

## **Courseware evaluation: where's the intelligence?\***

*I. Scott MacKenzie* Seneca College of Applied Arts and Technology

**Abstract** Questions are explored that courseware evaluators may pose in establishing the extent to which 'intelligence' is present in Intelligent Computer Assisted Instruction and Intelligent Tutoring Systems. The focus is on key features of intelligent systems including a knowledge base which grows, knowledge of student progress and interaction style, feedback or self-reflection of activities, tolerance of user input errors, learner control, and resources for attaining higher cognitive states. Ways in which courseware evaluators may establish the presence of these features are examined.

**Keywords:** Evaluation; Courseware; Intelligent tutoring systems; Artificial intelligence; Computer-assisted instruction.

### **Introduction**

In a popular TV commercial a few years ago, a slight, elderly lady approached the counter of a fast food outlet to buy a hamburger. To her consternation the delivered goods were short of the advertised claims. 'Where's the beef?' she demanded. In this article, we will pose a similar question with respect to a similar marketing phenomenon. Our question, of course, is 'Where's the intelligence?' and the advertised claims are those accompanying the current genre of software that asserts intelligence. Although our focus is on educational software, or courseware, the issues raised reflect the general drive of the computer and other high-tech industries to respond to the calling of a consumer-driven market. That calling is for more sophisticated tools and toys, often sporting claims of intelligence.

The term 'artificial intelligence' (AI) is somewhat outmoded (perhaps because of the gap between the goods promised and the goods delivered), but the spirit remains. Today, we are more likely to meet Expert Systems, User Adaptive

*Accepted:* 20th February 1990.

*Correspondance:* I. Scott MacKenzie, Seneca College of Applied Arts and Technology, North York, Canada M2J 2X5.

\*This paper is based on research conducted at the Ontario Institute for Studies in Education University of Toronto, Canada M5S 1V6.

Systems, Knowledge-Based Systems, Natural-Language Interfaces, Decision Support Systems, Intelligent Tutoring Systems, Intelligent Computer Assisted Instruction, etc. To some extent they all possess 'intelligence', or so they claim. Our goal is to peel back a layer of the onion and expose the specific dimensions of a system that may support the claim of intelligence. We restrict ourselves to educational applications and focus on Intelligent Tutoring Systems (ITS), Intelligent Computer Assisted Instruction (ICAI), and various 'tools' that boast intelligence via a knowledge base, student model, etc. Our aim is to provide courseware evaluators with questions and, hopefully, answers that can be used to qualitatively establish the extent to which intelligence can be ascribed to educational software products.

The term 'system' is adopted here in reference to both software and hardware, with the understanding that the software products are those used in schools and that typical microcomputers constitute the host hardware. The software is usually, but not always, courseware; word processors, spreadsheets, databases, and programming languages are also examples of software commonly found in educational settings.

### Characteristics of intelligent systems

#### *To what extent can the system learn or develop autonomously?*

The most fundamental characteristic of an intelligent system is that it learns. This can take several interpretations, but generally the implication is, firstly, that there is a knowledge base which grows, and, secondly, that the system is adaptable and/or adaptive to its surroundings. 'Surroundings' generally refers to interaction with the user but may encompass a larger world where sensors and actuators connect to human or other information receivers and transmitters. The issue of learning, therefore, must be addressed along two lines with respect to intelligent courseware: the learning capability of the system, and the learning of the student through interaction with the system.

Intelligent systems for learning generally fall under the spectre of expert systems, a sub-discipline within AI where systems are designed around two central modules: a *knowledge base* and an *inference engine*. The former is the information and the latter is the set of rules for acting on the information. In a later section we shall develop the notion that the knowledge base is dynamic in nature and must include both knowledge of the subject domain and knowledge of the user or student. In the context of ICAI, Dede (1986) calls the knowledge of the user, the *student model* and replaces the term *inference engine* with *pedagogical module*. Finally, the *user interface* is added as a basic component of ICAI systems.

In this paper we do not distinguish between ICAI and Intelligent Tutoring Systems. The distinction between the two terms and the evaluation of successful products is still too tenuous to merit separate analysis. As Wenger (1987) points out, 'ITS research is still far from the ideal goal of a system capable of entirely

autonomous pedagogical reasoning, purely on the basis of primitive principles in domain knowledge as well as in pedagogical expertise' (p. 5). Autonomy has been achieved only in the limited and somewhat naïve cases, where systems generate their own exercises for student practice or evaluation. Although ICAI systems tend to be more self-contained than ITS systems, both Dede and Wenger acknowledge that the features of each lie along a continuum with considerable overlap.

The tutor, tool, tutee partitions of the use of computers in education introduced by Taylor (1980) has been helpful in distinguishing the diverse roles that the computer may take in the school. The *tutee* model (the student programming the computer) is of little interest in the present discussion, but the tutor and tool models are quite convenient. Traditional CAI generally follows the *tutor* model, providing the student with a highly self-contained environment that delivers instruction, whereas open or passive systems, such as *LOGO*, provide microworlds for exploration. After reviewing the literature on educational computing it is apparent that the *tool* model is gaining ground over the tutor model. There is a trend for ICAI and Intelligent Tutoring Systems to move away from self-contained instruction delivery and to provide environments for students to explore a problem space. In a review covering several dozen ICAI packages, Kurland & Kurland (1987) note a trend for this genre of courseware to act as a problem solving monitor rather than to provide complete courses of instruction. Sometimes specific problems are provided, but the concept of delivering instruction and following it with test items (accompanied by appropriate branching heuristics for repeating, reviewing, or skipping frames) is de-emphasized in recent products. The intelligence lies in the nature of support for the exploration of the problem space.

The argument that computers have failed to replicate student-teacher interaction with any degree of success perhaps follows from this trend toward open environments. Surely it is easier to design passive (albeit sophisticated or intelligent) tools than to design highly interactive systems which balance the role of the student and teacher (i.e. system) in the efficient delivery of instruction. The migration of ICAI applications toward the tool model may suggest that traditional student-teacher roles are best left as is, and that new products should serve as adjuncts to the educational process, not as replacements for traditional methods.

We now examine several dimensions of intelligent systems, focusing on issues that must be addressed in establishing real increments of performance. These include learner control, knowledge acquisition, fault tolerance, and feedback.

### **Learner control**

*To what extent can the learner control and guide the activities engaged in with the system?*

Drill and practice, the first form of computer-assisted instruction and the easiest

to implement, affords the learner minimal control over his or her activities. Such applications are cited as especially pertinent to the learning of mathematics; however, there is also evidence that mathematics, particularly geometry, can be effectively taught in 'open' environments or *microworlds*. Turtle geometry using *LOGO* provides the student with the tools for interaction without explicitly setting the mode or goals of activities (Papert, 1980). Such environments afford the maximum control possible for learners. In a large study on the use of computers by elementary school students, Carmichael *et al.*, (1985) report the following:

The fact that most students saw a real purpose for and had control over their activities was a powerful motivator. This was particularly true when students were involved in problems of their own choosing. In an environment that gave a student some autonomy and where the teacher created an atmosphere in which mistakes were seen as a natural and healthy part of the learning process, many students retained or developed confidence in their ability to learn from their mistakes and to 'get it right eventually' (p. 285).

It follows from the above comment that perhaps intelligence should not be put into the system at all. *LOGO* is not the only environment that is open and goal-free. *CSILE* (Computer Supported Intentional Learning Environments) is an educational hypermedia application that *supports* learners' activities without determining actions. The designers have suggested that putting intelligence into the computer, in an educational context, is not only unrealistic, it is heading in the wrong direction (Scardamalia *et al.*, 1989):

It is not the computer that should be doing the diagnosing, the goal-setting, and the planning, it is the student. The computer environment should not be providing the knowledge and intelligence to guide learning, it should be providing the facilitating structure and tools that enable students to make maximum use of their own intelligence and knowledge (p. 54).

This 'facilitating structure' comes by way of *procedural facilitation*, an instructional approach that fosters higher-order processes by turning normally covert processes into overt processes, reducing potentially infinite sets of choices to limited, developmentally appropriate sets, providing aids to memory, and structuring procedures so as to make it easier to escape from habitual patterns.

The goals of procedural facilitation are obviously narrower than those of systems with general intelligence, however the potential for success may be proportionally greater. The system need only do the kinds of things that computer systems are already well equipped to do, such as providing formats and reminders, storing and retrieving information, facilitating the making of choices, and monitoring event sequences.

Learner control and intelligence may mesh well in open environments where facilitators are provided as reminders and guides without determining learner activity.

### Knowledge acquisition

Although intelligent systems all possess knowledge of some form, it is the acquisition of knowledge (the learning) that is the real issue, not the representation of knowledge. Acquiring knowledge includes a tacit assumption that the enabling mechanism can distinguish the trivial from the critical, a messy task for humans, particularly learners, and therefore a formidable challenge for machines. In this section we offer questions for determining the extent to which a system can acquire knowledge of the learner, mode of interaction, and subject matter or domain.

#### *To what extent does the system acquire knowledge of the learner?*

When a teacher stands in front of a new class of students at the beginning of a school year, there is little knowledge of the abilities and attitudes of the new learners. Several months later the situation is quite different. The day-to-day and minute-by-minute interaction of a teacher with students is probably guided to a great extent by a teacher's knowledge of the learner that is acquired as a school year progresses.

It seems reasonable that any system possessing intelligence should, similarly, acquire knowledge of the progress of the learner and make constructive use of this in guiding interaction. (Knowledge of 'abilities' and 'attitudes' maybe equally as important as knowledge of 'progress', but are considerably more difficult to acquire and accommodate). Probably, open-ended systems such as *LOGO* and *CSILE* are least likely to build up such a knowledge base while tight drill-and-practice systems are most likely to meet this need.

Of course, the key component that permits the emergence of such knowledge is learner evaluation. Open-ended microworlds are essentially goal-free, and therefore evaluation-free as well. Evaluation exists, but it is not *in* the system; it lies at the interface between the teacher and the student who uses the system for teacher-assigned projects. Drill-and-practice systems, on the other hand, are constantly evaluating students on test items and can easily build profiles of each student that can aid in guiding or sequencing computer-initiated activities. A simple answer to the question at the beginning of this section is that a system is intelligent if knowledge is acquired of student performance (usually based on test items) and guides subsequent activities through branching strategies. This is certainly true of a large body of traditional CAI, so the extent that ICAI goes beyond this needs to be established.

More sophisticated systems, however, may build a knowledge base of the learner without using test items. Human factors engineers have long recognized the need to monitor user actions in order to anticipate their intentions. Simple approaches involve the monitoring of actions, choices, and hesitations in users' sensorimotor actions; however, building a model of the user at this level can involve tremendous online computation (Rouse, 1988). Certainly a static or time-invariant view of the user is not sufficient—a model must be based on

recent activities, expected goals, and associated probabilities. As Rouse points out, appropriate models are very complex and must be developed from several disciplines, including signal detection theory, information theory, manual control theory, and utility theory.

A simple example of an adaptive interface would be one in which on-screen objects (perhaps icons) are large for novices and gradually get smaller as expertise develops. Criteria for scaling could be time spending in making a selection and the number of selection errors. Measurements and predictions can be made using Fitts' law, a model which predicts the time to complete a point-and-select task as a function of target distance, target size, and error probability (Card *et al.*, 1978; Fitts, 1954). As objects get smaller, more room is available on the display screen for other objects or information.

The underlying theories for adaptation are complicated. Regardless, they are quite irrelevant from the standpoint of courseware evaluation. Rouse has provided several Principles of Interaction (1988, p. 440) which raise some very basic issues that can help evaluators in establishing the extent to which a system is adaptive (i.e. intelligent). Adaptation should be appropriately supported to the extent that users always feel they are in control. Are users aware that adaptation is taking place? Do they feel in control? Confusion should be avoided as to the extent that adaptation is in effect; if a process is taking place automatically as a result of the adaptation, users should be able to pre-empt the aid and regain control. Can users easily override default settings or dynamically changing settings?

Adaptive systems with sophisticated sensing and decision mechanisms are likely to be more common in future products, some of which will find their way into classrooms. Since they will be correspondingly more difficult to evaluate, a more rigorous set of questions and criteria will have to be developed to meet this new breed of intelligent system.

*To what extent does the system acquire knowledge of the learner's mode or style of interaction?*

A second dimension of knowledge acquisition lies in a system's ability to mould itself to a learner's style of interaction. When a product is delivered, is its form permanent? Can the system accommodate different modes of interaction? Essentially, we are asking if the system is 'adaptable', rather than 'adaptive'. At a simplistic level, the assigning of multiple keystrokes to function keys is an example of an adaptable system. There is, of course, a whole spectrum of techniques employed to make systems adaptable or 'extensible' so that they match our demands.

Nickerson (1986) reports on an automated history-taking medical system based on a question-answer format. Doctors were highly reluctant to use the system until they had a chance to modify the questions. Even though the changes for the most part were slight, they were deemed necessary before physicians would use the system regularly (p. 243). It was not stated whether the changes

were implemented by the system's designers (as part of a pilot study) or by the physicians, however it is fitting that such power be in the hands of the users of the system.

Wixon *et al.* (1983) report on an interesting experiment with an electronic mail system. Subjects were not given any training on the system but were given specific instructions on what to do. They typed in commands that they thought might work to achieve the goals. Unknown to the subjects, expert human operators intercepted their commands, interpreted them according to their own knowledge of the system, and translated them into commands with the correct syntax. The subjects had the illusion they were interacting directly with the system. Over a 6-month period, changes were gradually introduced into the system's command language to accommodate the *user-defined* commands. The proportion of commands that could be interpreted by the system rose from 7% to 76% over this period of time. It is interesting to speculate on the possibility of a system that could demonstrate such adaptability in the absence of the human intermediaries. Such adaptability could only develop slowly and would have probabilities associated with responses which would be poor at the best of times (compared with an expert's delivery of commands in a conventional environment).

Both of the examples above demonstrate adaptability at the person-machine interface. The content may be deeper within the system, however the first link in the chain—the *interface*—must be strong (that is, powerful and flexible) lest users quickly distance themselves from the system, regardless of the rewards that lie within.

*To what extent does the system acquire knowledge of the subject-matter?*

Expert systems adopt the term 'knowledge base' with reference to domain knowledge, rather than to learner or interaction knowledge. In educational settings, there is a tremendous need for adaptability. Authoring systems boast that they 'bring control of the authoring process back to the content expert' (MacKnight & Balagopalan, 1989, p. 1231), but can they bring it into the classroom and into the hands of the teacher or student? This may be more important in the long run. Systems that can do this are those for which there is no content *per se*. The microworlds of *LOGO*, and *CSILE*, are examples. Content only exists in these systems to the extent that it is added by students or teachers.

HyperCard (by Apple Computer, Inc.) is another example. Nicol reports on the Open School/Vivarium Project in Los Angeles where students create the content of their courses by authoring HyperCard stacks:

In contrast with written or video materials, and even with more typical computer software, HyperCard stacks are 'malleable'. The appearance of cards can be dramatically changed by a few editing strokes; cards can be easily added and deleted at any point in a stack. A single stack can seem to be organized in very different ways depending on the route the user chooses through the stack, and a particular bit of information can be

portrayed in very different ways depending on the designer's choices (Nicol, in press).

In a sense, these systems can acquire content knowledge, but it comes through the system's passivity and its ability to be 'filled up', so to speak, by the users, students, or teachers.

Knowledge acquisition is a tricky business for intelligent systems and a great deal more research effort is needed before systems can claim to grow along with their users. Nevertheless, ICAI packages should support knowledge base ascension, be it of the subject matter or of the learner.

### **Fault tolerance**

*To what extent is the system tolerant of human error?*

Fault tolerance has long been an engineering goal for computer systems; but the focus is usually on hardware failures accompanied by backup and recovery. Since interface faults are also of paramount importance to overall success, systems that boast intelligence should not facilitate destructive operations. Although courseware may be designed to deliver content with follow-up evaluation based on correct and incorrect responses to test items, errors of another sort will occur. These errors, of course, occur at the interface, through slips of syntax or a mistaken understanding of how the system works. Norman (1983), in an analysis of common errors, suggests the following guidelines for designing systems amenable to human fallibility:

- provide *feedback* so that the state of the system is always clearly available to the user;
- include *dissimilarities* in response sequences which cause different actions to take place;
- make commands *revocable* by making those with high consequences difficult to do;
- make the interface *consistent* to minimize memory problems in retrieving system commands.

On the first point above, Fischer (1988) has pointed out that students are often expected to attain goal states that are too distant from their current state, with the result that they often get dumped into unfamiliar territory. A paradigm of 'increasingly complex microworlds' is proposed (p. 139) where each state carries with it protective shields for the novice. Precise representation of the knowledge contained in the microworld and knowledge of navigation rules are required before proceeding to the next microworld. For example, a student may be required to demonstrate the technique for returning to the current microworld before moving on.

The notion of 'consistency', Norman's last point, has been attacked recently by Grudin (1989) as a fallacy. He asked design professionals to choose the best layout from several possibilities for numeric keypads, cursor keys, alphabetic keys, and menus. The experts consistently failed to identify the demonstrably



superior design. Grudin's point is simply that the criterion for consistency should be content of use; that is, objects should be grouped together and organized based on 'how they are used' not on what they are.

ICAI packages should be expected to provide a simple interface that is functionally consistent and flexible enough to minimize errors and permit a high degree of latitude for acceptable user responses. An intelligent interface is one which is malleable to variations which are likely to arise when different learners use the same interface.

### **Feedback**

#### *To what extent does the system support self evaluation?*

Although any system with a screen display provides feedback to the learner, educational systems can benefit by providing feedback that supports 'reflection' of learner activities, progress, and results. Learners should be able to see what they have just done in a graphic rendering of the approach used in solving a particular problem. Collins & Brown (1988) call this *spatial reification* noting that multiple representations may serve an important role for learners who are attempting to solve a problem through exploration (p. 4). They demonstrate this using an intelligent tutoring package called *ALGEBRALAND*, where students are provided with an algebraic expression and are asked to solve for a particular variable. Access to multiple windows is provided allowing review of different aspects of the problem. The windows show basic operations (add, multiply, distribute, expand, etc.), planning possibilities (isolate, collect, group, etc.), recording operations, and graphs of the sequence of operations. The search space window shows the sequence of operations as connected nodes with the original question in a box at the top and the various steps shown within boxes connected to previous boxes. By examining the various paths (including dead ends), students can determine which strategy is the most effective.

A similar approach to self evaluation is present in *GEOMETRY TUTOR* (Anderson *et al.*, 1985), a learning environment for geometric proofs. The developers note that 'floundering' is always possible in such environments, since students can blindly proceed following a trial-and-error strategy. Obviously, a balance must be struck. Herein lies an opportunity for an adaptive systems approach. Students can be expected to demonstrate their knowledge of basic operators in prerequisite activities before they are allowed to use them later in more sophisticated domains. This is consistent with Fischer's notion of increasingly complex microworlds mentioned earlier.

*GEOMETRY TUTOR* supports multiple modes of self-evaluation by a graphic display of the problem in the same window in which the solution is worked out. Students may solve the problem piecemeal. Separate parts of the solution may be displayed simultaneously, such as the final solution along with several preceding steps, as well as the initial problem statement along with the first several steps. The goal, of course, is to use geometric rules in building a valid path from the initial problem to the solution.

It seems reasonable that ICAI courseware should support self-evaluation allowing the learner to investigate the problem space in a format that exploits the graphic capabilities of today's high-resolution bit-mapped displays. Multiple representations permit problems to be approached from different perspectives depending on the nature of the problem and the learner style or preference.

### General issues in machine intelligence

In reviewing the above comments several statements can be constructed of the form, 'A system is intelligent if ...', where the statements are filled out using any of the dimensions of system utility discussed. However, it is easy to replace 'intelligent' with 'sophisticated'. Perhaps we should reserve claims of 'intelligence' for some future time when advances are truly impressive, when systems effortlessly display traits such as intuition and common sense in interactions with humans. One problem with reserving judgement can be put as follows: whether or not something strikes us as intelligent depends on our own understanding, and once we understand, the intelligence—the mystique—is gone. One of the first attempts at natural language interaction was Weizenbaum's (1967) ELIZA, which played the role of a psychotherapist. At first glance, a transcript of patient-system dialogue is impressive, however, as Weizenbaum pointed out over 20 years ago:

Once a particular program is unmasked, once its inner workings are explained in language sufficiently plain to induce understanding, its magic crumbles away; it stands revealed as a mere collection of procedures, each quite comprehensible. The observer says to himself, 'I could have written that'. With that thought he moves the program in question from the shelf marked 'intelligent', to that reserved for curios, fit to be discussed only with people less enlightened than he (p. 36).

Similarly, Nickerson (1986) offers definitions provided by three AI specialists, preferring Kay's idea that '(artificial intelligence) is stuff that is interesting that we do not know how to do yet' (p. 276). Elaborating on this point, Nickerson describes the situation as follows:

AI researchers have observed on several occasions that the criterion for what constitutes thinking or intelligent behavior has changed along with the accomplishments of the AI community:  $x$  may be among the set of activities considered to be examples of intelligent behavior, until someone manages to program a computer to do  $x$ , at which time it is removed from the set ... (AI) has been a gradual clipping away at the set of things that people can do and computers cannot. That set is not likely to become empty any time soon, if indeed it ever will; but progress has been steady (pp. 276–277).

A problem germane to education is that we do not actually know what the cognitive processes are that contribute to the attainment of higher levels of learning. Designing systems that bring learners to advanced levels of knowing is

therefore attacking a problem before it is properly understood. The so-called 'learning paradox' occurs when learners are expected to draw upon concepts or procedures more complex than those presently available in order to attain a higher state. Bereiter (1985) offers 10 resources for 'bootstrapping' cognitive growth, each offered with cautious reservation that the problem is poorly understood. Techniques such as affective boosting, use of spare mental capacity, construction of a self-concept, etc., are offered, but only as 'resources'. The paradox remains. We may add to our set of questions, the following:

*Using what resources can the system advance a learner from one cognitive level to a higher cognitive level without being explicitly programmed to do so?*

The proviso 'without being explicitly programmed to do so' is needed lest the answer becomes trivial. Programmed instruction may easily lead students from lower to higher cognitive levels, say from an understanding of integer numbers to an understanding of real numbers, but an intelligent system must be able to do the same without the presence of explicit instructional sequences. Once the process is made explicit, a system's utility is focused and finite, not, we would say, intelligent. The question posed above is not likely to be asked by courseware evaluators concerned with outcomes, implementation, training, etc. Nevertheless it focuses on one of the most important issues for intelligent systems.

One of the resources offered by Bereiter may be particularly important—that learning may advance by a process of 'chance plus selection'. The idea is that advanced states may follow from random processes, and that 'if the organism can capitalize on the fortuitous successes by preserving a trace of the behavior that led up to them, it then has the possibility of acquiring new competence' (p. 208). This echoes Darwin's theory, and the connection is acknowledged. The ability to incorporate random variables with probabilities linked to outcomes may be a central, perhaps essential (Sen, 1989), ingredient of future intelligent systems. Certainly though, models must be developed in general settings before migrating into specific domains such as courseware.

### Conclusion

Intelligent courseware has arrived, but where's the intelligence? The answer may be that intelligence lies wherever one wishes to place it, as long as the featured activity is new or more sophisticated than that of a previous product with the same activity. Finally, we caution that marketing moguls will attach any tag to any product that raises it a cut above the competition. We have only begun to witness products cast as 'intelligent'. Education, as a bureaucracy taxed by the demands of politicians and parents, is particularly sensitive to such ploys. As Clark (1983) points out, '(rational choice) must compete with the advertising budgets of the multimillion dollar industry which has a vested interest in selling these machines for instruction' (p. 456).

It seems most reasonable—following the ideas of Weizenbaum, Kay,

Nickerson, and others—to reserve the tag 'intelligence' for human qualities, and to adopt a view that intelligence is a temporary state of knowing without knowing why. Once we understand why, we have uncovered one small piece of the puzzle, and machine implementation of the new piece brings sophistication, but not intelligence. Nevertheless, courseware evaluators will be confronted with intelligent products that may bewilder and even intimidate. It is hoped that the questions and discussions presented in this paper can provide a starting point for evaluating the extent to which 'intelligence' is present in these products.

### References

- Anderson, J. R., Boyle, C. F. & Reiser, B. J. (1985) Intelligent tutoring systems. *Science*, **228**, 456–468.
- Bereiter, C. (1985) Toward a solution of the learning paradox. *Review of Educational Research*, **55**, 201–226.
- Card, S. K., English, W. K. & Burr, B. J. (1978) Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, **21**, 601–613.
- Carmichael, H. W., Burnett, J. D., Higginson, W. C., Moore, B. G. & Pollard, P. J. (1985) *Computers, Children and Classrooms: A Multisite Evaluation of the Creative Use of Microcomputers by Elementary School Children*. Toronto: Ontario Ministry of Education.
- Clark, R. E. (1983) Reconsidering research on learning from media. *Review of Educational Research*, **53**, 445–459.
- Collins, A. & Brown, J. S. (1988) The computer as a tool for learning through reflection. In *Learning Issues for Intelligent Tutoring Systems* (eds H. Mandl & A. Lesgold) p. 1–18. Springer Verlag, New York.
- Dede, C. (1986) A review and synthesis of recent research in intelligent computer-assisted instruction. *International Journal of Man-Machine Studies*, **24**, 329–353.
- Fischer, G. (1988) Enhancing incremental learning processes with knowledge-based systems. In *Learning Issues for Intelligent Tutoring Systems* (eds H. Mandl & A. Lesgold) p. 138–163. Springer-Verlag, New York.
- Fitts, P. M. (1954) The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, **47**, 381–391.
- Grudin, J. (1989) The case against user interface consistency. *Communications of the ACM*, **32**, 1164–1173.
- Kurland, D. M. & Kurland, L. C. (1987) Computer applications in education: A historical overview. *Annual Review of Computer Science*, **2**, 317–358.
- MacKnight, C. B. & Balagopalan, S. (1989) An evaluation tool for measuring authoring system performance. *Communications of the ACM*, **32**, 1231–1236.
- Nickerson, R. S. (1986) *Using Computers: The Human Factors of Information Systems*. MIT Press, Cambridge.
- Nicol, A. (in press) Children using HyperCard. In *HyperCard and Education: Education Advisory Council Journal*. (eds S. Ambron & K. Hooper) Apple Computer, Inc., Cupertino.
- Norman, D. A. (1983) Design rules based on analyses of human error. *Communications of the ACM*, **26**, 254–258.
- Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.

- Rouse, W. B. (1988) Adaptive aiding for human-computer control. *Human Factors*, 30, 431–443.
- Scardamalia, M., Bereiter, C., McLean, R. S., Swallow, J. & Woodruff, E. (1989) Computer-supported intentional learning environments. *Journal of Educational Computing Research*, 5(1), 51–68.
- Sen, C. (1989, December) On intelligence and randomness. *Computer*, 22(12), 73.
- Taylor, R. P. (ed) (1980) *The Computer In The School: Tutor, Tool, Tutee*. Teachers College Press, New York.
- Weizenbaum, J. (1967) ELIZA—A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9, 36–45.
- Wenger, E. (1987) *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann, Los Altos.
- Wixon, D., Whiteside, J., Good, M. & Jones, S. (1983) Building a user-defined interface. *CHI'83 Conference on Human Factors in Computing Systems*, 24–27. ACM, New York.

### **Editor's comments**

It is a pleasure to find such clear perspectives on the possible existence of intelligence in software from Scott MacKenzie. His approach from the direction of software development complements that of Jim Greeno's (1991) educational standpoints when the latter refers to a didactic and exploratory dichotomy for the role of computers in education.

Both writers reflect views of a simple model for the role of computers about which I spoke some time ago to the British Association for the Advancement of Science and the British Educational Research Association:

'... two of the problems of CAI remain, namely the maintenance of a suitable model of the learner and a hierarchical framework for the subject matter being taught.'

(Lewis, 1979, p. 54)

'It is possible to draw a distinction between the roles by asking the simple question "Is the computer (program) assessing the student?" to which one can answer 'yes' or 'no' at least in terms of the computer's predominant role. The answer 'yes' gives rise to concern since the available models of learner and subject matter content are very crude, even non-existent, at present. Yet the essence of good teaching includes some appreciation of what the learner does and doesn't know, why he can and can't do; a thorough understanding of the subject and skill in communicating it to the learner. One way around these shortcomings would be to make the program 'learn'; in other words to be self-adaptive as experience in use indicates the most successful pathways and feedback loops. This, however, is beyond the state of the art. The more familiar ways of using the computer as a resource in the classroom . . . do not depend on the models which play such a critical role when the machine is used as a surrogate teacher. Here the computer acts as a resource on an equal footing to, say, laboratory apparatus, to be used as

and when it has a part to play in providing or supporting student enquiry.'

(Lewis, 1981, p. 42)

I was able to stress this model again recently when reacting to Jim Greeno's paper in which I went on to question how *educationalists* become convinced that software was behaving in an 'intelligent' fashion:

'A specific criticism of reports on so-called ITS systems arises over the lack of any explicit description of the learner model. All reports should enable the reader to find the answer to a series of quite simple, yet fundamental questions:

- how is the learner model represented in the software?
- how is a particular response by the learner assessed?
- in what terms is the learner informed of the assessment?
- how is this assessment incorporated into the learner model?
- how does this up-dated learner model influence the way in which the software continues to run?

The answers to these questions should be provided in a general form, supplemented by one or more quite specific examples. The answers are bound to include statements of the way in which domain knowledge is being represented and the pedagogic strategy being employed as well as providing detail of the control mechanisms of the software.

It is only when provided with such evidence that a reader can judge the "reasonableness" of the system being presented....'

(Lewis, 1991, p. 358)

The difficulties of discovering where (if at all) the intelligence lies in so-called AI-based products was met when we tried to analyse data from a world-wide survey of AI in learning projects, the DISTIL Survey. Readers may like to investigate the same problem for themselves by further analysis of the database which is outlined in Twidale & Mace (1990). More generally, readers' comments to JCAL on the claims for ITs will be welcomed.

### References

- Greeno, J. (1991) Productive Learning Environments. In *Advanced Research on Computers in Education*. (eds. R. Lewis & R. S. Otsuki) Elsevier Science Publishers, Amsterdam.
- Lewis, R. (1979) Education, Computers and Microelectronics. In *Microelectronics: advanced technology for the benefit of mankind*. (ed. J. R. Forrest) British Association for the Advancement of Science, London.
- Lewis, T. (1981) Pedagogical issues in designing programs. In *Microcomputers in Secondary Education: issues and techniques*. (eds. J. A. M. Howe & P. M. Ross) Kogan Page, London.
- Lewis, R. (1991) Perspectives on Information Technology Support for Learning. In *Advanced Research on Computers in Education*. (eds. R. Lewis and S. Otsuki) Elsevier Science Publishers, Amsterdam.
- Twidale, M. & Mace, T. (eds.) (1990) *Analysis of the DISTIL Survey*. ESRC-InTER Occasional Paper (InTER/19/90). Department of Psychology, University of Lancaster.