THE EDUCATIONAL IMPLICATIONS OF ARTIFICIAL INTELLIGENCE

Margaret. A. Boden

This paper will be published in Thinking: An Interdisciplinary Report, ed. W. Maxwell (Franklin Institute Press, Philadelphia, in 1983).

Cognitive Science Research Paper

Serial no: CSRP 017

The University of Sussex
Cognitive Studies Programme
School of Social Sciences
Falmer
Brighton BN1 9QN

THE EDUCATIONAL IMPLICATIONS OF ARTIFICIAL INTELLIGENCE

Margaret A. Boden

is paper will be published in Thinking: An Interdisciplinary Report,
. W. Maxwell (Franklin Institute Press, Philadelphia, in 1983).

Abstract

can help us to understand and improve thinking. Via its influence on gnitive and developmental psychology, it promises to illuminate the ocedural complexities of thought. Applied in the classroom, it can ster autonomy and self-confidence in normal and handicapped children, d provide tutorial aids significantly more flexible than those of aditional CAI. Through nonspecialist courses in higher education, AI n encourage the judicious computer literacy that modern societies will ed.

am grateful to the British Council (Committee for International operation in Higher Education), the Commonwealth Foundation, and the iversity of Sussex, for sponsoring my visit to the Conference on inking held in January 1982 at the University of the South Pacific, ji, for whom this paper was written.

EDUCATIONAL IMPLICATIONS OF ARTIFICIAL INTELLIGENCE

The volcanic peaks of Fiji, described by Rupert Brooke as the most fantastically shaped mountains in the world, remind me of the well-known reason for climbing Everest: "Because it is there". This reply may be adequate justification of the mountaineer's obsession, but it would not suffice to explain why people involved in education should be interested in artificial intelligence (AI). AI is the attempt to write programs enabling computers to do things that would involve intelligence if done by people (Boden, 1977). Like every human activity, it has its own peculiar fascination. But there are more pressing reasons why AI is educationally relevant, reasons of both a theoretical and a practical kind.

Many cognitive psychologists today look to AI for help in understanding problem-solving, learning, and intelligence. Even creativity might be illuminated by AI-ideas. Psychological theory can be expected to influence pedagogical practice, and relevant recommendations have already been drawn from the AI way of thinking about thinking. The entry of AI into the classroom in the form of AI-based automatic tutors calls for an appreciation of the differences between this approach and the traditional view of computer-assisted instruction. Current work with handicapped children suggests that AI-ideas can help them to realize their intellectual and emotional potential. And the increasing use of computers in schools and universities prompts one to ask whether social life will be impoverished by the widespread introduction of "intelligent" programs into educational institutions. For these various reasons, then, educationists might be expected to take an informed interest in AI.

Educational psychology and pedagogical practice alike are unavoidably (if often implicitly) influenced by general psychology. Today, theoretical psychologists increasingly draw concepts from AI and computer science in asking questions about thinking. According to the computational approach, thinking is a structured interpretative process. In this view, AI agrees with many non-behaviorist psychologists -- such as Piaget, for instance. Indeed, AI agrees with Piaget in a number of ways, including the commitment to formalism and cybernetics, and the insight that psychology (being concerned with meaning and symbol-manipulation) is semiotic rather than causal. However, Piaget gave only vague answers to his questions about thinking and its development, and also failed to make his questions about these matters sufficiently detailed: his vocabulary of "disturbance," "regulation," and "compensation" is inadequate to express the procedural complexity involved (Boden, 1979; in press, b) . Nor is Piaget alone in this. Non-computational psychologists in general tend to underemphasize mental process, taking it for granted as unproblematic rather than enquiring into it. This is hardly surprising, since computational concepts are needed to express the content, structure, construction, comparison, transformation, function, and development of differing representations and information-processes. A central lesson of AI, then, is that our theoretical aim should be to specify the procedural complexity of thinking.

One way of attempting to do this is to write computer programs that achieve an intellectual task that human thinkers can manage. Because programmed procedures must be explicitly and rigorously defined, this exercise may provide ideas as to what psychological processes might be involved -- and it will certainly help to locate lacunae in current psychological theory. However, the way in which a program does something may bear very little relation to the way in which human minds do it. Careful comparisons need to be made between the various levels of the program and psychological data, to assess the degree of match between the artificial and the natural systems. In many cases, the relevant data are not available. Often, there are methodological difficulties in deciding just which aspects of the program one might plausibly expect to be worth empirical testing (some aspects are included merely in order to produce a program which will run, and have no psychological interest). And many psychologists are not sufficiently interested in the activity of programming to want to spend their time in writing complex programs. For these reasons -- not to mention a positive commitment to working with human subjects -- many psychologists sympathetic to AI do not desert empirical research for the computer console. Rather, they try to plan their experiments with computational questions in mind, their studies being more closely focussed on the procedural details of thinking than is usual.

In developmental psychology, for instance, the computational influence has been largely responsible for the increasing interest in microdevelopmental research. This studies the dialectical interplay between action-sequences and changing cognitive representations (theories, models, heuristics, choice-criteria . . .). The emphasis of microdevelopmental studies differs from more traditional approaches in emphasizing the specifics of action, on the assumption that the procedural detail of performance (not only its overall structure) give clues to the underlying competence. Admittedly, Piaget (for instance) took seriously details of action which others had ignored as trivialities. But the degree of detail aimed at in microdevelopmental research is greater -- and that which would be needed to specify an adequate computational theory of these matters is greater still.

For example, a microdevelopmental study of children's learning to balance blocks found that a non-balanced block may at first be ignored as an apparently irrelevant anomaly, and only later be accepted as a genuine counterexample challenging (and prompting improvement of) the child's current theory (Karmiloff-Smith & Inhelder, 1975). This fact is not predicted, still less explained, by generalized talk of "accommodation". The experimenters suggested that time is needed for "consolidation" of any theory -- but they did not ask just what consolidation is, and how it is effected. These questions would need to be answered if "consolidation" were to be accepted within a computational theory of cognitive change (Boden, in press, a).

Again, microdevelopmental work has cast doubt on the common assumption that the classificatory power of 5- and 10-year-olds is very similar (Thornton, 1982). This view relies on the fact that the <u>product</u> of classification may be identical as between these two age-groups, but it ignores the fact that the <u>activity</u> of sorting is significantly different. The author's experimental design highlights many procedural differences, and she interprets her observations in broadly computational terms. She suggests that children of 10 treat the whole classification as a single unit composed of interrelated classes, that at 5 they proceed as though each class were independent of the others, and that 7-year-olds attend to the relations between classes so as spontaneously to effect the transition by organizing their initially "juxtaposed" procedures into more coherent systems. She admits that the procedural content of concepts like these needs to be clarified if cognitive development is to be understood, and is currently attempting such a clarification with the help of AI-ideas. (With reference to bugs and creativity, both discussed below, one should note that this author takes her work to show that cognitive change need not be failure-driven, a conclusion that is supported by the comparable finding that a child asked to draw maps may spontaneously construct a more powerful map even though the current one has always succeeded (Karmiloff-Smith, 1979).)

The educational potential of AI has been explicitly recognized by a number of workers in the field. One of these is Seymour Papert, an ex-colleague of Piaget who has been deeply influenced by Piaget's ideas about autonomous constructive learning and the epistemological relevance of the structure (not only of knowing but also) of what is known. Papert's ideas are likely to be influential, not least because in November 1981 he was invited by President Mitterand of France to advize on a new Paris computer research centre (with a budget of $20 million a year) devoted to the development of a low-cost pocket-sized computer that will be available on a mass scale throughout the world. In a recent book, Papert (1980) explores the promise of the nascent "computer culture", focussing not on the many uses people will find for computers, but rather on the power of computational environments to affect the way people think and learn -- and, crucially, the way they think about themselves.

Papert reminds us that psychological theories of thinking usually affect educational practice not *via* detailed hypotheses but *via* relatively general ideas, and he identifies a number of "powerful ideas" that enable one to think more confidently and effectively. An important example is the notion of "bugs" in thinking. This concept originated in computer programming, wherein one soon discovers the ubiquity of bugs. Bugs are mistakes, but not just any mistakes: a false factual assumption is not a bug, nor is a momentary slip in executing some procedure, nor the choice of a procedure that is wholly inappropriate to the goal. A bug is a precisely definable and relatively systematic erroneous variation of a correct procedure.

Several AI-workers have attempted a classification of bugs. Sussman (1975) distinguished several types in terms of general teleological notions such as goal, brother-goals, and prerequisite; he wrote a self-modifying learning program that diagnosed its bugs so as to criticize and repair its self-programming accordingly. More recently, O'Shea and Young (1978) have analysed a large sample of children's subtraction-errors in terms of the deletion or overgeneral application of individual rules, such as the "borrowing" rule. Brown and VanLehn (1980; Burton, 1981) have also studied subtraction, and their programs "BUGGY" and "DEBUGGY" provide a notation for precisely describing bugs and a diagnostic tool for identifying errors in students' work. They are developing a "generative theory of bugs", a set of formal principles that can be applied to a particular (correct) procedural skill to generate all the bugs actually observed in the data, and no others. They expect their theory to predict the bugs that occur during the learning of arithmetic, algebra, and calculus (and, possibly, operating computer systems or controlling air traffic).

Their central idea is that many bugs are "patches" (a term drawn from computer programming) that arise from the attempt to repair a procedure that has encountered an impasse while solving a particular problem. Various repair heuristics and critics are defined by the theory, and the way in which a repair will be attempted is theoretically independent of the reason why the procedure was incorrect in the first place. This enables the authors to explain the phenomenon of "bug-migration[11], wherein a subject has a different bug on two tests given only a few days apart. Using their diagnostic system, they find that only certain bugs migrate into each other, and that they seem to travel both ways. For instance, "Stops-Borrow-At-Zero" migrates into "Borrow-Across-Zero", and vice-versa. The hypothesis is that bugs will migrate into each other if (as in this example) they can be derived by different repairs to the same impasse. Repair theory thus makes empirical predictions about the detailed pattern of errors observed when people are learning skills of thinking.

Despite its emphasis on error, "bug" is an optimistic rather than a defeatist notion. For it implies that elements of the correct procedure or skill are already possessed by the thinker, and that what is wrong is a precisely definable error that can be identified and fixed. In this it differs from the broader notions of "anomaly" and "counterexample," the educational value of which has been stressed for instance in the Piagetian tradition (Groen, 1978). As Papert puts it, the concept of "bug" helps one to think about thinking in "mind-sized bites." These insights led Papert to develop the LOGO programming language (usable even by six-year-olds), in the conviction that AI in the classroom could help children to a fruitful insight into their own thinking-abilities. There is some evidence that the experience of LOGO-programming does indeed encourage children to replace the passively defeatist "I'm no good at this" with the more constructive "How can I make myself better at it"? (Papert, 1980; Howe e£ jiL., 1979).

Papert thus stresses the educational value of the activity of programming itself. But AI can enter the classroom in another way, namely, in the form of tutorial programs. Automatic teaching-aids, of a sort, have long been with us. B.F. Skinner's "teaching-machines", and their descendants in "computer-assisted instruction" (CAI), can vary their response to a limited degree with the student's level of understanding, by means of branched programs with predefined choice-points. But the flexibility of tutorial programs based in AI is much greater, because they incorporate complex computational models of students' reasoning that enable them to respond in more subtly adaptive ways. A number of such programs already exist that are useful in limited domains, and several groups around the world are working on these issues (Sleeman & Brown, 1981). Only if a clear articulation of the knowledge involved in the chosen domain has been achieved can it be embodied in an instructional program — though before this embodiment it might be usable by a human teacher in an instructional programme. "DEBUGGY", for instance, is as good as or better than human diagnosticians at discovering the (nearly one hundred) bugs that explain a student's subtraction-errors. In the hands of a specially-primed teacher, it can be put to use in the classroom. It has not yet been incorporated within a remedial program, with which students can interact to improve their subtraction skill; nor has it yet been presented in such a form as to be usable as a diagnostic aid by any maths teacher. But these educational developments are in the forefront of the authors' minds, one of Brown's main aims having long been to develop diagnostic and remedial principles that can be used by tutors – whether human or automatic – to help people Learn (Brown & Burton, 1975). (Some practice with DEBUG6Y might profitably be provided in teacher-training courses, even though it cannot yet be adopted as a classroom tool.)

We have seen that AI helps to foster a constructive rather than defeatist attitude to one's mistakes. But to emphasize the creative potential of bugs is not to say that all creative thinking is a reactive response to failure (Boden, in press, a). On the contrary, it often appears to be grounded in a spontaneous exploratory urge. This much is recognized by psychological accounts of creativity in terms of "competence", "adaptation-Level", "functional assimilation", and "play". However, creativity cannot be understood by way of these concepts, nor by any other structurally undifferentiated, quasi-quantitative, notions of novelty and familiarity. For such concepts enable us to say little or nothing about precisely how individual creative achievements come about. A theory of creative thinking should be able to explain how these or those novel thoughts are generated, how promising pathways are recognised in preference to probable dead-ends, and how potentially interesting ideas are distinguished from novel banalities.

The idea that AI might help answer these questions strikes many people as paradoxical. It is commonly assumed that, because of its programming provenance, AI must be fundamentally incapable of modelling creativity. Were this so, its educationaL relevance would be gravely limited, for a prime aim of education is to encourage creativity. However, unless it is either random or somehow essentially mysterious, creativity must be grounded in some systematic generative principles. From the viewpoint of theoretical psychology, which assumes thinking itself to be food for scientific thought, to regard creativity as essentially mysterious is intellectually defeatist. And that creativity cannot in general be a random process (though there is sometimes a random aspect to it) is recognized by all who scorn the idea that a barrowload of monkeys with typewriters could produce *Hamlet*. Rules or generative principles there must then be, and since AI is specifically concerned with transformations in generative structures one may expect it to be relevant.

Although most AI studies do not attempt to model systems in which genuinely novel ideas arise, or radical constraints are relaxed, some relevant work has been done. For instance, Lenat's (1977a; 1977b) "Automatic Mathematician" starts off with some elementary concepts of set-theory and a collection of heuristics (rules for combining, transforming, and comparing concepts), and sets out to explore their potential in an open-ended way so as to discover new mathematical concepts. Significantly, the program does not merely churn out new ideas, but focusses on some heuristic pathways as more likely to be promising than others, and some novel ideas as more interesting than others. Thus having discovered the natural numbers and decided to explore this path, it then discovers and dubs "interesting" concepts such as prime numbers, square roots, and maximally-divisible numbers (with respect to which last the program developed two minor new results in number theory).

Granted that the heuristics were thought up by Lenat rather than by the program, it is significant – and surprising to many people – that this sort of fruitful exploratory thinking can be formally represented at all. The degree of creativity evinced by the program is, however, difficult to assess. The concept of creativity is itself so unclear that it is not obvious just what would count as "discovering", or "creating", natural number theory (or anything else). Critics (Hanna & Ritchie, 1981) have remarked that Lenat does not list all the concepts regarded by the program as interesting: perhaps a high proportion were mathematically trivial. It is not clear from the published accounts whether some crucial "discoveries" were made possible only by the use of unacceptably ad hoc heuristics, nor is it easy to draw the line between an acceptably specialized expert heuristic and a disingenuous programming trick. Certainly, many of the heuristics are highly domain-specific, relevant only to set-theory. But it is a prime theoretical claim of Lenat's that intelligence depends heavily on expert knowledge, as opposed to very general skills,

Lenat's view that special-purpose heuristics are necessary to creative thinking is consonant with the view of intelligence now held by many people in AI. In the early days of AI-research, it was a common assumption that very general thinking procedures suffice to solve most problems. This faith was reflected in the title of one of the most famous early programs, the "General Problem Solver" (Newell & Simon, 1963), and it motivated much of the early work in "theorem-proving". Since then, it has become increasingly apparent that, while there are some relatively general strategies (such as depth-first or breadth-first search, for instance), the intelligent deployment of knowledge also involves large numbers of domain-specific heuristics suited to the structure of the subject-matter concerned.

Like the notion of "buggy thinking," this view of intelligence contradicts the all-too-common idea that intelligence is a monolithic ability, which one either has or lacks willy-nilly. If more ammunition against so-called "Intelligence Tests" were needed, there is a full arsenal here: the AI approach highlights the absurdity of trying to assess people's intelligence by deliberately preventing them from using any of their acquired expertise (Gregory, 1981, pp. 295-333).

Intelligence being the deployment of many special-purpose skills rather than one general-purpose ability, learning and microdevelopment must involve the gradual acquisition of myriad domain-specific facts and heuristics. Many of these are presumably picked up during the initial "immersion" in a problem-domain, when the unskilled person may appear to be merely thrashing-around. Just how they are picked up is, however, obscure. The microdevelopmental studies previously enjoined thus need to focus on precisely what information is being attended to by the child at a given time, and what micro-strategies she is using to deploy it, with what results.

The case for asking these informational questions, with reference to distinct procedural rules, has been argued in the context of an AI-model of children's seriation-behavior. Young (1976) showed that qualitative behavioral differences can result from the addition or deletion of one simply definable Condition-Action rule. Moreover, the use of a rule once it has been acquired depends on tests related to its appropriateness in a particular informational context. For example, even adults will use a trial-and-error seriation strategy if given a large number of blocks, differing only slightly in length. Piaget explained this in terms of "regression" from the formal to the concrete operational stage, implying that the subject chooses a sub-optimal method over an optimal one. However, the informational demands here differ from those when there are only a few blocks, of obviously differing lengths. The perceptual judgment of which block is the largest (or smallest) cannot now be made "instantly," since the information from so many blocks cannot be handled all at once. Consequently, the optimal informational strategy is to compare the blocks one-by-one. Young's study of seriation is in the microdevelopmental rather than the macrodevelopmental category, not only because he is able to explain minute details of behavior (such as the stretching out of the hand towards a block that is not then picked up), but because of his AI-based view that intelligent behavior is better described in terms of many independent rules than in terms of holistic structures.

Handicapped children can benefit greatly from an AI-based computational environment (Weir, 1981; Weir & Emanuel, 1976). I have in mind here not the uses of computers as gadgetry (controlling typewriters and the like), practically important though these are. Rather, I am thinking of recent research showing how AI can help encourage a variety of intellectual and emotional abilities. That is, AI can be used not only to study the mind of a handicapped person, but also to liberate and develop it. Weir, a psychiatrist with a mastery of AI techniques, has worked with a number of different handicaps and has started a long-term project with the sponsorship of the MIT School of Education. Commenting on the varied examples she describes, Weir points out that we have as yet only scratched the surface of what is possible.

For example, her work with a severely autistic child suggests that a sense of autonomous control (over oneself and others) may develop for the first time as a result of the experience of interactive (LOGO-) programming. The immediacy of results and the non-human context (in which the threat of personal rejection or adverse judgment is removed) combine to provide an inducement for the emotionally withdrawn child to venture into a world not only of action, but of interaction. Interaction with human beings follows, apparently having been facilitated by the computational experience.

Again, one may wish to build on and improve the spatial intelligence of severely palsied children. Since they lack normal sensorimotor experience, one might expect them to suffer from generalized disabilities of spatial cognition. But manipulative tests are clearly of little value in assessing just what abilities a palsied child has or lacks. The use of computer graphics (for which LOGO was developed) provides a window onto the intelligence of these children, one that allows diagnosis of their specific difficulties in understanding spatial concepts. Weir's aim being not just to understand their minds but to help change them, she has the satisfaction of reporting considerable advances in the children's intellectual achievement and general self-confidence.

Linguistic defects, too, may be bypassed in assessments based on computer graphics. For instance, a grossly dyslexic boy was found by Weir to have a superior spatial intelligence, involving highly developed metaknowledge in the spatial domain. The dissociation between linguistic and spatial knowledge is, of course, consonant with the AI view of intelligence discussed above. Much as I suggested above that "DEBUGGY" mnight be useful for teacher-training, even though it is not ready for use as a classroom tool, so ideas arising from Weir's LOGO-projects might be put to use in the training of teachers for the handicapped. But since it is a prime claim of her approach that the experience o1 interaction with a LOGO-machine is itself highly therapeutic, she woulc recommend increased availability of computers for use by handicapped people.

This raises an aspect of the "computer culture" awaiting our children that has not yet been mentioned, namely, the enormous increase in the number of computers used in society. By 1980 there were already two million personal computers in use in the USA (Levin & Kareev, 1980), and the market is expanding; and there is an increasing use of programs by institutions (governmental, medical, educational, and commercial). In their discussion of "the future with microelectronics", Barron and Curnow (1979) point out that, as well as vocational training and adult retraining, we shall need contextual education to ensure that everyone is aware of the technology and its possible consequences. As users get less expert, there will be an increasingly urgent need for relevant nonspecialist courses in higher education. They conclude that "It should perhaps be a target that every graduate has the capability to use computer systems and a thorough understanding of their potential [and, I would add, of their limitations}" (p. 231) .

Several universities are already running courses with these aims in mind, and some people are already doing comparable work with school pupils. For instance, we at the University of Sussex have found that one can alert naive (and non-numerate) users, on their first day of programming experience, to the facts that even an "intelligent" program is incapable of doing many things that one might _prima facie_ expect it to do, and that even a nonspecialist user may be able to modify the program so as to make it less limited. A conversational or visual program, for example, is initially impressive, but the user soon realizes that apparently "obvious" inferences about the meaning of the input words or pictures are not actually being made by it. The beginner-student can then attempt to supply the missing rule so that the un-made inference can now be drawn. Since they themselves are altering these complex systems, students gain confidence in the activity of programming. More important, they realize that programs, however impressive they may be, are neither godlike nor unalterable.

These insights would not readily be communicated merely by teaching students to program — in FORTRAN, for example, or BASIC. They are best conveyed by way of specially prepared teaching-demonstrations making use of AI techniques (ours owe a great debt to the late Max Clowes, whose imaginative vision of student-friendly computing environments inspired us all). Educational projects such as these are socially important, since for most people the ability to write usable programs will be less important than the ability to use — and to avoid misusing — programs written by others. This sort of computer literacy will be necessary if people are to be able to take advantage of this new technology rather than being taken advantage of by it.

At my own university, AI enters undergraduate teaching in a number of ways besides that just mentioned. Experimental psychologists, for instance, take a course with a strong AI element. Especially pertinent here, because of its broad educational relevance and the fact that it is already being used as a source of ideas in other institutions of higher education, is the Cognitive Studies Programme.

This is a three-year "B.A." degree course (situated within the School of Social Sciences) in which students spend at least two-ninths[1] of their time on AI. About half of their time is devoted to their •'major[11] subject: either philosophy, or social psychology, or developmental psychology, or linguistics. Each of these, of course, is a discipline that attempts in some way to study the systematic principles and processes that make intelligence possible. And two-ninths more are taken up either by another of the subjects on this list, or by further AI. As their introduction, students take the "Computers and Thought[11] course referred to in the preceding paragraph, and also a course which sketches the theoretical connections between the several subjects included in the Programme. The Programme is interdisciplinary rather than multidisciplinary, for each tutor has a knowledge of several of the areas and an intellectual commitment to highlighting their connections. The AI element has turned out to be the intellectual core of the Programme. Its status as _primus inter pares_ results from its power to clarify theoretical thinking about thinking, and .to provide concepts applicable to any discipline concerned with the study of thought (we hope to include majors in education, anthropology, and sociology in addition to those already listed). Other universities and polytechnics are embarking on similar educational enterprises, so that higher education in psychology and the other human sciences will increasingly include some acquaintance with AI ideas.

Widespread access to computing environments, especially in primary or middle schools, may have social-psychological effects that educationists should think about. The computer-junkie, or "hacker" (Weizenbaum, 1976, pp. 115-126) has already appeared in infantile form — so much so that a brochure for a children's computer-camp reassured parents that their offspring would not be allowed to remain at the terminal all day, that they would be _forced_ to ride, swim, or play tennis. Whether this presents a threat to normal social development is not yet known. Research on the impact of such environments on young children's play-patterns is currently being planned (Robert Hughes, National Playing-Fields Association: personal communication), in the hope that any unwelcome changes in play-behavior which ensue could be forestalled in future.

One should not assume, however, that any changes in social interaction would necessarily be unwelcome ones. For instance, there is some evidence in the LOGO-projects that the greater self-confidence induced by a child's experience of computing can lead to less anti-social behavior. Moreover, programming contexts are in some ways less oppressive than interpersonal ones, and so have a liberating potential that could be useful in education. This potential has already been mentioned with respect to the autistic child who was led to interact with people after the safer experimentation in a computational environment (and it has been observed also in the context of medical interviewing (Card et^jj^*/ 1974)). A computer system is something to which (not to whom) one can direct remarks that do not carry their usual social consequences (Pateman, 1981). Interaction with the system thus avoids the sort of face-saving manoeuvres which, in interpersonal contexts, can inhibit the creative exploration of ideas: "I wonder what it will do if I say this?" is significantly less threatening than "I wonder what she will think of me if I say that?"

In sum, AI has much to offer to people involved in the theory or practice of education. It can help both in the understanding and the improvement of thinking. Through its influence on cognitive and developmental psychology, AI promises to deepen our insight into the procedural complexities of thought. Through its applications in the classroom, AI's view of intelligence as a self-corrective constructive activity can help foster personal autonomy and self-confidence. This is so with respect to normal and handicapped students, children and adults. Used as the basis of intelligent tutorial programs, AI can offer greater aid and challenge to both student and teacher than the more familiar forms of computer-assisted instruction. Last but not least, AI-ideas can be used to convey a deeper understanding of the potential and limitations of programs, in societies where computer literacy will be an increasingly important aspect of the communal good. The satisfactions of viewing AI are not those of scaling Tomaniivi or the Namosi Peaks. But AI, too, is there: let us not fail to explore it.

REFERENCES

Barron, R. & Curnow, I. _The Future With Microelectronics: Forecasting the Effects of Information Technology_. Milton Keynes: Open University Press, 1979.

Boden, M. A. _Artificial Intelligence and Natural Man_. New York: Basic Books, 1977.

Boden, M. A. _Piaget_. London: Fontana, 1979.

Boden, M. A. Failure is Not the Spur. In _Adaptive Control in Ill-Defined Systems_ (NATO Workshop, 198l), eds. M. Arbib, O. Selfridge, & E. Rissler. In press (a).

Boden, M. A. Is Equilibration Important? -- A View from Artificial Intelligence. _Brit. J. Psychology_ (Piaget Memorial Issue, 1982), in press (b).

Brown, J. S. & VanLehn, K. Repair Theory: A Generative Theory of Bugs in Procedural Skills. _Cognitive Science_, 1980, _4_, 379-426.

Brown, J. S. & Burton, R. R. Multiple Representations of Knowledge for Tutorial Reasoning. In _Representation and Understanding: Studies in Cognitive Science_, eds. D. G. Bobrow & A. Collins. New York: Academic Press, 1975.

Burton, R. R. DEBUGGY: Diagnosis of Errors in Basic Mathematical Skills. In _Intelligent Tutoring Systems_, eds. D. Sleeman & J. S. Brown. London: Academic Press, 1981.

Card, W. I., _et al_. A Comparison of Doctor and Computer Interrogation of Patients. _Int. J. Bio-Medical Computing_, 1974, _5_, 175-187.

Gregory, R. L. _Mind in Science: A History of Explanations in Psychology and Physics_. London: Weidenfeld & Nicolson, 1981.

Groen, G. The Theoretical Ideas of Piaget and Educational Practice. In P. Suppes (ed.), _The Impact of Research on Education_ Washington: National Academy of Education, 1978.

Hanna, F. K., & Ritchie, G. D. _AM: A Case Study in AI Methodology_. Electronics Lab., Univ. of Kent, 1981.

Howe, J. M., _et al_. Teaching Maths Through LOGO Programming: An Evaluation Study. _Proc. IFIP Working Conference on CAI_, London,

1979.
Karmiloff-Smith, A. Micro- and Macro-Developmental Changes in Language Acquisition and Other Representational Systems. Cognitive Science, 1979, 3, 81-118.
Karmiloff-Smith, A., & Inhelder, B. If You Want to Get Ahead, Get a Theory. Cognition, 1975, 3, 195-212.
Lenat, D. B. The Ubiquity of Discovery. Artificial Intelligence, 1977 (a), £, 257-286.
Lenat, D. B. Automated Theory Formation in Mathematics. Proc. Fifth Int. Joint Conf. Artificial Intelligence, , Cambridge, Mass., 1977 (b), 833-842.
Levin, J. A. & Kareev, Y. Personal Computers and Education: The Challenge ^t£ Schools. Univ. Calif. San Diego, Chip Report 98, 1980.
Newell, A., & Simon, H. GPS — A Program that Simulates Human Thought. *n Computers and Thought, eds. E. A. Feigenbaum & J. Feldman. New York: McGraw-Hill, 1963.
O'Shea, T, & Young, R. M. A Production Rules Account of Errors in Children's Subtraction. Proc. AISB Conference, Hamburg, 1978, 229-237.
Papert, S. Mindstorms; Children, Computers, and Powerful Ideas. Brighton, Sussex: Harvester Press, 1980.
Pateman, T. Communicating With Computer Programs. Language and Communication, 1981, J[, 3-12.
Sleeman, D. H. & Brown, J. S. Intelligent Tutoring Systems. London: Academic Press, 1981.
Sussman, G. J. jA Computer Model of_ Skill Acquisition. New York: American Elsevier, 1975.
Thornton, S. Challenging 'Early Competence': A Process Oriented Analysis of Children's Classifying. Cognitive Science, 1982, (in press).
Weir, S. LOGO as an Information Prosthetic for Communication and Control. Seventh Int. Joint Conf. Artificial Intelligence. 1981, 970-974.
Weir, S. & Emanuel, R. Using LOGO jto Catalyse Communication jLn jm Autistic Child. D.A.I. Research Report 13T Dept. A.I., Univ. Edinburgh, 1976.
Weizenbaum, J. Computer Power and Human Reason: From Judgment Jto Calculation. San Francisco: Freeman, 1976.
Young, R. M. Seriation by Children: An Artificial Intelligence Analysis of a Piagetian Task. Basel: Birkhauser, 1976.