspects of teacher questioning which have been found to have significant effects on initiating real communication in class include the types and levels of questions asked, teacher wait-time (i.e. the pause between the end of a teacher's question and the beginning of a student's response, or after a student's response and teacher feedback), and teacher follow-up to student answers.

Dillon (1984) made the distinction between two types of classroom interaction to which he refers as recitation and discussion. Recitation is totally teacher-centred and characterised by recurring sequences of teacher questions and student answers, in which students recite what they have learnt previously or what they are currently learning in response to the teacher's questioning. Real classroom communication, however, involves group interaction in which students discuss what they know as well as what they do not know or understand. More than one point of view is brought forth and considered. The teacher acts as a facilitator by creating a non-threatening atmosphere of equality for students, by providing clarification and guidance, by unobtrusively moving the discussion into desired directions. Capable adept use of discussion is a most important tool for the facilitation of independent thinking and learning.

Students who are working on their own projects can solve many of the problems themselves or through discussion with peers, but sometimes they will demand help from the teacher. A teacher who is less familiar with working with, for example, Logo may feel unsure about the best way to respond to these requests, because there is a tension between wanting to help the students get on with the job, and wanting them to think about solutions themselves. The most appropriate strategy in any particular case will naturally depend on the circumstances. The following examples illustrate this for some typical questions:

*How can one move the turtle without drawing lines?* This type of question is straightforward. The student knows exactly what he/she requires. The knowledge sought is not linked to any conceptual understanding, and there is no way the students could work it out for themselves. There is a Logo command to lift the turtle's pen, and the teacher's response must be to provide this information or to direct the students to a resource, e.g. the handbook, from which they can find out.

*How do we draw a circle?* This request is less straightforward and there are several possible levels of response. Because the teacher's aim is to encourage problem solving and independent learning, he/she would regard this problem as one the students could work out for themselves and encourage them to do so. Pretending to be the turtle would help students to understand the need for short forward movements, alternating with small turns. This involves either implicitly or explicitly refusing to answer the question directly. How a teacher does this will depend on the students involved, and the teacher's personal style.

*How does REPEAT work?* Here the student knows the purpose of REPEAT, but needs to be reminded of its syntactic form. Again, the most appropriate response depends on the student and the circumstances in which the question is posed. A straightforward answer may suffice, as most students will not need much encouragement to explore commands of the form REPEAT 100|FD 100...
Deflecting questions which students should answer for themselves can have positive effects in terms of encouraging independent thinking, but it can also be disconcerting for the students. To them, asking how to draw a circle may not seem any different from asking how to raise the pen, and the teacher's behaviour in answering one question directly but not the other may appear inconsistent and/or obstructive. Some students' answer to the suggestion of walking like a turtle etc., when they have asked for help in drawing circles, might be to make the wheels a different shape. There are countless questions students ask when their work is important to them. The teacher's response may be to give them information, or to create a situation where the students can discover a solution for themselves. What is important is that the students feel free to ask questions.

Positive and Negative Responses

The way in which the teacher responds to and interacts with students in the classroom determines the degree of trust, warmth, openness, rapport and psychological safety in the classroom. It also strongly influences the preparedness of students (especially the more reticent ones) to take risks in trying out new ideas and computing strategies.

Some teacher responses such as criticism (and other ways in which students are put down) and praise result in the termination or temporary closing down of communication with the student. Other responses result in extending or opening the communication process. Examples in the latter category are acceptance, the use of silence, clarification and facilitation.

Criticisms. This is the expression of a negative value judgment. There is an abundance of research evidence to show that criticism does not promote cognitive or affective learning, and that it lowers students' self-esteem and achievement. When a teacher reacts to a student's performance with brief, negative words such as poor or wrong, he/she is likely to terminate interaction with the student as well as the thinking process of the student. More subtle and less negative signals of the inadequacy of a response might be You are almost right, can anyone add to this answer? or You are getting close. Ridicule, sarcasm and other responses which are designed to put the student down should be avoided at all cost. Criticising students and making them feel a failure certainly does not enhance thinking and learning.

Responses which are more useful than criticism in promoting student thinking and learning tend to be the ones which represent extending and opening behaviours. Prominent amongst these are the appropriate and not too frequent use of praise, large amounts of acceptance, clarification, facilitation and the skilled use of silence.

Praise. Praise might be seen as the opposite of criticism. It manifests itself in the expression of positive value judgments such as good, excellent, very useful, great, etc. While most educators strongly advocate the use of praise to reinforce desired behaviours and for the building of self-esteem in students, there are some problems related to the indiscriminate use of praise. Praise builds conformity and it is thus less useful where our goal is diversity. The effect of praise on some students is to make them dependent on others for their feelings of self-worth rather than on themselves. Also, praise, like criticism, tends to terminate the interaction between student and teacher. It is important for teachers to recognise this, and use praise judiciously with those students and in relation to those objectives for which it is suitable (e.g. with reluctant and unmotivated students, young children in rote learning of low level cognitive tasks).

The teacher's long-term goal should be to decrease the use of terminal behaviours including praise and criticism. Teachers can replace their habit of offering praise too frequently with an enlarged repertoire of response behaviours which have been shown to be more conducive to developing student thinking, learning and self-esteem. Amongst these are acceptance of a variety of views, clarification, facilitation and silence.

Acceptance. This term describes responses which are non-judgmental and non-evaluative. Neither words nor gesture, posture, etc., give clues as to whether the teacher regards the student's response or idea as correct, good, bad, worse, better, etc. Alternative ways of reacting to a student's answer are by acknowledging it, paraphrasing or summarising it, applying it or comparing it with another idea.

The intention of acceptance is to build a psychologically safe classroom climate in which students can take risks, feel that they are entrusted with the responsibility of making decisions and can explore the consequences of their own actions. An atmosphere of acceptance encourages students to examine and compare their own views, feelings, reactions, values and criteria of success with those of other students as well as with those of the teacher. Even when students' views, feelings, etc. differ from those of the teacher in a seemingly unacceptable way, the teacher can still accept them temporarily, because he/she realises that only the student is able to modify them. The task of the teacher is to provide information and to guide the discussion in the classroom in such a way that it facilitates the processes which lead to students' modifying their feelings, criteria for action, etc. making them more consistent with reality and the demands of the task or the situation.

The classroom where independent thinking and learning are fostered is one where a spirit of inquiry prevails. Student questions and intellectual challenges are valued. The teacher admits uncertainty: We are not really sure how evolution...
comes about, I am not sure about my interpretation of this poem, I continue to find new things in it. In this way the teacher emphasises education as an exploration of the unknown, as well as learning what is known.

Teacher acceptance encourages problem finding on the part of students. In many classrooms quick answers and solutions are sought, encouraged and valued. In an independence oriented classroom, students are taught and encouraged to identify problems, to wonder and to speculate. The unthinking person may observe graffiti and either smile or frown. The thinking person wonders why in Europe graffiti is so often political while in the USA it is more commonly scatological and in Australia childish. The teacher nurtures a problem finding disposition by encouraging students to ask questions of their own, not just answering the questions posed by others. Here are some data about income distribution in Australia, what questions could we ask? We'll be looking at the role of the nuclear family in Aboriginal communities, what questions would you like to have answered? Note that acceptance can be demonstrated in different ways, e.g. it can be quite passive, active or even empathic.

Passive acceptance refers to instances in which the teacher merely receives and acknowledges what the student says, without making any value judgment. It shows the student that his/her response has been heard. Examples of passive acceptance behaviours teachers can use are: That is one possibility, I understand, Could be, Hmmm. Non-verbal passive acceptance behaviours include nodding of the head or writing the student's statement on the blackboard.

Active acceptance refers to instances in which the teacher demonstrates an understanding of the student's response. The teacher actively accepts by reflecting (not merely repeating), extending, building on, comparing or giving an example based on the student's response. While rewording the student's response, the teacher strives to maintain the intent and accurate meaning of the student's idea. Active acceptance is stronger than passive acceptance because the teacher not only acknowledges that the student's message has been received, but also that the intent of the message has been understood.

Empathic acceptance is an acceptance of feelings as well as the products of thought. It means that the teacher not only hears the student's ideas but is also sensitive to the emotions underlying or accompanying these. Teachers can show empathy when they express similar feelings to those of the student from their own experience. Empathic acceptance does not mean that the teacher condones acts of aggression or destructive behaviour. Rather, it demonstrates an understanding and acceptance of the emotions that produce such behaviours.

Clarification This is similar to active acceptance in that both behaviours are concerned with the teacher's understanding of the student's idea. While active acceptance conveys that the teacher understands, questions of clarification convey that the teacher is seeking understanding but requires more information. Nearly 30 years ago, Flanders showed that student achievement is higher in classes where teachers use, build on, extend or clarify students' responses. When teachers encourage students to elaborate on their answers and use other methods of clarification, students tend to increase the consistency of their thinking, i.e. they become more task oriented and purposeful in their problem solving and learning.

One of the most compelling reasons why teachers should make frequent use of clarification is that it contributes to the development of students' metacognitive skills. There is a high correlation between the degree of metacognitive awareness and the level of performance on complex problem solving tasks. Students become better problem solvers and learners if they are able to become aware of and talk about the strategies and steps they use in their problem solving and learning.

Some students follow computing instructions and perform tasks without asking themselves why they are doing what they are doing. They seldomly, if ever, evaluate their own learning strategies or the efficiency of their own performance. They have virtually no idea what they are doing when they perform a task, and are thus unable to explain the strategies and steps they used. When the teacher asks students to explain their work, i.e. to show how they arrived at a solution or to share their rationale for a certain procedure, the teacher causes the student to use metacognition. For 40 years research evidence has been building up for the view that thinking and talking about thinking leads to more thinking. Causing students to talk about their thinking and learning processes during and after performance enhances their ability to think.

Facilitation Facilitating the acquisition of information, knowledge and skills is a basic aim of teaching. To do this the teacher must be sensitive to and able to perceive students' needs, provide information and make it possible for students to do so themselves. Knowledge of results, i.e. feedback, is the most important variable governing the acquisition of skill, but also in the development of independent learning dispositions and habits of thinking. Note that there is a difference between rewards and feedback. Rewards can either control behaviour or give information about competence. If students perceive the teacher's praise as controlling, their intrinsic motivation is likely to decrease. If, however, students perceive rewards as providing feedback about their skill or competence, intrinsic motivation is likely to increase.

Silence Silence and waiting time are important aspects of teacher/student interaction. Many classrooms could do with a more deliberate pace rather than encouraging impulsiveness. The teacher asks a question, expects an immediate answer and calls on the first student who puts up his/her hand. Such rapid-fire recitations can be useful in several ways. They facilitate the assessment of a single student's knowledge, permit rehearsal of facts and keep students attentive. However, if the aim is to develop problem solving and thinking, as when learning with computers, this style of interaction is counterproductive. Students need time
to deliberate, i.e. to reflect about alternative possibilities, to weigh the evidence and to come to a tentative conclusion.

Research has shown differences in student behaviour in traditional classrooms where teachers waited after asking a question or after a student responded. Teachers who wait just a short time, i.e. one or two seconds, tend to receive from their students short, often one word, responses. Teachers who wait for longer periods tend to elicit responses based on more complete thought, and in whole sentences. Also, there is an increase in the creativity of responses as shown by the more frequent use of descriptive and modifying words, and an increased speculativeness in students’ thinking. Interactions among students in the group are increased, the number of questions students ask increases and, most important of all, reticent, shy and slower thinking students begin to contribute. The same processes are likely to operate in the classroom where students learn with computers.

Teachers can communicate their expectations to students through the use of silence. Teachers who ask questions and then wait before they invite a student to answer show that they expect an answer but also that they have faith in the student’s ability to answer, given sufficient time. Teachers who ask a question of a student, wait only a short time, and then give the answer, call on another student or give a hint, only demonstrate to the student their belief that he/she is really unable to answer the question, i.e. is considered too poor a student to offer an answer or to reason independently.

Modelling

How do students choose their models and how do they know what to emulate? Teachers and some class experts provide immediate models. In addition, however, students need to be presented with models and case studies of successful experts in computing and other domains. Students can learn a great deal through the direct observation and the study of detailed accounts of peers and adults (including their teachers) struggling with problems. This allows them to observe, discuss and understand why certain processes are operating differently in different situations and with respect to different tasks. They will learn that creative thinking and independent learning habits are not limited to a given age group, to certain occupations/professions, to particular ethnic groups or social classes, or to scientific and other scholarly effort. They need to observe that the best thinkers, including their teachers, more experienced students, parents, and significant others can be wrong, and that the path to success is often uncertain and may be full of torture.

Modelling by the teacher serves to share with students not only what the teacher might be thinking about the content to be learnt, but perhaps more importantly, information and feelings about the processes of learning and problem solving.

CONCLUSION

The implications of teaching to facilitate independent thinking, problem solving and self-regulated learning, with and without computers, include the enhancement of communication processes between peers, and between students and teachers, an enrichment of teacher conceptions of individual differences among students in motivation, interest, learning style and information processing, previous experience and behaviours, an improvement of instructional methods, and the broadening of educational goals and outcomes. This type of teaching leads to new kinds of educational structuring which extend beyond the boundaries of the classroom and school. The teaching and learning advocated in this chapter is characterised by shared knowledge among teachers and learners, students and teachers sharing planning and control with respect to the content and processes of learning, and students accepting personal responsibility for their learning as members of a community of inquiry and learning.
Assessment and Evaluation

ASSESSMENT VERSUS EVALUATION

In educational contexts, assessment, as an aspect of evaluation, if not a prerequisite for it, involves gathering and transmitting information which is relevant to and assists in making certain kinds of decisions.

The Encyclopaedia of Educational Evaluation (Anderson, Ball, Murphy & Associates, 1975) defines assessment as a process for gathering information which can meet a variety of evaluation needs. The process of assessment involves multiple indicators and sources of evidence, and in this sense is different from testing.

Assessment, as opposed to simple one dimensional measurement, is frequently described as multtrait-multimethod; that is, it focuses upon a number of variables judged to be important and utilizes a number of techniques to assay them ... Its techniques may also be multisource ... and/or multijudge. (Anderson et al., 1975, p. 27)

Assessment is the process of collecting and organising information or data in ways that make it possible to judge or evaluate performance, the operation of a program, etc. Assessment data of any kind are no more than indicators of a phenomenon. The evidence associated with such indicators must be unambiguous to the extent that the context and means of its collection are understood by the students, parents, other teachers and whoever else might wish to use it. Assessment data provide evidence, but the evaluation of that evidence can still be open to interpretation as different people might form somewhat different judgments concerning the implications of the data.

It therefore seems appropriate ... to limit the term assessment to the process of gathering data and fashioning them into an interpretable form; judgments can then be made ... Assessment, then, as we define it, precedes the final decision making stage in evaluation. (Anderson et al., 1975, p. 27)

Hayman, Rayder, Stenner & Madey (1979) suggest that for assessment to be used effectively, four criteria must be met:

1. The information resulting from the assessment must reduce uncertainty. If uncertainty is not reduced, no information is provided by the assessment.
2. A proper format must be chosen for representing, transmitting, receiving and relating assessment information.
3. The recipient must understand the meaning of the information. In other words, the information must be accessible to the intended user.
4. The information must be capable of motivating human action.

IMPORTANT ISSUES

The goal of introducing computers into the classroom is to assist with the intellectual development of students. Assessment and evaluation of student progress is an integral part of this process of development. Evaluation is needed not only to judge the intellectual and personal development of the students, but also to determine the effectiveness and validity of curriculum content, classroom organisation and the teaching strategies used.

How can learning processes and educational outcomes in computing be evaluated? At present, there appears to be some disagreement and uncertainty about what is to be expected from students who are learning with computers. This uncertainty is the reason for the current lack of agreed curriculum guidelines and criteria for student assessment. Individual teachers thus need to clarify for themselves what they expect to achieve through the use of a computer and the computer-based activities in the classroom. In learning with computers, as in all other areas of education, assessment is a positive aid to learning and instruction.

The important components which need to be evaluated fall into six broad categories: computing skills, knowledge, awareness, attitude, learning processes and learning outcomes.

Assessment should have both formative and summative aspects. The process skills used by students to complete tasks are at least as important as the final product of an assignment. Learning outcomes may show whether a student has acquired certain facts, rules and procedures. However, measures of learning outcomes do not provide any information about the way in which the assessed knowledge and skills have been acquired, nor do they tell us what prevented a certain individual from developing the required knowledge or skills. The major purpose of formative assessment is to provide diagnostic and remedial feedback to the student; in addition it will provide information which will update the student's profile of skill and knowledge development for the teacher. Assessment should be concerned with the qualitative aspects of the learning process as well as the final outcome.

Some aspects of computer awareness can be assessed through written or oral procedures, even by means of multiple choice tests. Student attitudes to computing can be assessed by talking with individuals or groups of students and through questionnaires. However, the physical handling of the hardware and
software, and computer programming, are best evaluated through the observation of performance and the evaluation of the workability, efficiency and effectiveness of the final outcome or product. The most revealing way for students to demonstrate their abilities and skills is through practical application. This encourages them to display their knowledge and skills in the planning, design, creation and appraisal of solutions and other cognitive outcomes. This also allows the observation of student attitudes, learning styles and other personal characteristics, their ability to concentrate, persevere and apply themselves to tasks, their capacity to work with others and on their own.

In classrooms where students are working with computers, we are likely to see a lot of group activity and discussion. Students move about the room to watch other students, to ask for assistance, to debate and argue their view. Students gain valuable insights through listening to the comments of others about their work and ways in which it could be improved. Pieces of work at various stages of completion are evident in the classroom. Because students may work on individual or group projects for hours, days or weeks, they are engaged in a variety of activities as they experiment, develop ideas, test new theories, plan and draft solutions, revise and refine their work. Through these activities, students show their understanding, difficulties, and their developing knowledge and skills. Self-evaluation (written or oral) is an excellent method of enabling students to reflect on their learning.

As with other areas of learning, how computing knowledge and skills develop should be monitored regularly and as an integral part of classroom learning and teaching. Much information can be gained quite quickly by the teacher by simply talking to the students about their work, what they have learnt, enjoyed or disliked. At this time teachers would also discuss with the student how they think the student is working, decide whether quality and quantity of work is of an agreed standard (and if not, to try and find a solution with the student to overcome the problem). Assessment which occurs within the context of learning has a certain naturalness about it. Realistic competence is displayed by the students as members of a community of learners, in contrast to the decontextualised, isolated, and competitive character of testing as it is often carried out. As members of groups and the broader classroom community, students learn to assess their own performance and development, and they come to rely on group feedback and peer commentary about their ideas and work strategies (in addition to feedback from the teacher) as forms of assessment. They learn to understand the importance of the part individuals play in group work, and they learn to evaluate their own contribution and that of others. In these contexts, students are able to monitor their performances and observe the performances of more competent as well as less competent peers more consistently than is possible in situations where learning and problem solving proceed individually and silently, and where all that matters is the answer or end product.

As noted above, assessment and evaluation need to be formative and summative, but above all they need to be continuous. A balanced approach to the evaluation of learning in all areas of the curriculum requires the collection of many kinds of evidence over a long period of time, i.e. the whole term or school year. Assessment should identify strengths and weaknesses, and point to how learning processes and learning outcomes of students can be improved (cf. criterion 4 noted above). Assessment includes describing and monitoring student progress, and making summary statements of achievements in terms of both learning processes and outcomes. Both of these should be based on previously determined goals and on specified work requirements. Judgments of capabilities could then be guided by the outcomes described for each band of schooling (e.g. lower primary, upper primary, lower secondary and upper secondary).

Over the past five years, most State Education Departments in Australia have produced charts listing goals for attainment in educational computing. At present, there is little communality between these charts and no followup as to the usefulness of the goals themselves is available. Having declared technology education as one of the eight mainstream areas of learning in the Hobart Declaration in April 1989, the Australian Education Council has mounted a major National Technology Education Project. The resulting National Statement on Technology Education for Australian Schools is expected to include detailed statements of goals and expected achievements. The document should be released in the near future. In the meantime, the following generally agreed upon broad aims might serve as initial expected outcomes for primary and lower secondary school students.

**EXPECTED OUTCOMES**

Two quite basic outcomes of educational computing can be expected in primary and lower secondary school classrooms:

1. Students will come to feel confident and comfortable about using the computer as a learning tool; and
2. They will use this tool regularly across the curriculum to achieve learning objectives and to solve problems in the context of their daily classroom activities.
Indicators which could serve as evaluative measures for (1), i.e. to assess the extent to which students are confident and comfortable in using computers as tools, might be as follows:

1-0 student does not use the computer;
1-1 student uses computer only when directed and for limited purposes, e.g. drill and practice;
1-2 student uses computer regularly with guidance from the teacher;
1-3 student confidently uses software or procedures written by others in various domains and for various purposes;
1-4 student is able to match a particular software application to his/her specific need;
1-5 student takes initiative to use computer appropriately;
1-6 student explores computer use for divergent purposes.

Indicators for (2), i.e. to assess the extent to which students regularly use computers to solve problems across the curriculum, might be:

2-0 no problem solving use of the computer;
2-1 attempts to use computer for problem solving in one or more areas of the curriculum;
2-2 computer used regularly for problem solving activities in a particular area of the curriculum;
2-3 student displays an ability to select a software application or module which matches his/her problem;
2-4 student uses subject specific problem solving procedures in mathematics, science, social studies, language, etc., and is able to help peers with this;
2-5 student tries to use computer for problem solving across the curriculum, e.g. attempts integrated use of applications such as word processing, graphics design, databases, writing of procedures, information processing (i.e. retrieval, analysis and presentation);
2-6 student uses computer across the curriculum to process information by organising, manipulating, analysing and synthesising information, and is able to help peers.

MONITORING PRODUCT AND PROCESS

In addition to the above discussed broad outcomes of learning with computers, there are more difficult issues of assessment and evaluation. Key questions include:

1 How can we assess or evaluate the outcomes of work in educational computing? What are the criteria for assessment?
2 How can we assess the processes of thinking and problem solving taking place during computing, rather than facts and final solutions?
3 What is the final solution if learners are working with programs and in media which invite them to return and encourage further editing?
4 How do we assess the work of individuals when the work is collaborative?

With respect to questions 1 and 2, it is obvious that all learning and teaching must include monitoring. Further curriculum preparation and planning depends on the results of the ongoing monitoring of what has been covered. The techniques outlined later in this chapter will be applicable for courses in computing as a subject as well as for the evaluation of the development of computing knowledge and skills in integrated courses.

There are many factors to be considered in monitoring both a course in computing studies and the use of computers as a tool integrated in subject areas. In both cases monitoring helps ensure that the requirements of the syllabus are met. Objectives, perspectives, content, approaches to planning units of work and different teaching techniques must all be considered.

In the absence of a formal curriculum statement and syllabus, teachers will be setting their own goals and devising suitable learning sequences for their students. These will relate to the broad areas of computer awareness and attitudes to computing, knowledge about computers and computing, and computing skills. Whether computing is taught as a separate subject or is integrated into other subject areas, the processes involved in learning computing and in learning by means of computing include the following:

- analysis of the topic or task,
- comprehension of what is required,
- finding out facts, rules, etc., which might apply,
- planning,
- preparation and presentation of information,
- use of software and hardware,
- design of solution,
- execution of task,
- self-monitoring and evaluation,
- sensitivity to feedback.

The problem of monitoring is not so much a matter of evaluating what is done but rather with finding methods of keeping track of what is happening as learning and teaching evolve. Whether a curriculum is written down in its entirety before teaching any part of it, or developed as teaching proceeds, there must be careful documentation and monitoring of all the parts.
Different teachers will have different strategies to monitor progress throughout their teaching. Some will prefer to have pages in a book, or computer records that allow portability, others wall charts of various sizes so that the whole course can be seen at a glance. This is a matter for individual preference. The only necessity is that there is monitoring. A later section of this chapter entitled Evaluating Progress provides specific suggestions as to how such monitoring could be accomplished.

With respect to questions 3 and 4, concern has been expressed (e.g. Heppell, 1989) that we have not yet fully appreciated the significance of the non-linearity of working with computers. In particular, it could be argued that we have not yet developed a language for the use of such non-linearity. In our attempts to evaluate the work of students who use computers we behave as though we are evaluating the outcomes of working with pen and paper technology. We look for originality, for first drafts, and for work that has been completed or preplanned. Unfortunately these terms are less useful in the computing environment, because they prevent us from actually making use of the freedom provided by the medium.

The facility to return to a piece of work, to re-use, build upon and modify previously produced material is a major benefit of the computer. It is not just labour saving in the conventional sense of time and physical effort, but it allows all learners to build upon a store of personal as well as collective past experiences and ideas. This is an ability that experienced learners have in abundance. The reworking of ideas and knowledge is an essential part of Bruner's (1966) spiral curriculum. Whereas linear curriculum models forge limited links between pieces of knowledge, engendering perceptions of completion and closure, in the spiral curriculum knowledge has multiple links and is constantly growing in an associative network and is therefore more readily available to use.

Should we reward students who use all the search, sort and graphical tools at their disposal, or those who ask good questions and achieve relevant and useful answers? Even here we are concerned with summative evaluation, the end-product of the learning experience, rather than formative evaluation of the process of learning. A Logo case study described by Hoyles, Sutherland & Evans (1986) shows how we can be misled by an evaluation which is restricted to the end-product. It reports the work of pairs of children over an extended period of time. While most pairs in the class had both successes and setbacks, two boys continuously produced exciting visual patterns using recursive procedures. Further analysis showed, however, that the complex patterns of one week varied little in structure from those of previous weeks, even though the patterns were often visually dramatically different. The boys were manipulating variables in a procedure by handle turning. What is even more disturbing is that they had borrowed the initial procedure that was being manipulated from a neighbouring group in the first week of the project. Summative evaluation of their work marked them as being highly successful, but more careful evaluation of the processes they were using revealed that the two students were stuck in their own procedural loop, trapped by initial success and now unwilling or unable to experiment.

Heppell (1989) argued for a technological solution to assessment in which the computer would keep a log of the investigative strategies of the learners. What is suggested is not a skills tick list so much as a monitoring of an ongoing process. This method of monitoring could go a long way towards forcing students and teachers to focus on process, although the logged data would still need very time-consuming evaluation. In some instances the computer can be set up to manage simple instruction, e.g. to keep track of students' performances on drill and practice activities. Other computers are set up for diagnostic testing. The hoped-for outcome of these measures is to free the teacher for more essential work.

Computer-based activity of the more open-ended variety can provide teachers with new insights into what their students can do. Anecdotal accounts have for a long time described how teachers have learnt new things about their students' capabilities as a result of observing them interacting with the computer and peers (e.g. Burns, Cook, Dubitsky, 1982; Papert, Watt, DiSessa & Weir, 1979).

With a greater emphasis on skills of abstraction and comprehension, what student achievement consists of and how it is measured will need to change (Fredericksen, 1984). For example, the advent of the pocket calculator has meant that mathematical operations and estimation can be emphasised over calculation. Word processors have resulted in a new emphasis on the writing process, as opposed to spelling and the forming of letters. Programming will allow the observation of students' planning, monitoring, problem solving and decision making skills.

Determining whether a student is a good problem solver who can imagine multiple solutions, plan solution strategies, and estimate outcomes is very different from counting how many problems a student can answer correctly in a given time. A composition may no longer be judged simply by the number of spelling and grammatical errors it contains.

Intelligent computer systems, which are presently being developed, will make it possible in the future to promote and diagnose student performances in new ways. Based on the student's performance, these systems might prompt the student to reconsider an answer, demonstrate a different process for solving a particular problem, or ask the student to indicate why he/she thought a particular response was correct. Other types of intelligent systems might help teachers understand how students learn and solve problem by analysing students' errors (e.g. Burton, 1981; Ohlsson, 1986).

GROUP WORK

Learning with computers usually involves a considerable amount of group work. Formative as well as summative evaluation of group work is important. Assessment at various stages of a group project provides the teacher with more
opportunity to observe the contribution of individuals to the group effort than assessment of the final product can.

In the assessment of performance on tasks tackled by collaborating students the same mark might be given to each of the students in the group. Where this is unsatisfactory, the contribution and achievement of individuals to the group effort could be estimated as well. For example, a common mark might be given for the product of the group's work, but in addition individual members of the group might be awarded marks above or below this group mark, according to the level of their contribution. The assessment of the contribution of individuals is judged on the basis of observation by the teacher, group discussion and subsequent peer assessment, or a combination of both. Our research in the SUNRISE classes at Coombabah showed peer assessment to be objective and highly consistent with teachers' judgments. The latter observation is supported by the literature relating to peer assessment.

Some criteria for the assessment of the processes of group work might be provided by answers to the following questions:

- Did everyone contribute?
- Was there a group leader?
- Did different aspects of the task emerge?
- Did everyone have an opportunity to contribute?
- Did everyone choose to contribute?
- Did students take turns using the keyboard?
- Did the students select one keyboard operator?
- Was there evidence of group planning and decision making?
- Did the students vote on how to proceed?
- Did some students lose interest or show frustration?
- How did the students evaluate their work?

It is also possible to focus on the performance of individual students within groups:

- Did Student X contribute to the whole group's activity or task?
- Did Student X show that he/she is able to listen to others as well as talk?
- Did the student share in the more difficult as well as in the easy aspects of the task?
- Did the student show self-motivation?
- Did the student accept responsibility for his/her own suggestions, actions, etc.?
- Does the student perceive the project to be his/hers?

DIFFERENT ASSESSMENT PROCEDURES

Standardised tests of computing are not currently available. Even if such tests became available, they would be of limited value for the assessment of process skills. More suitable alternative methods of assessment include direct observation, screening procedures, informal interviews, structured interviews, situational try-outs, work samples and analyses of tasks and learning contexts. Each has advantages and disadvantages, and requires differential time commitments and varying levels of expertise on the part of the teacher utilising the method.

Direct observation can be spontaneous or systematic. During spontaneous or non-systematic observation, the teacher makes notes of behaviours that appear important at the time. Systematic observation focuses on one or more previously specified behaviours. These target behaviours are operationally defined and then counted in terms of frequency, magnitude or duration. The latter procedure is used extensively in applied behaviour analysis and for purposes of behaviour modification.

Screening procedures consist of short and easily administered inventories, questionnaires, check lists or rating scales which provide some initial information about the characteristics, i.e. attitudes, feelings, knowledge, or skills, of an individual student or a group in relation to a variety of learning topics or areas. Screening instruments are often constructed by teachers to cover a particular context of learning or student behaviour. Screening can be quite efficient but the accuracy of the information obtained in this way can be unreliable.

Informal interviews have much in common with direct observation. The teacher asks questions and discusses the variables to be assessed with the student, produces notes and might make a report of the interview.

Structured interviews require considerable preparation and planning in advance. The areas focused on may be general or quite specific. The success of this approach depends on the skill of the interviewer in choosing and correctly phrasing the relevant questions, and the ability of the person being interviewed to understand the question and provide answers. An advantage of structured interviews is that they assure that every student is asked the same questions in the same sequence. This facilitates comparison between the responses and reactions of different students and helps to provide a more valid picture of the performance, knowledge, attitudes, etc., of the class as a whole. Structured student interviews are particularly valuable when the teacher's major aim is to evaluate a teaching program or module.

Situational try-outs are based on the assumption that the best estimate of an individual's skills and abilities comes from direct observation of that individual in specific learning or problem solving situations, rather than from a test or indirect report showing what has been learnt in the past. Situational try-outs can provide process as well as outcome information. Peer teaching is the suitable type of situational try-out in the classroom. Students take turns at being the teacher. A
Leaming with Personal Computers: Issues, Observations and Perspectives

considerable amount of information regarding the student's knowledge about how to learn, practise or go about a certain problem is made explicit by this method. Asking more experienced students to explain to their less experienced peers how they would handle a variety of computing tasks forces these experts to make explicit what they know and how they go about their work. They could also be asked to teach less efficient programmers to produce more effective programs. In devising their instructional plans, students should probably be advised by their teachers. By observing students' questions, learning processes and solution paths, valuable insights into the processes leading to their performance outcomes can be gained. Obviously, after this type of peer teaching adequate debriefing compensating for poor instruction must be provided by the teacher.

Work samples are easily obtained from the students or their parents. The information sought in this method relates to the number, types and patterns of errors as well as successes. Work sample information can be used to develop and modify learning procedures which can correct or circumvent the difficulties experienced by the student. Analyses of tasks and learning contexts focus on the demands made of the student rather than on the student's skills and personal characteristics. Task analysis identifies the major component skills and appropriate sequencing required to complete a given task. An analysis of the learning context is applicable to any classroom, social, leisure or family setting. It aims to identify the most important characteristics and components in a situation, i.e. the major demands, stresses, obstacles or barriers it might produce for a particular student.

In summary, the following means of monitoring student progress in computing as well as in particular subject areas have been found useful by teachers:

- work records, work samples and folders of completed work;
- student diaries or journals;
- practical performance tasks;
- observation and interviews;
- observation of peer teaching;
- group and peer evaluation;
- checklists;
- learning contracts.

PROGRAMMING RELATED EVALUATION

Many methods can be used for the evaluation of a student's or group of students' planning of a computer program. For example, all activities attempted might be listed by the student or students in an activity book. The teacher meets with individuals or groups of students on a regular basis to comment on, accept or ask for revision of the work as it is presented by the students.

Although in computing assignments and computer-based project work the students might be free to choose the order in which they complete activities, it should be made clear, early in the school year or term, that they will need to justify the time that is spent on each activity. Not all students might be expected to complete every activity, but they need to be encouraged to select those activities they choose to complete and then persevere with their selection.

Sources of information which can provide evidence of student performance and achievements include:

- student diaries of thoughts, plans and descriptions;
- descriptions and analyses of techniques used;
- sketches of ideas and products, designs and plans;
- listings of information sources used, with justifications for their selection;
- working models and their modifications;
- errors made and their correction;
- protocols produced from recordings of discussions between students, thinking aloud and interviews;
- videotaped records of problem solving activities;

The collection of exemplar, not only of students' programming efforts but also to accompany the student profiles and attainment levels, will assist in comparing the performances of students at different points of learning. Exemplar of students' work can also help teachers who are novices to educational computing to validate their expectations, and they are crucial if we want to avoid a narrowing of the curriculum content and of teaching methods. Exemplar can provide directions leading to more reliable methods assessment and intervention than, for example, statewide testing programs. Exemplar can take various forms, such as

- written case studies, showing a range of a particular student's work and his/her development of learning over time;
- portfolios illustrating the work of a number of individual students on the same topic;
- videos, portraying various learning activities.

TEACHER RECORDS ON STUDENT LEARNING

Having documented information relating to the strengths and weaknesses of individual students, their progress in knowledge and skills in computing and subject areas, and characteristics of cognitive and personal development, the teacher is in a position to develop comprehensive individual profiles for all students in the class. These profiles include quantitative as well as qualitative information, gathered by both formal and informal methods, on the performance and learning processes of individual students.
Table 5.1 Class Record of Informal Comments

<table>
<thead>
<tr>
<th>Week Ending</th>
<th>2.3</th>
<th>9.3</th>
<th>16.3</th>
<th>23.3</th>
<th>30.3</th>
<th>6.4</th>
<th>13.4</th>
<th>20.4</th>
<th>27.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frank A.</td>
<td>++</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td>++</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mary A.</td>
<td>+</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td>++</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ivan A.</td>
<td>+</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ann O.</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Michelle E.</td>
<td>++</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Record sheets for informal comments. Quick, anecdotal comments resulting from observations of individual students can be recorded on file cards, accompanying a classlist on a large sheet of paper, or in a computer file containing a classlist and individual students’ records. As all the names are on the same list, it is easy to quickly identify the students who have not been observed. The system might be set up in such a way that there should be at least one comment on each child each week. Comments may be both behavioural and conceptual. An example is shown in Table 5.1. The plus signs (+) indicate the number of comments made on the student in the particular week. Table 5.2 contains examples of records of informal comments on individual students.

Objectives

Whether to fulfil the requirements of a course or to assess whether students have learnt what they were expected to learn, certain objectives must be formulated. Where computing is taught as an integrated course with content areas, the objectives of learning computing should, ideally, be brought together with the objectives of the subject area.

A variety of ways could be used to monitor which objectives have been addressed as teaching proceeds. One way is to state which objectives are being addressed within each unit or topic of work. Another is to list all of the computing objectives, leaving space to write in which units of work have addressed them. Obviously, many objectives will be addressed on many occasions.

Such listings of objectives could be in the form of a table. Alternatively, an objectives wheel, as shown in Figure 5.1, could be used. The completed wheel contains all the learning objectives near the circumference. The learning objectives which have been addressed can be indicated for each unit of work. This type of record makes it easy to see whether any objectives are being neglected and gives a clear indication of how they are all being addressed. Obviously, the number of wheels required will depend on the number of objectives and the number of units of work.

Objectives wheels could be developed for computing knowledge and skills in combination with different subject domains, such as mathematics, social studies, language, etc. It is possible to place one or more subject areas on one wheel depending on the choice of subjects, or else a number of different wheels could be developed for sets of units within each subject area. A blank wheel is provided for duplication in Figure 5.2.

Helping students to become proficient and knowledgeable computer users requires that a balance between practical skills and knowledge about computers and computing be maintained. The actual numbers of objectives covering different areas will be determined by the orientation of the course or syllabus. Obviously, more class time might be spent on some objectives than others. For example, write procedures may take longer to achieve than use a range of computer equipment safely in the classroom or identify the components of a typical system.
UNIT OF WORK

Figure 5.1 Objectives Wheel

Figure 5.2 Empty Objectives Wheel
While all objectives need to be addressed, many will be taught by concentrating on others. It is only necessary to ensure that all are addressed, not that each one has a specific activity related just to it, or that all are given an equal time allocation.

Each time a new unit of work is prepared, it is essential that teachers identify what objectives will be addressed so as to monitor progress towards satisfying the learning objectives.

Tools

Monitoring the use of basic tools available through computing is simple but important. Most teachers' learning objectives would suggest that all should be used. It is easy to neglect one or more tools or to overemphasise one unless there is some formal means of recording the attention given to them. A grid can be drawn up as shown in Table 5.3, and an indication of the unit, part of the unit, or subject area which uses the tool can be noted on such a grid.

Table 5.3 Record of Tool Use

<table>
<thead>
<tr>
<th>Unit</th>
<th>Databases</th>
<th>Spreadsheet</th>
<th>Word Processor</th>
<th>Dictionary</th>
<th>Simulation</th>
<th>Programming</th>
<th>Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td>History</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social Studies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maths.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Science</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spelling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Writing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reading</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Music</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As units of work are developed it is important to record the content that has been covered in such a way that it is easy to identify what remains to be done. This will then assist the development of further units of work. One way is to keep a historical record of the content of a particular unit of work. This is particularly necessary when units cut across multiple topic areas and include parts of the core knowledge and skills of computing. Again a chart would accomplish this.

Listing all the content to be taught and then indicating when it has been presented is another way of monitoring learning objectives. For this purpose a large wall chart, a series of charts or a computer file which is accessible to the students as a read only file could be used. An indication of the type of the unit of work could be used beside each tool. This would help to show the links between different subjects. Alternatively, or as well, the date on which the unit was taught could be indicated. Such charts could also be used as devices for planning.

EVALUATION

Evaluation is concerned with educational improvement in the areas of programs, organisation, learning and teaching strategies, assessment procedures, resource usage and all other aspects of the curriculum. A system of evaluation needs to be established and incorporated into learning and teaching computing, whether this takes place under the umbrella of computer studies or is integrated into all subject areas of the curriculum. The evaluation procedure needs to be simple and ongoing. The information collected only becomes useful if it is analysed, interpreted and appropriate change decisions based on it are made and implemented. The total process of evaluation is slow and gradual, but it does provide directions for possible change.

Perhaps the most important part of designing assessment and evaluation is specifying exactly what their purpose is. The purpose will determine the scope of the evaluation and direct limited time and other resources towards important targets. The purpose of the evaluation will determine for whom I will collect the information, who will be affected by and informed of the resultant decisions.

The following are some of the questions one might answer before committing oneself to certain assessment and evaluation procedures:

1. What am I trying to evaluate?
   - the appropriateness of objectives;
   - the relevance of topics;
   - the effectiveness of teaching strategies;
   - the appropriateness of assessment procedures;
   - student progress.

2. How will I collect the information?
   - questionnaires;
interviews;
observation -- checklists;
diaries -- teaching records, student records;
student projects;
formal tests -- analyse results;
peer assessment;
student’s self-assessments.

3 When will I collect the information?
during a lesson;
during the teaching of topics;
at the end of each topic;
at the end of each term or year;
by the end of each week.

4 How will I document the information for myself?
charts;
class records;
records on individual students;
records on groups of students;
description of critical events;
descriptive record of each lesson/day/week.

5 How will I report the collected information?
in a written report;
in informally in student interviews;
in informally at a staff meeting;
at a parent-teacher meeting.

CONCLUSION

Current standardised tests do not provide information about thinking processes which a student might be using in his/her attempts to solve a problem, so they do not help in the diagnosis of misconceptions or other specific sources of difficulty. For the assessment of computational learning no such tests are available at all. Different assessment procedures are needed which focus on specific higher order thinking skills and which provide information about such thinking processes as the individual learner’s representation of the task in hand, his/her construction of a mental model, the generating of hypotheses, the identification of a solution path, the planning of steps towards solution, and the self-monitoring of the application of solution strategies. Such assessment can yield diagnostic information that can help guide instruction that is tailored to the needs of individuals with respect to learning processes and the development of understanding and knowledge. In addition to alternative assessment that taps cognitive strategies and component processes of learning, there is a need for the assessment of metacognition. This is of particular importance for educationally disadvantaged students. Traditionally, at all levels of the educational system, programs designed to serve students with temporary or more basic learning difficulties have focused on basic skills training and remediation. However, a growing body of research indicates that many of these students perform badly because of an impoverished or underdeveloped repertoire of problem solving, learning and study strategies. To diagnose these difficulties better, we require assessment of the student’s employment of metacognitive strategies in planning, monitoring, revising, etc. their learning performance.

Multiple choice format can be used to do more than measure the recall of facts. For example, it can be used to assess the student’s understanding of a statement or argument, and to evaluate the student’s ability to check the consistency of information, arguments, etc. However, the limitation of multiple choice questions for the assessment of problem solving and thinking processes is obvious. Problem solving skills such as the identification of reasonable alternatives, the monitoring of solution processes, etc. require responses constructed by the student. Written or verbal responses are needed not only to assess these skills, but to probe the depth of the student’s understanding of arguments, problems, complex situations and his/her ability to understand and think about even more complex messages than are used in multiple choice items. Open-ended test items seem a reasonable way of dealing with the assessment of problem solving, thinking and learning processes, but such testing is very expensive. A compromise could be multiple choice items which include the use of open-ended questions that ask why a given multiple choice option was selected and by following multiple choice tests with interviews.

Other methods which have been used successfully in research and could be used in the assessment and mediation of higher order cognitive processes of students learning with computers include interviews by teachers, student diaries, direct observation of student problem solving, student dialogue, role playing, peer teaching. Eventually, computer simulation (focusing attention on the information and strategies which a student uses to solve problems, including the way he/she attacks the task, the number of hints needed, the efficiency of the solution path, etc.) and intelligent tutoring systems can be added to the list of feasible approaches to the assessment of higher order processes. The long-term aim should be a completely integrated learning and assessment environment in which students are given precisely specified cognitive feedback as part of assessment-based instruction.

As we design settings where teaching and assessment are integrated, the instrumental importance of learning as a basis for future learning will be emphasised. Students learn to read so that they will be able to learn through reading and so that they can acquire and interpret information. We learn to write so that we can organise our thoughts and communicate them to others, and so that
we can clarify ideas and build persuasive arguments. In science, students learn to think systematically and critically about our physical and social world, and to ask questions. Students learn computing so that this powerful tool becomes part of their repertoire of instruments, skills and knowledge that allow them to function in and eventually contribute to society as a whole. In focusing on the development of such enabling competencies certain standards will apply, just as they do in current achievement testing, but the types of skills, knowledge and understanding which are assessed, and the way in which knowledge is exercised, will lead to different criteria for evaluation.

PART III

THE EMPIRICAL STUDY