

Artificial intelligence and expert systems in education: Progress, promise and problems

Alexander Romiszowski
Syracuse University, USA

Basic concepts

It is not our purpose here to get into a deep analysis of the meanings and accepted definitions of Artificial Intelligence and Expert Systems. Indeed it appears that no universally accepted definitions exist, a fact that is bemoaned by some authors as a source of much confusion in the field and an open invitation to "exploit the hype - being generated about the field in popular and quasi-technical media" (Harris and Owens, 1986). In their opinion, the *artificial* half of AI is generally taken to mean *man-made*, but there is much disagreement and lack of definition as regards the term *intelligence* as used in this context. Some researchers even refuse to be associated with this term. For example, I once attended a seminar given by Gordon Pask the cybernetician and creator of the CASTE system of conversational CAI (a name often quoted in the AI literature), in which he argued that AI does not exist, on the grounds that:

1. if a machine were to act in a way that one recognised as being intelligent, then the machine will be intelligent (why the need for the prefix *artificial*), but
2. he has not as yet seen a machine that is truly intelligent and doubts if he ever will (on the grounds that, for him, true intelligence has an affective / emotional component and he has never seen machine display emotion).

So, to avoid stepping yet deeper into this semantic quagmire, let us accept a simple functional definition, quoted by Rich (1983), which is quite adequate for our purpose here, despite Harris and Owen's criticism that it intentionally avoids the definition of both *artificial* and *intelligence*:

AI is the study of how to make computers do things at which, for the moment people are better ...

Despite the implication of a *fluid* field (once we succeed in matching or exceeding the performance of humans our computer application suddenly ceases to be intelligent) this definition is particularly pertinent to the theme of this conference as it raises the question of whether, and if so when and in what tasks, will computers be better than teachers.

Following the same functional approach, let us select the definition of *expert system* quoted in the user's manual for the *Personal Consultant* expert system shell produced and marketed by Texas Instruments (1985):

Expert Systems are software programs that help you solve complex reasoning tasks that normally require a (human) expert ...

In other words, an expert system should help a novice, or partly experienced, problem solver, to match acknowledged experts in the particular domain of problem solving that the system is designed to assist. By the way, the term *shell* used above is another way of saying *expert system development tools* but rather an apt way, as the tools provide an empty (content free) structure into which the system developer may fit the specific content and inference strategy that is necessary to solve problems in the given domain.

To be more specific, expert systems are generally conceptualised as depicted in *Figure 1*. The user makes a consultation through the interface system (the communication hardware and also the software which defines the types of queries and formal language to be used) and the system questions the user through this same interface in order to obtain the essential information upon which a judgement is to be made. Behind this interface are two other sub-systems: - the knowledge base, made up of all the domain-specific knowledge that human experts use when solving that category of problems and - the inference engine, or system that performs the necessary reasoning and uses knowledge from the knowledge base in order to come to a decision with respect to the problem posed.

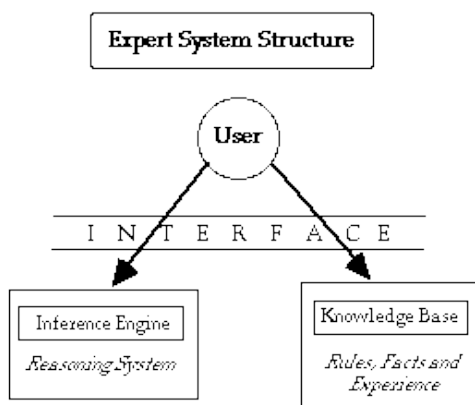


Figure 1

In a sense, the knowledge base and inference engine are analogous to the knowledge stored in memory and the reasoning capabilities of the human experts that the system is emulating. However, the analogy is imperfect. The inference engine contains a set of formal logic relationships which may or may not resemble exactly the way that real human experts reach conclusions (but that does not matter as long as the conclusions reached are identical). And the knowledge base is structured in a formal manner, which varies with the *shell*, but most commonly is *rule-based*. The rules have to be defined in a limited number of formal ways. Typically they may be a set of some hundreds of *if-then* (or *if A and B but not C then D*) types of relationships that describe all the domain-specific knowledge used by the human expert.

And here lies the rub, because human experts do not usually have all their knowledge about a subject neatly structured into a large number of rules of a particular format - life and humanity just is not that well organised. It may be that at a the brain-chemistry level, human knowledge is stored in specific standard ways, but we do not know how for sure and, certainly, we are not personally aware of the exact way that we structure what we know. As a result, the most difficult and time consuming part of developing an expert system is the extraction of knowledge from the head of an acknowledged expert (or better a group of experts) and then transforming it into a form acceptable to the expert system's knowledge-base structure. This task is so important, complex and time-consuming that it has earned itself the rather grand name of *knowledge engineering* (some prefer *cognitive engineering*, though this is not quite a synonym). Some writers (eg, Harris and Owens, 1986) query however to what extent this activity warrants to be termed engineering, rather than *art or craft*.

The field of AI in education: Its structure

In order to structure our analysis of the potential areas of application for artificial intelligence-based systems in education, I have borrowed the organising framework used by Robert Taylor (1980) when writing in general about the educational uses of computers. Taylor organised the book he edited around three possible roles for the computer. These roles are:

TOOL - the student uses the computer and appropriate applications software to perform a task related to the course of study - typical examples are the use of word processing, statistics packages or spreadsheets - these would normally be used in support of other learning/teaching activities, but in some cases they may become integral parts of the instructional system, as in the case of the ever more common use of word processors as a means of teaching creative writing;

TUTOR - the computer takes on the teaching function, presenting a series of screens of information, test questions and feedback, much as in programmed instruction, or involves the student in a simulation or game situation built up to promote specific educational objectives - this *role* includes most of the common modes of CAI, such as drill-and-practice, tutorial, simulation and also dialogue and conversational modes that attempt to adapt to and learn from the individual learner, much as a human tutor;

TUTEE - the student *teaches* the computer and, in so doing, learns something as well - this mode includes programming the computer and thus learning computer programming and systems analysis, but the best known example of this category is the use of the LOGO language, by which means the student in programming the computer is supposed to gain all manner of insights and *powerful ideas* of high transfer value. (Papert, 1980)

However, it is not only the student who uses computers in education. Teachers also may do so, and so may instructional designers, book authors and many other groups. They certainly use computers as TOOLS and no doubt often learn a great deal in the process. Less frequently, perhaps, do these groups use computers in the TUTOR or TUTEE roles, specifically to learn something that may help them in their jobs. But it does happen and may become more frequent in the future. So, to fully explore the potential field of applications of AI and expert systems in educating, let us carve up the field into a two by three matrix, as shown in figure 2.

User	Tool	Tutor	Tutee
Student	1 Commercially produced expert systems used to solve problems (JOB - AID)	2 Intelligent computer-assisted instruction (ICAI)	3 Use of LISP or PROLOG in schools
	Student learns by repeated use	Intelligent database search assistant (EMBEDDED TRAINING)	Students construct expert systems for the domain under study
Teacher/ Instruction Designer/ Author	4 Stand-alone expert systems on instructional design, decision making, planning, controlling, etc.	5 Intelligent embedded training in school applications software	6 Teacher/Subject matter expert constructs expert systems on specific domains to use in modes 1 and 2
	Intelligent authoring systems for CAI	Intelligent teacher-training systems	

Figure 2

The examples listed in the six cells of this matrix have been chosen to illustrate practical applications of AI systems and in particular *Expert Systems* which have already been attempted or are currently in the stages of development or design. Some are more of a current reality, whilst others, though technically feasible, may for a variety of practical or economic reasons remain in the realm of laboratory research or even pipe dreams for a long time yet. Let us briefly examine each of the six cells of our *field*.

Progress in the application of AI to education

Cell 1: As a TOOL used by the STUDENT.

The most common application of AI so far is in the form of expert systems. Although not yet very common, there are already many examples in

commercial use and the market is estimated to grow exponentially during the next few years. It should not be long before commercially developed expert systems will be available in large numbers to assist in all manner of problem solving tasks, including many of interest and relevance in the schoolroom. After all, an expert system is just another form of JOB PERFORMANCE AID, and as such will surely permeate into school use as slide rules 50 years ago, calculating machines 20 years ago and work processors and spreadsheets yesterday. Many teachers resisted those innovations in their time and no doubt some may resist the entry of expert systems now. I can hear the arguments - "it's all right for the kids to use computers for the repetitive or routine tasks of calculation or writing, but with expert systems on tap, they'll stop thinking for themselves". But will they? And indeed do they think now in the way that the expert software would illustrate? After all, an expert system, to be commercially viable, must represent an exceptionally outstanding level of reasoning, albeit in a tightly defined domain. And most expert system software permits (indeed always should permit) the user to interrogate and analyse the reasoning process. At every stage of a consultation the user may press function keys to ask either why a certain item of information was requested by the computer or how the computer has reached the conclusion that it is recommending. Therefore the user may follow in detail the reasoning processes built into the expert system and, in so doing, may learn to reason in a similar manner.

This argument is akin to the argument for presenting formulae, worked examples, and all forms of guidelines, algorithms and so on, as learning aids. One may argue that it is not always desirable that students learn by following pre-prepared examples and procedures and that sometimes they should discover the procedures for themselves. But life is too short to discover everything. And anyway, in interacting with an expert system, the user is being shown the reasoning steps that were followed in one particular example problem - it rests with the student to generalise (or not) from the example and discover the general principles of logical reasoning that the system is applying. It may be an important function of the teacher, using expert systems in the classroom, to make sure that the discovery of those general principles really does occur. But to adopt the neo-Luddite attitude of keeping expert systems out, is not all that different from attempting to keep students from interacting with real experts, or from watching one of C P Snow's excellent television lectures on world history and affairs. The better approach would seem to be to study how the educational potential of interacting with expert systems might best be exploited in the school environment and work towards the sensible integration of these new tools into our instructional systems, before events overtake us and, through out-of-school contact, our students become more expert with expert systems than we are.

Cell 2: As a TUTOR of the STUDENT

This list [below], particularly the last observation, would suggest that intelligent tutoring systems have not yet left the laboratory stage. This opinion was shared by Frank Roberts of Control Data Corporation in a Keynote address to the 1986 conference of the Association for the Development of

Computer-based Instructional systems (ADCIS), when he remarked that to date there have been about 20 ICAI systems developed, about 15 of these ever reached the stage of practical implementation, only about five have been used on a large scale for a reasonable time and none has been formally evaluated.

<i>Contrast</i>	CAI	ICAI
The developers (mainly) ...	Educators ...	Computer scientists
Their goals are to ...	Instruct students ...	Explore AI
They start from ...	Task analysis ...	Knowledge structures
They try to teach ...	Almost any subject ...	Well structured areas
Their methodology is ...	Systems approach ...	Personal, varied
Instructional mode(s) ...	Vary W/ objective ...	Mainly discovery
Adapts to student's ...	Single responses ...	Response pattern
Validation/evaluation ...	Formative to effectiveness efficiency	Technical opinion and debugging

Perhaps one reason for the relatively slow progress in the sub-field of intelligent tutoring systems is that the developers are trying to break new ground in several directions at once. The general structure of a typical ICAI system may be represented by the schematic in Figure 3.

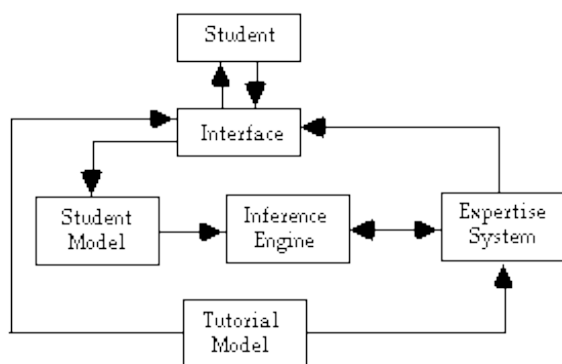


Figure 3

The student interacts with the system by means of some interface system, receiving information and providing responses to questions or sometimes INITIATING DIALOGUE by asking questions - this raises a host of problems associated with natural language communication, two-way graphic communication and so on, which are still in the early stages of research and development. The system then must respond to the student in a manner appropriate to the individual pattern of responses and queries that are received. It must adapt to the individual student's needs, learning style, difficulties, etc. It does this by building and constantly updating a MODEL of the individual student's characteristics - how to do this is another field for much research, involving the social as well as computer sciences. Information from the student model is then used by the inferencing system to modify the instruction accordingly - there is a need for an adaptive TUTORIAL MODEL

and the principles on which this adaptation should take place should be drawn from our as yet fledgling instructional science. These adaptations in the tutorial and indeed in its direction, mean that the system must have at its disposal a very expensive and well structured knowledge based, or rather TWO knowledge bases, one on the subject matter that is to be taught and the other on the teaching methods and tricks associated specifically with that subject. This *Expertise System* is a combination of the knowledge of the subject matter expert (our famous SME) and of the expert teacher of that subject.

Lest it be thought that we are writing off ICAI as a development which is yet to prove itself and destined to remain for many years the plaything of researchers, let us remember the surprising speed of recent developments in the computer field and not underestimate the cumulative effect of the many developers now working in this field. At this time, some dozens of development projects are under way in the United States alone, many aiming to implement practical military training ICAI systems in the shortest possible timespan. We may well have reached a *critical mass* in the body of research that will lead to very fast practical progress from now on.

We must also remember that partial implementation of the model described above may make good sense in many situations. One approach which may cut some of the R and D corners and may have considerable practical application is exemplified by a project which I began to develop a few years ago in Brazil. The overall aims of the project were to develop efficient, individualised updating/retraining systems for electronics technicians from a variety of industries where microchip technology was being introduced (Arce and Romiszowski, 1985). The instructional materials were prepared in printed paper form, but in order to be usable for a variety of differing objectives, with a variety of trainees, in a subject area in constant development, the materials were written in a highly modularized and very carefully structured manner. Each concept, principle, practical procedure, etc. of digital logic theory, circuit design and troubleshooting was isolated and described in a separate MAP, using the authoring, or *mapping* techniques suggested by Robert Horn of Information Mapping Inc. (Horn 1974). The resultant set of over 1000 maps formed a comprehensive and coherently structured knowledge-base on digital circuits, suitable for both learning the theory or practical application in industry, covering just about any current need and accessible, through the map titles as keywords, in a non-linear and totally user-orientated manner.

In practice, a particular trainee, or group, follow a sequence of practical laboratory activities, determined by their individual learning needs rather like a conventional, mastery-learning oriented, practical course. Each laboratory activity is separated by a pre/post test which verifies the current state of knowledge of the trainee. This test will direct the trainee to the study of relevant maps in the knowledge base. In the original implementation, this diagnosis of need and prescription of relevant study material was performed by an instructor with the aid of a data-base management system (set up in Dbase-II) which stored the structure of the knowledge-base (but not the actual materials). The general procedure is shown in Figure 4. This shows that the

knowledge-base is composed of maps that contain all the theory (TE) of the domain, maps that explain and guide practical application (applied technology - TA) and specific references to library materials, which the trainee may use to go beyond or deeper into any topic that proves to be of particular interest.

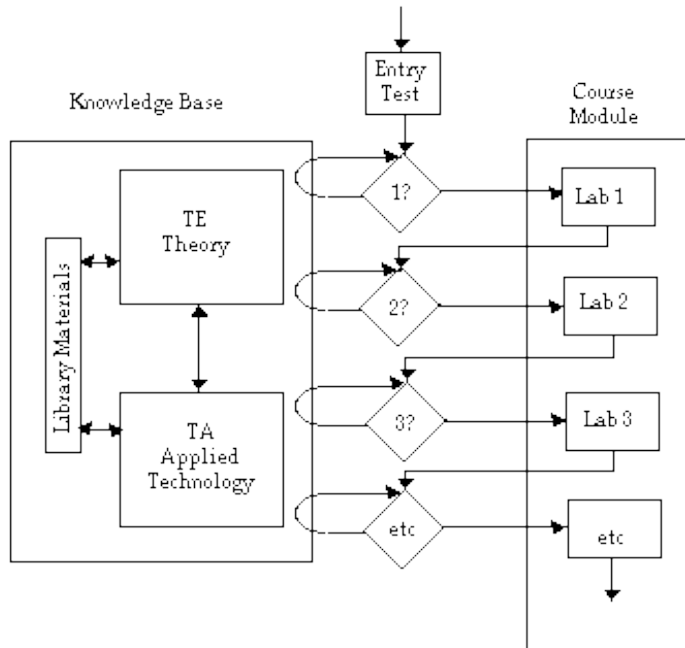


Figure 4

The evaluation of the trainee and diagnosis of needs, selection of further study material, etc. is performed by the instructor who takes into account the trainee's individual long term objectives, learning pattern to date, history of errors and difficulties and so on. Data on all these aspects is accumulated and stored in the same data base (in Dbase-II) that is used to access and select the instructional materials. The instructor uses the data available in order to make more informed and therefore hopefully more intelligent decisions concerning the needs of each trainee at each point in the course. By tracking the instructor's decisions over a series of courses, evaluating, and comparing the learning effects of following those decisions, it should be possible, over time, to build up, almost automatically, an expert system to perform the diagnosis and prescription tasks. Thus by linking up a subject matter knowledge base (either stored in or merely accessed and managed by means of proprietary data base management software) to an expert *instructor* system (developed from practical observations using a proprietary expert system shell) we may construct a useful student evaluation and guidance system which operates with at least a modicum of intelligence (Romiszowski, 1986) This may not be a

full ICAI system, but it's not far off and may be realised with existing research know how and with existing and readily available hardware and software.

Another potential *semi-intelligent* approach, which is an adaptation of the above, is *Automatically Diagnosed Embedded Procedural Training*, or ADEPT (the name and acronym are my invention). This is an extension of the trend towards the development of computer software packages with *Embedded Training* which in some way forms part of the package itself rather than being a stand-alone training course to be taken before beginning to use the package. One may think of embedded training in the context of computer-related work as a form of in-service training. Horine and Erickson (1986) refer to this as "the explosive growth area in computer based training" and justify it on economic and practical grounds as well as didactically, in that training is taken when required to be applied in practice and only what is needed is learnt. They define two classes of embedded training:

Integrated, which forms part of the software package itself and may be called up by hitting the help key (except that the help you get well designed and appropriate instruction rather than the encounters on help screens;

Concurrent, which is separately packaged and may be written in other programming languages or even stored on, say, videodisc, but which may be accessed rapidly and accurately whenever a difficulty or a new procedure is encountered.

Most embedded training, even when well designed, has to be called up the user/ trainee when experiencing difficulty. Furthermore, the type of difficulty and its origin is not very expertly diagnosed, as it is usually the trainee who is experiencing the difficulty who has to do the diagnosis. Matters would be a bit different if an expert user had been peering over the trainee's shoulder and had observed (he response patterns that had led up to the difficulty. Chances are that the training needs diagnosis would be much more accurate and profound. Often the expert may even diagnose an operational weakness on the part of the trainee of which the trainee is completely unaware - for example backspacing by single letters for wordprocessor editing, when there are procedures to return by whole words, lines or paragraphs at a time, or using an inefficient search strategy in using a data base. The ADEPT approach is to build that "expert peering over the shoulder" into the applications software, by tracking the user's response pattern, identifying patterns that are sub-optimal for the task in hand, diagnosing the probable knowledge deficiency and interrupting the user - "hey, I don't think you're using me right - why not try it this way - do you want me to show you?"

Cell 3: As a TUTEE of the STUDENT

In this cell, we may, by direct analogy to the teaching of other programming languages and Logo, in the hope of achieving some more general educational benefits, expect that sooner or later at least some teachers will see value in the teaching of the AI programming languages in schools. Some schools already use the simplified Apple or PC versions of LISP and PROLOG with computer science classes, but there is little evidence as yet of any significant

generalisable benefits, or of a movement to get all schools on the bandwagon, as was the case with BASIC and LOGO. LOGO, after all, was developed from LISP, but with children in mind, so why complicate matters? There are however the beginnings of a movement towards the more systematic teaching of the AI languages in schools. In Europe, a project supported by the European Community and involving Belgium, France, Greece and Italy is investigating the teaching of PROLOG in the secondary school.

The educational aims they believe achievable through this is to "give students an operative knowledge of databases ... skills of organising and structuring their knowledge ... and to help to develop logical-deductive skills ..." (Bottino et. al., 1986).

Another potential development in this cell, and one about which I am personally very excited, is the use of simple expert system "shells" to enable students in school to develop their own expert systems on topics that they are studying. This combines a whole series of pedagogically attractive aspects in a practical and motivational new type of exercise:

- Learning by doing and in the process creating new knowledge, or at least new structure for the knowledge found in books and other sources, as well as working job-aids that may be useful later;
- Learning by discovery, in the domain of higher order problem solving and the creation of cognitive strategies and theories;
- Applying logic of the structure of knowledge to segments of one's own store of knowledge in order to create a computerised knowledge-base and thus learning to reflect on and reorganise the rest of one's personal knowledge base.

An interesting application of this approach is reported by Lippert and Trollip (1986), using a specially constructed, simplified expert system shell with 12 to 15 year old students on a variety of domains of problem-solving, ranging from legal issues (the copyright laws and the copying of diskettes) to science topics. The chief claims they make for this method include:

- Improved Problem Solving, in identifying problems, defining refining and representing them, exploring, assessing and acting on alternative strategies, identifying and evaluating their effects;
- Gains in Procedural Knowledge, of analysis of scope, meaning and critically of concepts, pattern recognition, synthesis of discrete steps into logical procedures, testing of knowledge bases for accuracy, validity, consistency and coherency;
- High Motivation, due probably to practical, goal-directed activity, conceptual conflict, challenge, curiosity, competition;
- High Transfer to other tasks, promoted by development of skills of organisation of knowledge, identification of its essential features, understanding, of basic principles and the similarity of the classroom training and testing to real-life problem solving.

They also report some problems or interactions which may reduce the benefits of this method in some cases. These include: individual differences in cognitive style and learning style which do not always match what the task demands; age as a factor which influences overall success; the difficulty and natural structure (or lack of structure) of the content under study; basic motivation to learn or its absence and, last but not least, the large amount of time necessary to employ the technique and the large individual differences in the amount of time required to reach criterion.

This last point is of great importance, as it may outweigh the many potential benefits in the reality of the classroom. Lippert and Trollip (1986) are using a simple to learn shell on problem domains that typically may require the formulation of 5 to 15 rules that may combine some 5 to 10 items of data to come to one of 3 or 4 alternative conclusions. Already they find the classroom time required excessively long. As we go up the problem solving ladder to domains that are less tightly defined, where the knowledge base must take into consideration cases of incomplete information, where some decisions are probabilistic with percentage weightings and where the total number of rules to be isolated, defined and integrated into the knowledge base, runs into the hundreds, then the time required becomes prohibitive. Desmond (1986) quotes studies of commercial expert system development projects that ranged from 18 to 36 months for the development of medium-size expert systems (about 250 rules) and 36 to 60 months for really large systems (about 1000) rules. Whereas real-life system development times, with all the real life delays, complications and uncertainties, bear no relationship to the times that may be achieved in simulated, simplified and somewhat prompted training exercises, it is evident that times would still be excessive for anything but the development of small sample systems.

Coupled to this is the complication that, as the size and complexity of the knowledge base to be constructed increases, so does the size, complexity, learning difficulty and cost of the necessary software development tools - the shell. I have been using a commercial shell, capable of supporting about 400 rules, for work with postgraduate students in instructional design. We found the learning time to master the development tools to be almost as long as the time to develop a 30 rule expert system.

However, if the benefits quoted by Lippert and Trollip (1986) are achievable through limited exposure the time may be well invested. And we can make the larger systems for dissertation projects.

Cell 4: As a TOOL of the TEACHER (and instructional designer)

If and when effective expert systems are developed for complex decision making tasks that are carried out by teachers, school administrators, curriculum developers instructional designers and media specialists, then they will be used in proportion to the value that educators find they get out of using them. There is no shortage of candidate domains of expert decision making in education, where access to an effective and efficient Job Performance Aid would be welcomed by many and would no doubt make

itself felt in terms of improved decisions and thus results. The moot point is whether the field of education will attract the investment necessary to develop such special aids. The currently quoted figures for time and cost of commercial expert systems development may, as yet, keep them out of the reach of most educational budgets.

But this will not be the case in the training arena, especially in military training. The US armed forces appear to have taken the stance that the sheer quantity, as well as the quality of training that must be developed, updated and constantly revived, precludes the use of traditional, unaided human efforts. The development of automatic training design systems is the target of several current military-sponsored research and development projects. One example is a project to develop an expert system for the design of training devices such as complex simulators and working models (Singer and Perez, 1986). Apparently the US army spends over 7 billion dollars a year on such training devices, so even a small improvement in cost effectiveness as a result of better design should pay for the research project. A similar project, but aimed to develop automatic expert assistants for the design of lessons and development of training materials is being undertaken by the US Navy at the Navy Personnel Research and Development Center in San Diego, California. It is called project AIM, which stands for Authoring Instructional Materials. I am reminded of the 1970's, when a similar project undertaken by the British Royal Navy, was abandoned after about 3 years of development as the number of rules identified and time needed to identify and code them grew well in excess of project resources. In its lifespan, the project identified several hundred separate rules which it successfully integrated to take decisions on the design of psychomotor skills and lower level concept instruction (Dodd, LeHunte and Shepard, 1974). We have come a long way since then, it would seem.

There will no doubt be some spinoff into the educational arena from these military training projects. Also, the currently high costs and times for expert system development will fall. The developmental software is getting cheaper to buy and easier to use with every generation. A recent trade article listed over 20 different expert system development packages, ranging in price from \$US 100 to \$US 50,000 (Desmond, 1986). And the next jump will be expert systems that develop expert systems.

Also, we may expect some early activity by enthusiasts at the small systems end of the market. Greg Kearsley, the author of several how-to-do-it books for trainers has supported his book on cost-benefit in training by a small cost-benefit analysis expert system and another book on computer-based training, by an expert system that helps in the selection of projects for CBT (Kearsley, 1984, 1985, 1986). Work I am presently undertaking at Syracuse University is developing an expert assistant in front-end analysis in training projects and will attempt another and will attempt another in the domain of instructional media selection. No doubt such activities will snowball as more people gain access to ever more user friendly development systems.

However, there is another development in this cell of our field which may be of much greater significance. This is the effort that is currently being devoted to the development of instructional design expert systems as integral parts of CAI authoring languages and systems. Currently available courseware authoring languages offer a wide variety of layout options, effects, graphic support and everything else that computer technology has managed to invent (voice synthesisers and interpreters seem to be one of the current hot buttons). But they provide little or no guidance as to which of these techniques, if any, should be used at any given moment. What is badly needed is advice on how to plan the instructional sequences and messages and how to select appropriate screen presentation effects. Several researchers are working on this problem.

Once again, there are several military-training-inspired projects, but in this case, there is also much education-related work. One of the earlier CAI systems which paid attention to the instructional design aspects of the courseware, was the TICCIT system, which provides templates for design that are based on instructional theory principles. One of the original TICCIT design group, David Merrill, has continued to refine this approach and is now in process of developing the LDS or Lesson Design System, that is destined to be a part of a future authoring system that would take the author in an interactively controlled manner through all stages of the design/development process, starting with task or topic analysis, then specification of measurable objectives, sequencing and classification of the objectives and finally matching instructional strategies and tactics to objectives. In this last stage, the system will be a practical embodiment of Merrill's Component Display Theory of instruction (Merrill, 1983; Merrill and Wood, 1984).

Many other researchers are working towards similar goals. In the United Kingdom, McAleese and his collaborators are developing a Concept Learning and Authoring System (CLAS), which is based on knowledge representation theories (McAleese 1986). Pask (1984) in Canada has been working on *Thoughtstricker* and so on.

The reason why I see these developments as particularly important and interesting is that the improvement and sophistication of educational courseware authoring systems and languages may well be a necessary condition for the survival of "conventional" CAI in education. The argument goes as follows:

- Computers are getting more plentiful, cheaper and easier to use. So is applications software. As educational courseware authoring systems become more user-friendly, more versatile in terms of the "bells and whistles" they offer, more accessible and cheaper, an ever growing number of teachers and would-be authors will develop an ever increasing number of competing packages;

- As the number of new authors swells, the average quality of the packages produced will decrease, for the growing, but relatively small number of first class packages will be swamped by the very much larger number of mediocre courseware products. All or most publishers will want to get in on the act while the going is good and, as they will be in a hurry and in any case will not have the skills to judge quality CAI, a lot of mediocre materials will be published and hard-sold successfully in a gullible market;
- As CAI applications get more frequent, the failures, the errors and the wasted resources will become ever more apparent and will form an ever greater proportion of the visible reality. At some point in this process, the administrators and controllers of the educational budgets will realise that the expected benefits have not been achieved, that costs are ever rising and that other so far overlooked parts of the educational enterprise are badly in need of the funds being spent on CAI. Funds will start to be cut off. The first to notice the trend will be the publishers, who will stop publishing new packages, especially the ones which perhaps are of high quality (for they inevitably cost more to develop unless sponsored by some grant - and these too will suddenly become very scarce).

Have we seen this trend before anywhere? Was not this the main reason for the so rapid rise and fall of the Programmed Instruction (PI) movement? (Romiszowski, 1974). PI was not all bad and many programs were first class bits of instruction.

Where are they now? Like the baby with the bath water, they went down the educational drain with the overwhelmingly larger quantity of second rate materials. The same fate almost befell instructional television in the USA. Initially interest projects, program production and budgets soared to glorious heights, only to plummet as producers and scriptwriters paid the price of over enthusiasm not backed by competence or judgement. Now we have seen a slow and painful rise in quality and finally budgets, like a Phoenix, which has to kill itself in order to reborn. This "phoenix effect" has been observed worldwide (see Tiffin 1980 for an account of the Latin American scene).

Let us leave the speculation of whether the CAI scene will suffer the "phoenix effect", whether perhaps the bird is dying already, and whether the improvement of authoring systems by the incorporation of AI instructional design aids might lead to its rebirth, for our later discussion. And let us now deal quickly with the two remaining cells of our organising framework. I have included these in the matrix for symmetry, though I do not have much to add to previous discussion in their respect.

Cell 5: The computer as TUTOR of the TEACHER (or designer ...)

All that has been said in the context of TUTOR/STUDENT earlier on, may well apply in this case. Three factors may influence what will actually happen. Firstly, the "student" group in this case is smaller and highly diversified in the professional needs of the individuals that make up the group. This will make it more difficult to justify the costs of developing expensive AI courseware,

especially if education budgets for research and development continue at present low levels.

Secondly, the student group are adults and a relatively selected group, who have above average learning skills. The group as a whole may have less need for the more carefully designed and more interactive courseware that the AI approach should provide. Also, being adult, the group may in the main reject CAI and with it ICAI as a preferred means of study - and who is going to force them?

But thirdly, CAI, whether intelligent or not, may not be the appropriate method or medium for the instruction of the truly important skills of the future educators. If, for argument's sake, the use of CAI and especially ICAI were really to provide a breakthrough in educational effectiveness and efficiency and if it were then to be widely adopted as the most cost effective approach to the bulk of cognitive instruction, then indeed the role of the teachers would change out of all recognition. They would not be eliminated from the educational scene, but would spend most of their time doing those things that (to return to our definition of AI) at that moment, people will still be doing better. And a large part of this will be the affective, moral, ethical and general personality development of students. It's not that teachers don't do these things now, but they spend only a fraction of their time in these domains. In the "brave new world" scenario I just painted, this would become the Major part of their duties. And it is probably that computers will not prove to be the best media for learning how to perform these duties, unless, or course, Gordon Pask were to be proved wrong and machines did come to exhibit feelings and emotions. As this is not very likely, we shall be able to cling to our dignity, agreeing with Pask that all this machine progress is not really true intelligence.

Cell 6: Computer as TUTEE of the teaching professions (Teacher, Instructional Designer, Author, Administrator).

All members of the teaching professions and their support services may obviously find a need to learn computing in order to teach a computer to perform some special purpose program of tricks. But this will become ever rarer. The trend is for ever more user friendly applications software, which will require no special training to use, or will be full supported by ADEPT (our friendly Automatically Diagnosing Embedded Procedural Training), which would act more like an intelligent "job-aid using a job-aid", than a formal course of instruction.

The analogy to today's teaching of LOGO to kids, that they will learn some generalisable powerful ideas, does not hold in this case. Most adult programming activity will, as ever, be directed toward specific professional goals. Time for serendipity learning (Rushby 1979) will be scarce for busy professionals then, as it is now. Some of us, who have the time or inclination, might explore the ever wider seas of computer science for the fun or the general educational benefits we might glean from our travels. But most of us will leave that to the professional computer scientists, I believe.

However, there may be an important development analogous to our kids in school constructing expert systems to structure their knowledge-base on a *Specific Well Defined Domain*. If things really were to move in the direction of computer dominance of most cognitive instruction (they probably won't, but let's imagine), then who is going to develop the intelligent tutoring and expert systems. Not the computer scientists, though they may play a supportive role. It will be our instructional designers, now in their role of knowledge engineers, together with the academically inclined section of the teaching profession, who will of course be the subject matter experts. So most academics will be involved in the enterprise of knowledge representation in formally logical structures, of structuring their knowledge in a form that can be taught to computers. And, who knows, if it can be taught to computers, maybe it can be taught to anyone? After all, computers are so dumb, aren't they?

Will books vanish from the marketplace? By no means! Quite apart from the fact that the bulk of books sold are for reading for pleasure and not exclusively for learning, a lot of professional information needs will continue to be best served by presentations in straight narrative text and not in interactive instructional format. Some of this text may be stored and distributed electronically, but much will continue to be printed on paper.

However, the *authorship quality* of these professional texts may undergo significant improvement once the authors have, for other purposes, been involved in structuring the knowledge domain in logical format for, say, an expert system to use. After that, it's just child's play to structure the outline of a book version much superior to anything else on the market. There may also be some spin-off benefits for the organisation of the search for new knowledge identification of gaps in the knowledge base that indicate priority areas for research.

To illustrate this point and to conclude this section I quote from the final conclusion of the paper of Singer and Perez (1986) on their work to develop an expert system for the design of training devices:

With the use of a knowledge engineering tool such as KES, we have *Successfully* structured what is known about the characteristics and instructional features of training devices. As researchers in training and training devices, we have *Acquired some Knowledge* about the design of training devices. We believe that this process *can be applied to other fuzzily bounded and non well-defined knowledge domains* as well. The process has forced us to structure the knowledge available from field experiments and technical reports of validations studies. *That structuring has shown us where answers are needed and what variables are involved in obtaining the answers.*
(The italics are my emphasis - A.R.)

Aspect	+	-
Tool/Student (Commercial Expert System)	<ul style="list-style-type: none"> • No educational R and D investment • Access through local networks • JOB and Learning aid 	<ul style="list-style-type: none"> • Wait for relevant tools to become available • Wait for price to drop
Tutor/Student (ICAI etc.)	<ul style="list-style-type: none"> • Potentially more effective, efficient, educational 	<ul style="list-style-type: none"> • No evidence of superiority so far • Will always be more expensive than CAI
Tutee/Student (Construct Expert System)	<ul style="list-style-type: none"> • Most appropriate for higher order learning 	<ul style="list-style-type: none"> • Time consuming
Tool/Teacher (Designer, Author)	<ul style="list-style-type: none"> • Improve instructional design and other expert tasks • Guarantee quality of CAI courseware 	<ul style="list-style-type: none"> • Change role of instructional design • Change role of teacher
Tutor/Teacher (Designer Author)	<ul style="list-style-type: none"> • Efficiency of embedded training 	<ul style="list-style-type: none"> • Small target audience • Low cost/benefit
Tutee/Teacher (Designer, Author)	<ul style="list-style-type: none"> • Improve the structure of knowledge and quality of materials 	<ul style="list-style-type: none"> • Time consuming

Figure 5

Conclusion: Promise and Problems

In the previous section, I have already mentioned many of the possible sources of promising new developments and also the problems associated with these potential developments. In Figure 5 I bring together these observations and add a few more not discussed above, to act as a stimulant to our final discussion. I present this table of potentially positive and negative aspects without further discussion, hoping that it may stimulate the audience to comment - query, disagreement, elaboration, addition to the list, or whatever. I have, in the paper, been outspoken at times and may have overstated the case at others. It is now your chance to correct me or chide me as needs be.

References

- Arce, J. F. and Romiszowski, A.J. (1985). Using a relational data-base as a means of integrating instructional and library materials in a computer managed course. *Proceedings of 20th ADCIS Conference*. Association for the Development of Computer-based Instructional Systems (ADCIS), Philadelphia, April.
- Bottino, R. M., Forcheri, P., and Molfino, M. T. (1986). Experimenting with PROLOG in high schools: A project. In *Technology - XIX: Educational, training and information technologies - Economics and other realities*. Kogan Page, London.
- Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer assisted instruction. *IEEE Transactions on Man Machine Systems*. Issue No. 11.
- Desmond, J. (1986). AI develops a deeper, louder voice in commercial markets. *Software News*, October 1986.

- Dodd, B. T., LeHunte, R. J. G., and Shepard, C. (1974). Decision making in instructional design. In J. Baggaley et. al. (eds.) *Aspects of educational technology - VIII*. London: Kogan Page.
- Harris, S. D. and Owens, J. M. (1986). Some critical factors that limit the effectiveness of machine intelligence technology in military system applications. *Journal of Computer Based Instruction*, 13(2).
- Horine, J. and Erickson, J. (1986). Embedded training: The explosive growth area in CBT. Paper presented at the 28th International Conference of the Association for the Development of Computer Based Instructional Systems (ADCIS). Washington DC, November.
- Horn, R. E. (1974). Course notes for information mapping workshop. Lexington Mass: Information Resources Inc. (Now revised and reprinted as part of the chapter "Structured writing, or mapping: An information processing approach", in Romiszowski, A.J. (1986). *Developing auto-instructional materials: From programmed texts to CAL and interactive video*. London: Kogan Page.)
- Kearsley, G. (1986). *CBT Analyst - DISKETTE* and *Costs/Benefits - DISKETTE* (Expert Systems). La Jolla, California: Park Row Software.
- Kearsley, G. (1985). *Computer-based training: A guide to selection and implementation*. La Jolla, California: Park Row Publications.
- Kearsley, G. (1984). *Costs, benefits and productivity in training systems*. La Jolla California: Park Row Publications.
- Lippert, R. and Trollip, S. (1986). Building knowledge bases for expert systems. Paper presented at the 28th International Conference of the Association for the Development of Computer-based Instructional Systems (ADCIS). Washington DC, November.
- McAleese, R. (1986). The representation of knowledge in authoring environments. In N. Rushby and A. Howe, (eds.) *Aspects of Educational Technology - XIX: Educational, Training and Information Technologies - Economics and other Realities*. London: Kogan Page.
- Merrill, M. D. and Wood, L. E. (1984). Computer guided instructional design. *Journal of Computer-Based Instruction*, 11(2), Spring, 60-63.
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. New York: Basic Books.
- Park, O. and Seidel, R. J. (1986). ICAI: Intelligent applications of AI. Paper presented at the 28th International Conference of the Association for the Development of Computer-based Instructional Systems (ADCIS), Washington, November.
- Pask, G. (1984). Review of conversation theory and a Protologic (or protolanguage) - Lp. ERIC/ECTJ Annual Review Paper. *Educational Communications and Technology Journal*, 32(1).
- Rich, E. (1983). *Artificial intelligence*. New York: McGraw Hill.
- Romiszowski, A. J. (1986a). Applying the new technologies to education and training in Brazil. In N. Rushby and A. Howe (eds.), *Aspects of educational technology - XIX: Educational, training and information technologies - Economics and other realities*. London: Kogan Page.
- Romiszowski, A. J. (1986b). *Developing auto-instructional materials: From programmed texts to CAL and interactive video*. (Chapter 8 on Mapping of information, chapter 17 on interactive video and chapter 20 on the Brazilian SENAI project and expert systems). London: Kogan Page.
- Romiszowski A. J. (1974). *Selection and use of instructional media: A systems approach*. London: Kogan Page.
- Rushby, N. J. (1979). *An introduction to educational computing*. London: Croom Helm.

- Singer, M. J. and Perez, R. S. (1986). A demonstration of an expert system for training device design. *Journal of Computer-Based Instruction*, 13(2).
- Sleeman, D. H. and Brown, J. S. (1982). *Intelligent tutoring systems*. New York: Academic Press.
- Taylor, R. P. (ed.) (1980). *The computer in the school: Tutor, tool, tutee*. New York: Teachers College Press.
- Texas Instruments Inc. (1985). *Personal consultant: Expert system development tools*. User's guide and software (for PC).
- Tiffin, J. (1980). Educational television: A phoenix in Latin America? In A.J. Romiszowski, and C. Chadwick (eds.), *Educational technology in Latin America*. Special issue of Programmed Learning and Educational Technology.

Please cite as: Romiszowski, A. (1987). Artificial intelligence and expert systems in education: Progress, promise and problems. *Australian Journal of Educational Technology*, 3(1), 6-24.
<http://www.ascilite.org.au/ajet/ajet3/romiszowski.html>