Intelligent Tutoring Systems: A Review for Beginners

Peter Holt
Peter Wood

Abstract: This paper presents the genesis of the use of artificial intelligence (AI) in education. It introduces the concept of an intelligent tutoring system (ITS) and outlines a typical architecture while differentiating ITS's from conventional Computer Assisted Learning (CAL). It outlines recent directions in this field and describes the inputs of various disciplinary areas (Psychology, Education, Cognitive Science, and Artificial Intelligence) that continue to contribute to ITS development. It reviews the potential benefits of ITS's for education and potential issues in implementing them in a large scale way. Finally, the authors suggest areas for research and application development for ITS's (The paper includes a select bibliography.)

INTELLIGENT TUTORING SYSTEMS AND DISTANCE EDUCATION

Introduction

The role of Artificial Intelligence (AI) in education has been of increasing interest to researchers in recent years. However, many of the publications which deal with this topic have been too technical for a general audience in the educational field. More general publications have contained too little depth to be of use to educator who wish to become involved in applications of AI in education. This paper introduces AI techniques to various professionals in the field of education, such as subject matter experts, educational technologists, instructional designers, and management. It assumes some basic computer literacy but no real familiarity with artificial intelligence or cognitive psychology. It is intended that after reading this paper, the readers will be in a position to begin serious investigation of how ITS theory and practice can be applied in their area of expertise.

CJEC, VOL. 19, NO. 2, PAGES 107 · 123, ISSN 0710 4340

THE BASICS OF INTELLIGENT TUTORING SYSTEMS

Artificial Intelligence and Knowledge Representation

A major research focus of artificial intelligence (AI) is to develop heuristics and algorithms that will allow computers to perform tasks that seem to depend upon human intelligence. Such research includes pattern recognition, general problem solving, game playing, and problem solving in specific narrow domains, Attempts at general problem solving programs have not been highly successful. This failure has been attributed to large degree to the lack of a representation of knowledge about the real world in these programs (Minsky, 1986). For instance, to solve medical problems by computer, the machine should possess specific knowledge about the medical domain and diagnostic techniques as well as general problem solving capabilities. It has been conceded that representing the type of general knowledge that a human being has about the real world is an intractable task for computing technology, at least for the immediate future. Research in the last fifteen years has focused on restricted domains of knowledge-limited enough that some representation by a computer is feasible (e.g., in the medical domain, the knowledge necessary for diagnosing the bacterial agent in infections and prescribing the appropriate antibiotic). This research has resulted in the concept of an knowledge based system that performs at an expert level for a restricted domain. It is generally conceded that these "expert" systems are best developed for domains of expertise involvingrestricted procedural knowledge- generally of an heuristic rather than algorithmic nature (Waterman, 1986).

If performance is the major concern, procedures that are algorithmic in nature can generally be more efficiently executed in traditional computer pro grams. Expert systems generally have at least two components: a knowledge base and an inference engine. In the expert systems knowledge base the knowledge of domain experts is represented in an appropriate data structure. Types of representations include production rules of the form 'IF X is true, then do Y" and frames. A frame is a data structure representing an object or situation and holding"slots"which contain values for specific attributes of that object. Slots may also contain actions (procedural attachments) to be performed in specific situations. Most frame base systems are hierarchical and include the concept of inheritance – frames lower in the hierarchy inherit default values for common slots from their ancestor frames higher in the hierarchy For more detail on the such matters the reader should refer to a introductory text on AI (e.g., Charniak & McDermott, 1984).

The inference engine module of expert system makes inferences based on the production rules and other knowledge representations in the knowledge base(s). Commonly used inference methods include backwards chaining, forward chaining, and reasoning with uncertainty. Backward chaining is "an inferencing strategy which involves working back from a conclusion or goal to see if the conditions which would make it true are satisfied" (Slatter, 1987). Also termed "inductive reasoning;" it is particularly useful in domains requiring diagnostic expertise. Forward chaining is an inferencing strategy which builds up from the available data about a problem to deduce conclusions" (Slatter, 1987). It is used in such situations as defining configurations of computer systems where the final outcome is not specified in advance (deductive reasoning). Reasoning with uncertainty is a method of reasoning when "facts" are less than one hundred percent certain. It may be based on one of several methods for evaluating assertions that have a certainty or confidence factor of less than one hundred percent(e.g., "if it is cloudy there is a fifty percent chance of rain" or "if it is foggy there is a ten percent chance of rain"). Ways of combining weights (cloudy and foggy) include standard Bayesian statistics and domain specific algorithms.

Most of the original systems were coded in the LISP or PROLOG languages. Both are good for manipulating symbols which is a prerequisite for these types of systems. Prolog also has built in logical inference capabilities. Newer commercial expert system tools often include multiple ways of representing knowledge and inferencing. Such systems are referred to as "expert systems shells" in that the programmed infrastructure is in place and domain knowledge is simply added to the shell. Shells are not applicable in all domain areas as the inference process and system design are not always domain independent. Even where shells are applicable, systems development is still not trivial. One cannot overemphasize the time that is usually involved in the complex "knowledge engineering" process of obtaining knowledge from experts in a form that is amenable to representation in typical knowledge base data structures (Waterman, 1986). There have been attempts at automating this process but these are mainly applicable for very simple systems.

Expert systems have been successfully applied in many areas of expertise (e.g., chemistry, computing, geology, law, and medicine). MYCIN (Shoreliff, 1977) was one of the first expert systems; it diagnosed bacterial infections and recommended appropriate doses of specific antibiotics. Although it was an early system, it inspired much of the research into the potential use of such systems in education. Researchers reasoned that if the medical expertise necessary to solve problems in a domain could be represented in a computer, this representation could form the basis for a tutorial program for teaching such expertise.

Education and Artificial Intelligence

Researchers in the field of ITS's apply artificial intelligence techniques, such as the knowledge representation and inferencing in expert systems, to computer based education and training. These techniques allow the development of computerized learning systems that are more adaptive to the students needs than are systems based on more standard computer programming techniques. One of the earliest systems developed in the late 70's was based on the MYCIN system. This system, called GUIDON (Clancey, 1983), used the expertise represented in MYCIN as a basis for instruction. It had a number of problems such as ineffective tutorial strategies but spurred further research into the application of artificial intelligence techniques to education (Wenger, 1987).

GUIDON and other prototypes were developed through the late 70's and early 80's but the number of functioning ITS's remained very small, as development was limited by the large computational demands of such systems. The single textbook (Sleeman & Brown, 1982) written in the field until 1987 indicates the level of activity. However, in recent years general acceptance of computing technology, efforts to increase cost-effectiveness through automation, a desire to connect education to industry more directly, and the improved cost-performance of hardware have made ITS's more feasible and research in the ITS area has expanded rapidly. This expansion is manifested by the several texts published on the ITS field since 1987 and a biennial international conference established in 1987 concerned with ITS design and development, There is now a journal (Artificial Intelligence and Education) dedicated to this area as well as frequent articles in other journals.

A TYPICAL SYSTEM

There has evolved a generally accepted architecture of an ITS that has endured with little change until quite recently (Self, 1989). This typical ITS (Burns & Capps, 1988) includes four modules: the expert module (a representation of the subject area knowledge that the student is to learn), a student model module, a tutorial module, and the student-machine interface module. The interaction of these modules controls the students instruction or learning environment. Along with this typical architecture there are some general software languages/tools and typical hardware environments which are used for development and delivery.

Expert Module

The expert knowledge is generally procedural, that is, consisting of a set of actions to be performed given the prerequisite conditions amenable to rep resentation in "if.. . then.. ." rules. For example, in GUIDON the expertise is a body of rules for diagnosing bacterial infections and prescribing antibiotics. Very few systems involve declarative knowledge (e.g., knowledge of the form 'X is a Y," or "B has a C"). One example of a system representing declarative knowledge is SCHOLAR (Carbonell, 1970) which represents geographic knowledge in a semantic network. In such a network knowledge is written as a set of "nodes," representing concepts, connected by "links," representing relationships between the concepts; e.g. concepts "x" and "variable" can be linked by the relationship "is-a." Other systems use combinations of rules and frames. Expert systems have focussed on areas of heuristic reasoning because often algorithmic reasoning can be represented more efficiently in more traditional computer algorithms. However, there is no reason that the expert module of an ITS could not represent algorithmic reasoning of an expert. In the expert module of an ITS, it is the articulation of the expert's knowledge in a form amenable to learning that is important, and not concern with the speed of execution in automating the expert's performance.

As with expert systems in general, there are a number of stages in implementing an expert module. First, one must define the knowledge at the expert level, This involves time consuming interviews with experts. A second stage is determining how best to represent this knowledge in the computer. A third stage is the often arduous implementation. Designing and implementing the expert module has all the tasks associated with developing an expert system (see for example, Walters & Nielsen, 1988) plus the constraints and complexities imposed by the interaction with other modules.

Student Model Module

There are various approaches to representing the state of the student's knowledge in such a way as to aid in diagnosis of student problems and in remediation. A comprehensive student model would include all the student's prior learning that might be applied to the current task, the student's progress within the system and the student's learning style, as well as other types of student related information. Implementing a comprehensive model is a such formidable task that Self (1988) questions whether it is feasible or necessary, and other researchers significantly limit the scope of the student model (Elsom-Cook, 1988). Many systems attempt to model the student only in relation to the knowledge represented in the expert module. A model based on such a comparison is called an "overlay" model (Carr & Goldstein, 1977). The student's knowledge is compared to that in the expert module and the differences comprise what the student must learn. GUIDON represented medical students as an overlay of the rules in the domain module. Thus instruction was be aimed at those rules that the student did not know. An elaboration of the overlay model uses a "genetic graph," a variant of the "semantic network" method, which contains assumptions about the order in which the student develops various aspects of expertise. The student's knowledge is described in terms of the nodes of the graph, and his learningbehaviour in terms of the edges. The student's progress is shown by the paths through the graph (see Wasson & Jones, 1985).

Even more ambitious systems might attempt to implement the student model as a program which can be executed to simulate the student's behaviour. Such a simulation could be used to validate the model and to generate alternative models.

The major criticism of the overlay model is that students do not simply lack concepts or rules, they also have incorrect rules called "mal-rules" or 'bugs" (Brown & Burton, 1978). These are misconceptions or misunderstandings of the domain that lead to incorrect answers to problems. help diagnose and assist in remediation, an addition to the student module referred to as the "bug library' is incorporated into the system. In order to diagnose errors, this module must be able to produce the errors generated by these misconceptions or this specific error must be explicitly represented in the library. The best example of such a diagnostic module is a library of faulty procedures used by students in arithmetic subtraction (Brown & Burton, 1978) which produces many of the errors commonly made by students learning the process of subtraction. While this system led to much research including a "repair" theory of how bugs are generated (VanLehn, 1982), it has not been an unqualified success. Problems with this approach include differentiating random casual slips from instances of "buggy" rules and diagnosing higher order interactions between different bugs.

Representation issues for the comprehensive student model module are a superset of those for the expert module. As well as representing the student's correct domain knowledge it may also represent mal-rules and be able to reflect learning induced changes in both types of these representations.

Tutor Module

ITS research is concerned with defining a tutorial module which will use theoptimalstrategiesandtactics for instructingthestudent. This "tutor" must take into account the target subject matter expertise and the student's current level of knowledge. Different approaches might be more appropriate for different subject matters and levels of expertise. A coaching method using hints and examples combined with exploration of simulations might be better for some physics topics, whereas a very guided tutorial with student exercises might be best for instructing LISP programming. An example of an approach used by an ITS for tutoring the LISP programming language is to define an optimal solution path and guide the student along the path, minimizing deviations from that path (Reiser, Anderson, & Farrell, 1985). Much of the apparent "intelligence" in an ITS has to do with how the tutor module uses the knowledge represented in the student module and the expert module in interacting with the system. Knowledge representation could be in terms of frames (e.g., representing particular student states) and rules (e.g., in situation A take action B).

Student-Machine Interface

This module handles the interface between the student and computer (Miller, 1988). As such it is not unique to an ITS but is a critical component of any successful system. An interface might becommand driven, menu driven or use objects which are directly manipulated as in a "mouse"controlled interface. The design and implementation of an effective interface is a complex task in any situation (Card, Moran, Newell, 1983).

Interaction of Modules

These four modules generate and control the interaction of the ITS with the student. The interaction of the student with the system is through the interface module; the knowledge that is to be learned is in the expert module, the state of the student's knowledge is in the student module, and the method of instruction used is in the tutorial module in terms of computer assisted learning (CAL), this is analogous to the computer generating the programs in response to the student's behavior, instead of all the programs being determined prior to any interaction with the student. The interaction of the ITS and the student is shaped dynamically by the knowledge represented in the four modules. The student model changes with the interaction and in the prototypical ITS architecture is particularly important in determining the ITS's tutorial tactics at any point in time.

Software Languages/ Tools and Hardware Environments

Most systems have been developed in LISP or PROLOG, some in other AI oriented languages such as LOOPS, and OPS5, some in more general object oriented languages such as SMALLTALK and C++, and some even in C. As with expert systems there are shells for ITS development such as DOMINIE (Elsom-Cook & Spensley, 1988) but they are research or prototype systems and have very restricted domains.

Development hardware originally consisted of minicomputers (such as VAX 780) or special LISP processing machines. However, with the rapid growth of hardware capabilities more work has been done on supermicros rated at several MIPs such as SUN3's and SUN4's and even INTEL 80386 machines. Generally for development, hardware requirements include over one hundred megabytes of disk storage and several megabytes of RAM. Some systems have been targeted for delivery on smaller machines such as IBM PC compatibles or Macintosh computers (e.g., Quigley, 1989) but these are a very small subset of the total number of systems developed.

Intelligent Tutoring Systems and Computer Assisted Learning

Park, Perez, and Siedel (1987) presented many dimensions which discriminate between ITS and CAL technology in the early 80s. The three most relevant dimensions of discrimination are the methods of structuring domain knowledge, the process of presenting the knowledge (or the tutorial strategy), and the modelling of the student. ITS's manipulate knowledge using representations by rules and/or frames as contained in expert systems. The knowledge of the domain is represented explicitly outside the controlling program or interpreter. In CAL there is no attempt to represent the knowledge explicitly. Instead, there is a pre-specified series of templates which present subsets of the subject matter in an order determined beforehand by the programmer and the instructional design. Thus in CAL the tutorial strategy is built in on a step-by-step basis by the designer and programmer, whereas in ITS's the designer attempts to give the system the rules and tutorial expertise with which it can react dynamically to the student's actions. Finally, in CAL the student is modelled by quantitative scores or binary judgements of student responses. In an ITS as previously explained, the student is modelled by an overlay of the domain expertise and perhaps by a library of "bugs."

In summary, properly designed ITS's should be more flexible and responsive than CAL systems. However, it is not yet clear that ITS technology is the best solution to instructional problems in all areas. In some domains (e.g., remedial grammar) simple drill and practice may be the best strategy. Furthermore, few if any ITS's have been demonstrated to be successful for other than procedurally oriented tasks (Park et al., 1987; Anderson, 1988).

RESEARCH FOUNDATIONS OF INTELLIGENT TUTORING SYSTEMS

The ITS area is interdisciplinary. Basic and applied research in education, psychology, cognitivescience, and artificial intelligence have contributed to the emerging ITS technology and will continue to play a large role in its maturation As the ITS field develops, other areas such as linguistics (e.g., natural language processing) and anthropology (e.g., the cultural aspects of learning) should actively contribute to the field, however, their direct contributions to date have been very limited.

Education

lb a large extent development of the ITS field has been driven by AI technology rather than by educational needs or research findings. Very little of the knowledge gained from research into "unintelligent"ComputerAssisted Learning has been incorporated into ITS's. These have focused on the representation of domain, tutorial, and student model knowledge with little consideration of factors such as reinforcement and feedback that have been research issues in CAL.

Within the field of instructional design much research has been done on how to organize instructional materials and the learning process to optimize student learning (e.g., Gagne, Briggs, & Wager, 1988 is an example of one approach). However, this large body of research seems to have had little impact on the ITS field (Park, Perez, & Siedel, 1987). According to Wenger (1987), most early systems focussed mainly on intelligent responses to the students actions at a local level. Wenger characterizes these systems as "opportunistic" as opposed to plan-based tutoring architectures which are more in the tradition of much of the instructional design research (Gagne et al, 1988). More recently work by researchers such as Brecht, McCalla, Greer, and Jones (1989), Winne (1988), Derry, Hawkes, and Ziegler (1988) and Woolf (1988) have addressed this issue of using "intelligence" in planning tutorial interactions and curriculum planning.

There is a tradition in education of focussing upon learning environments where the student learns rather than is taught. It has had an large influence in earlier computer based learning systems such as the "microworld" learning systems developed by Papert and his co-workers (Papert, 1981). In these systems learners can create a problem domain and explore it at leisure under self-determined conditions and "construct" their own solutions. In contrast, instead of providing the student with a learning environment, most of the original ITS's are based on a philosophy of representing the expert's knowledge so that it could then be transmitted to students. Recently researchers in the ITS field have acknowledged the need to develop ITS's more in line with the learning environment approach or "microworld" approach (Brown, 1989; Cumming & Self, 1989; , Pea & Soloway, 1988; Self, 1989). Ultimately, while ITS has much to contribute to educational theory development, the ITS field must keep pace with other educational research to remain relevant. At a much more mundane but still critical level there must be more educational evaluation of working systems (Littman & Soloway, 1988) and meta-evaluation of the ITS approach to education.

Cognitive Science and Psychology

The series of versions of ITS systems relating to subtraction (BUGGY, DEBUGGY, IDEBUGGY see Wenger, 1987 for a review) developed by Brown, Burton, and their collaborators has been very influential in the ITS field. These systems assumed that when students learned the basic skills of subtraction many of their errors were due to the use of faulty or incorrect operating rules (e.g., 0 - any number = that number). The goal of the system was to diagnose and remediate these 'bugs'. The systems have had limited success but generated a great deal of useful research. One interesting research finding was the development of the "Repair" theory of how students acquire these bugs (VanLehn, 1982).

A "mental model" (Norman, 1983) refers to a person's internal representation of things with which they interact which provide predictive and explanatory power for understanding the interactions. The term "mental model" is most commonly used in regard to physical devices and systems. The basic concept of a user (or student) having an internal representation with some isomorphic relationship to an external device or subject matter is inherent in the ITS representing the student's knowledge of the area (Kieras, 1988). Research in this area should continue to play a large role in the ITS field.

The research related to J. Anderson's ACT model of human cognition (Anderson, 1983) has contributed directly and significantly to Anderson's ITS's. This theory models human cognition as production systems. Expertise in an area such as LISP programming is then represented as a set of production rules. These rules can be executed to simulate human competence. In the LISP Tutor (Beiser, Anderson, & Farrell, 1985), the domain expertise is represented in just such a fashion and students are tutored to acquire the appropriate rules in the appropriate order.

More generally any new theory of cognition, particularly in relation to the acquisition of cognitive skills and natural language, will be relevant to the ITS field. As well as cognitive research, it is apparent that research from learning theory must be relevant to the design of feedback to the student. Also it seems that as the ITS field becomes more sophisticated, other areas of psychology will become more relevant. For example, as distributed processing and networking become more sophisticated, a "distributed ITS" could interface with groups of students and mediate their interactions. There may well be a a role for social psychology in planning such group learning systems.

Artificial Intelligence

Research into expert systems technology and associated knowledge engineering methodologies has been the most influential upon ITS development. The most immediate impact on the field seems likely to come from research into the area of automated knowledge acquisition,

AI research in 'belief revision" (Vardi, 1988,) aimed at representing human beliefs, is very relevant to ITSs. Since human belief systems appear to be non-monotonic, that is, new information added to the system can invalidate a previously "correct" conclusion which then must be deleted or modified, it is quite a challenge to revise and maintain such systems. However, as pointed out by McCalla (1987), any sophisticated ITS will need to represent such non-monotonic changes in the student model. Research into various types of knowledge representation such as semantic nets and frames as well as the investigation of natural language understanding remain relevant to ITS's. Another area of relevant research work is that on qualitative reasoning systems (Bobrow, 1984). In qualitative reasoning, experts work with non-quantitative models of physical and other systems. For example, a representation of current flow might include the rule that "if the voltage at A is higher than the voltage at B a current will flow from A to B," with no explicit quantitative representation of voltage.

Another potential development of great impact would be the production of 'shell systems" for developing ITS's. However, since the conceptual foundations, thearchitectures, and design methodologies of ITSs are in an early stage of evolution, and there may be no general purpose shells in the immediate future. In fact it may be that the representations of knowledge and tutorial strategies will become so domain specific that there will never be a general purpose, domain independent ITS shell. However, this would not preclude the development of tool boxes for the development of common elements (e.g., genetic graphs) or of shells for domains with common characteristics.

Neural Networks

The connectionist or neural network approach to perceptual and cognitive modelling (Anderson & Rosenfeldt, 1988) is currently of great interest in psychology, cognitive science, artificial intelligence and other areas contributing to ITS development. For this reason it will be discussed in somewhat more detail than its current contributions to ITS's may justify. The connectionist approach has a long history (Rosenblatt, 1958) but remained dormant from the 60's until quite recently after a critique (Minsky & Papert, 1969). An information-processing approach to cognition (e.g., Newell & Simon, 1972) assumes that most human cognition can be modelled by an architecture based on Von Neumann computer architecture. This approach assumes a single, limited capacity, relatively complex central processor which processes symbols serially; a short term memory, a longer term memory, and some mechanism for switching attention or allocating resources. Retrieval from memory is based on a method of specifying memory addresses. All of these assumptions have been questioned at one time or another, but never with more vehemence and to such an extent as with the recent revival of connectionism (e.g., Rumelhart & McClelland, 1986). Aconnectionist model assumes that there are a very large number of simple processing units operating in parallel on input, and that memory is distributed and content addressable. Such a model assumes the simple processing units share some number of inhibitory and excitatory connections. The basis of the model is an extremely simplified view of the behaviour of neurons (hence "neural nets"). Much of the recent revival in interest in this approach is due to the fact that these 'models" can now be implemented in software and hardware. Such implementations have shown many interesting behaviours. In fact, this technology has had some remarkable early successes in pattern recognition (Gorman & Sejnowski, 1988) and transformation (Sejnowski & Rosenberg, 1986).

How much success the connectionist approach will have in modelling high level cognitive processes is currently the subject of intense debate in cognitive psychology and artificial intelligence Pinker & Mehler, 1988; Rumelhart & McClelland, 1986). Regardless of the eventual success or failure of neural net models of cognition, they do not seem directly relevant to the current or emerging generation of ITS's. Neural net models are not articulated in a easily understandable fashion and such articulation of knowledge is the underpinning of this ITS technology. It may be that in the future ITS's based on neural network models of expertise will be implemented, but they will be radically different from today's ITS's. In the shorter term the contribution of the neural net model to ITS design might be the application of their pattern recognition and generation capabilities to improve user interfaces. For instance, neural net technology could be used to provide the capability for recognizing a particular student (perhaps by visual input or input device response patterns, much as old-time Morse telegraphists knew each other's "fist" or key-operating pattern). Even more immediately, neural nets might be used to provide a classification of a student by patterns of response (Beale & Finlay, 1989). It may also be that neural nets will provide general pattern recognizing capabilities tostudents as explicit tools in their learning environment (in the same way databases, spreadsheets, and statistical packages are tools).

Human Factors

Without a good interface, the most sophisticated inferencing systems are going to fail in a general educational setting. ITS's will have to stay abreast or ahead of advances in interfaces available in commercial applications to be successful. Human Factors is a broad interdisciplinary area within which human computer interactions are one focus. Much of the research has been at a perceptual-motor level (e.g., colour and contrasts in screens and keyboard layout; Card, Moran, & Newell, 1983) but there is also considerable research into what are the user's conceptions of systems and the design of more "intelligent" user interfaces (Baecker, 1987; Miller, 1988; Norman & Draper, 1986). Work in this area of cognitive engineering will be particularly relevant to human interface issues.

ITS APPLICATIONS IN EDUCATION

The Benefits of Implementing ITS Technology

The obvious use of ITS technology in education is in intelligent courseware. The need is greatest in areas which are not amenable to text presentation due to requirements for immediate interactive feedback or sensory input such as lab simulations or case studies. It may also be that there are areas which traditionally have not gone beyond text but this is due solely to the limitations of text and lecture media. Computer media may provide breakthroughs in these areas (e.g., one can imagine simulated battles in alternate history scenarios – for instance, what if Hitler had concentrated entirely on the Russian front at the expense of the Italy campaign?).

These systems have the potential to be more responsive to an individual learner's requirements than systems based on printed materials or conventional CAL. The explicit representation of subject matter expertise, tutorial strategies, and student models creates a system which ideally, in a limited domain, can behave as if it "understands" the students' competencies and apply the correct teaching methods without human intervention. ITS's, like CAL, allow self pacing while in general being more responsive and flexible. ITS's can also serve as guides to students exploring online information and knowledge bases.

In classroom educational settings such systems could ease the workload of teachers, thus freeing teachers' time for tutoring students on the more conceptually complex problems. They can provide education in areas where there is a shortage of human expertise. In distance education, where students often rely almost entirely on printed materials, such systems could be surrogates for certain teacher-student interactions.

Other uses of ITS's are computer managed learning and course design (Wipond & Jones, 1988; Winne (1988), online help with computing and data communications systems (Mathews, Biswas, & Neelakandan, 1988), and intelligent guides for knowledge and database exploration. ITS's also could act as repositories for expertise on subject matter tutorial strategies that are not easily stored in text format. As well as specific benefits associated with their ITS's can access the general capabilities of computers (graphics, simulations, data communications, hypertext tools) that are not so well integrated into text-based materials. Finally, developing an ITS can enhance the expert's view of the domain. Investigating the best way to represent the domain and related tutorial strategies in algorithmic and heuristic form is likely to uncover new ways to think about and represent domain knowledge.

Issues in Implementing ITS Technology

There are problems with ITS technology at a number of levels. It is a new interdisciplinary area, and communications between the various contributing disciplines need to be enhanced. Specifically, there needs to be more direct educational input to the field. More systems must be developed for evaluation and more ITS tools must be made available. There must be ongoing applied research at implementation sites. At a more practical level, there are a number of problems with implementing ITS's within any standard educational organization. Although such considerations may not seem germane to academic researchers, these types of problems may be the most difficult to resolve. It is the opinion of some veterans of CAL that the main stumbling block is not instructional efficacy but organizational issues in implementing a new instructional technology in a lecture oriented institution. (e.g., Hunka, 1988).

Zealous promotion of ITS's on their strengths of "intelligence" and flexibility combined with criticism of existing educational techniques may make educational staff see these systems as competing for their jobs and they will resist their implementation. Without enthusiastic cooperation ofstaffand a major training effort, these systems will require the creation of new positions within the institution which may compete with existing positions for funding. Such competition will create more staff resistance to this technology, Career advancement generally is based on existing structures and functions and there currently is little motivation for staff to become involved in development and implementation of these systems. Even without active resistance, ITS's may not fit well with the existing technical infrastructure for production or delivery of educational materials.

Beyond problems with staffing and organizational structure, there are cost issues. It is generally conceded that these systems take significant time to develop (Begg & Hogg, 1987). Although it is logical to assume that costs will decrease after an initial startup, there is no doubt that to implement ITS's on any large scale will be extremely expensive. One scenario would be nationally centralized production, but this might raise other problems related to the standardization of education which runs counter to Canada's currently decentralized and pluralistic approach to educational philosophy and practice. Once developed (or purchased) there would still be implementation costs for software, hardware, and data communications. These systems are very demanding of these resources and any major implementation would require extensive upgrades in even the most computerized institutions. Even with the rapid drop of costs for hardware, these costs would be substantial.

Getting Started with ITS Technology

There will need to be simultaneous acceptance at the grass-roots, support and management levels to implement ITS's on a large scale in an educational institution. Our advice to management is to work on getting the technical infrastructure for all computing related course delivery in place. That is the most important step. Put a delivery system in place. It is critical for staff to have experience with the network, student workstations and simple technology (such as file transfers, editing tools, electronic mail) before implementing advanced instructional technologies. However, at the same time management should encourage staff experimentation with more advanced technologies.

For the staff who wish to experiment, there are several review papers from different perspectives that are good starting points. Woolf (1988) gives a good review of the current status of the field. Seeley Brown (1989) points out recent trends in education that developers of ITS's must take into account. Self and Cumming (1989) provide a similar perspective on what the educational strategy of ITS's should be. For more depth, the beginner should read the following texts. Wenger (1987) reviews various systems in some depth and attempts to provide a framework for conceptualizing the similarities and differences between these systems. This booksupplies a good review of the field since its beginning. A book of readings edited by Psotka (1988) pays less attention to early systems and is less oriented towards AI. Instead it looks in some detail at recent work involving ITS's in areas that must be considered by any developers of ITS's; cognitive science (mental models, problem solving), education (instructional design), AI (knowledge acquisition), and human factors (interface design). It also presents overviews of some of the more recent systems. A book of papers put together as an introduction to the area for the U.S. Armed Forces (Polson & Richardson, 1988) takes an even more pragmatic approach and presents the closest thing to a cookbook for ITS's. It takes a close look at the standard modules of ITS's and other issues (such as evaluation) for would-be developers to consider. While it may be somewhat short on technical detail and somewhat premature in relation to the current status of ITS technology to be a real cookbook, it certainly helps present the field from a pragmatic viewpoint.

From there individuals should review the most recent proceedings from the two biennial conferences dedicated to this area. (More readings are listed in the select bibliography.)

After getting a good overview of the field, there are a myriad of potentially productive paths that an individual could follow. There is basic and applied research required in all of the areas contributing to ITS technology as well as within the area itself, giving researchers a great deal of flexibility in following the path most suited to their aptitudes, needs, and ambitions. Richardson (1988) lists a number current research and development needs and opportunities. For example, one critical research direction for education is determining how to develop successful systems requiring non-procedural knowledge (e.g., declarative knowledge and qualitative reasoning). Currently, ITS technology has not dealt to any degree with subject matter other than very procedurally defined tasks (Anderson, 1988). Another issue particularly relevant to education is how to present different viewpoints of the same domain (Moyse, 1989; Suthers, 1988; Self (1989).

Interested individuals should obtain at least one ITS with which to experiment. More technically oriented individuals should also experiment with one or more shells for ITS or expert systems development. Despite the rapid improvement in technology and assertions about the cost-effectiveness of ITS (Woolf, 1988b), developing these systems is still a complex process at all stages including the implementation. One very good way to appreciate this complexity is to conduct some development work. It may be possible to obtain an ITS shell from researchers in the field. However, since shells specific to ITS development are mostly in a prototype stage, readers may have to start by using commercial shells for expert systems development. Some of these shells are in the public domain or can be acquired for a relatively small sum of money (Lippert, 1987). Whether or not such a shell will be suitable for development of any module of an ITS depends upon the content and design complexity of the subject matter. It is unlikely that an entire ITS can be developed with one. Regardless, such shells are a good starting place for the novice.

REFERENCES

Anderson, J. A., & Rosenfeld, E. (Eds.) (1988). ***Neurocomputing: A reader.*** MA: MIT Press.

Barr, A., & Feigenbaum, E. A. (Eds.) (1982). ***The handbook of artificial intelligence (Vols. I-II).*** Los Altos, CA: William Kaufmann, Inc.

Bierman, D., Breuker, J., & Sandberg, J. (Eds.) (1989). ***Artificial intelligence and education: Proceedings of the 4th International Conference on AI and Education.*** Amsterdam, Holland: IOS.

Boden, M. A. (1987). ***Artificial intelligence and natural man.*** (2nd ed.). New York, NY: Basic Books, Inc.

Born, R. (Ed.) ***(1989). Artificial Intelligence: The case against.*** Beckenham, UK Croom Helm.

Brown, J. S. (In Press). Towards a new epistemology for learning. In C. Fraisson & J. Gauthiers (Eds.), ***Intelligent tutoring systems: At the crossroad of AI and education.*** Norwood, NJ: Ablex.

Chambers, J.A., & Sprecher, J. W. (1983). ***Computer-assisted instruction: Its use in the classroom.*** Englewood Cliffs, NJ: Prentice-Hall, Inc.

Clancey, W. J. (1987). Methodology for building an intelligent tutoring system. In G. P. Kearsley (Ed.), ***Artificial intelligence & instruction: Applications and methods,*** pp. 193-227). Reading, MA: Addison-Wesley Publishing Co.

Duchastel, P. (1988, July). Research directions for ICAI in Canada. ***Canadian Artificial Intelligence/Intelligence Artificielle au Canada, 16,*** 16-19.

Gagne, R., Briggs, L., & Wager, W. W. (1988). ***Principles of instructional design (3rd ed.).*** New York, NY: Holt, Rinehart and Winston.

Gevarter, W. B. (1984). ***Artificialintelligence, expertsystems, computer vin, and natural language processing.*** Park Ridge, NJ: Noyes Publications.

Glass, A. L., & Holyoak, K. J. (1986). ***Cognition.*** (2nd ed.). New York, NY: Random House.

Haugeland, J. (1985). ***Artificial intelligence: The very idea.*** Cambridge, MA: MIT Press.

ITS-88 Montreal: ***Intelligent tutoring systems: Systemes d 'enseignment intelligemment assistespar ordinateur.*** (1988). Montreal, Canada: Universitede Montreal.

Keamley, ***G.*** P. (Ed.) (1987). ***Artificial intelligence and instruction: Applications and methods.*** Reading, MA: Addison-Wesley.

Lawler, R., & Yazdani, M. (Eds.) (1987). ***Artificial intelligence and education. Volume 1: Learning environments and tutoring systems.*** Norwood, NJ: Ablex Publishing.

McCalla, ***G.*** I., & Greer, J. E. (1987). ***Thepractical use of artificial intelligence in automated tutoring: Current status and impediments to progress.*** (Research Report 87-2) Saskatoon, SK: University of Saskatchewan, Department of Computational Science.

Newell, A., & Simon, H. A. (1972). ***Human problem solving.*** Englewood Cliffs, NJ: Prentice-Hall.

Norman, D. (1983). Some observations on mental models. In D. Gentner &A. Stevens (Eds.), ***Mental models*** (pp. 7-14). Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D. & Draper, ***S.*** (Eds.). (1986). ***User centred system design.*** Hillsdale, NJ: Lawrence Erlbaum Associates.

O'Shea, T., & Self, J. (1983). ***Learning and teching with computers: Artificial intelligence in education.*** Brighton, UK: Harvester Press.

Park, O., Perez, R. S., & Seidel, R. J. (1987). Intelligent CAI: Old wine in new bottles, or a new vintage? In G. P. Kearsley (Ed.), ***Artificial intelligence and instruction: Applications*** and ***methods*** (pp. 11-45). Reading, MA: Addison-Wesley Publishing Company.

Pinker, S.,& Mehler, J. (Eds). (1988). ***Connections and symbols.*** Cambridge, MA: MIT Press.

Poison, M. C., & Richardson, J. J. (Eds.) (1988). ***Foundations of intelligent tutoring systems: Lessons learned.*** Hillsdale, NJ: Lawrence Erlbaum Associates.

Self, J. (Ed.) (1988). ***Artificial intelligence and human learning: Intelligent computer-aided instruction.*** New York, NY: Chapman and Hall.

Slatter, P. E. (1987). ***Buildingexpert systems: Cognitiveemulation.*** Chichester, UK: Ellis Horwood Limited.

Sleeman, D. & Brown, J. ***S.*** (Eds.). (1982). ***Intelligent tutoring systems. New*** York, NY: Academic Press.

Swigger, K. M. (1985, March/April). Intelligent tutoring systems: A tutorial. ***AEDS Monitor,*** 23(9-10), 6-9.

Waldrop, M. M. (1987). ***Man-mademinds: The promise of artificial intelligence.*** Rexdale, ON: John Wiley and Sons.

Wenger, E. (1987). ***Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge.*** Los Altos, CA: Morgan Kaufmann.

Woolf, B. (198a). Intelligent tutoring systems: A survey ***Survey lectures from the American Association of Artificial Intelligence*** 1986-1987, pp. 1-43. Los Altos, CA: Morgan Kaufmann.

Woolf, B., & McDonald, D. (1985, March/April). Building a computer tutor: Design issues. ***AEDS Monitor,*** 23(9-10), 10-18.

AUTHORS

Peter Holt is an Assistant Professor in Computer Science at Athabasca University, Box 10000, Athabasca, Alberta TOG 2RO.

Peter Wood is a PhD. candidate in Secondary Education at the University of Alberta.