N89-19854

Intelligent Tutoring Systems Research in the Training Systems Division: Space Applications

J. Wesley Regian, Ph.D.
Air Force Human Resources Laboratory
Training Systems Division

The CBT Continuum. Computer-Aided Instruction (CAI) is a mature technology used to teach students in a wide variety of domains. The introduction of Artificial Intelligence (AI) technology to the field of CAI has prompted research and development efforts in an area known as Intelligent Computer-Aided Instruction (ICAI). In some cases, ICAI has been touted as a revolutionary alternative to traditional CAI. "With the advent of powerful, inexpensive school computers, ICAI is emerging as a potential rival to CAI." (Dede & Swigger, 1987) In contrast to this, one may conceive of Computer-Based Training (CBT) systems as lying along a continuum which runs from CAI to ICAI. Although the key difference between the two is intelligence, there is no commonly accepted definition of what constitutes an intelligent instructional system (VanLehn, 1986).

For my purposes, I discriminate among CBT systems according to the degree to which the instruction they provide is individualized. My choice of this particular dimension is based on more of a desire for utilitarianism than for precision. A great deal of data from the traditional educational world indicates that one-on-one tutoring is superior to both mastery teaching and conventional teaching (Woolf, 1987; Bloom, 1984). Thus, an important way in which CBT systems differ is in the degree to which their behavior is modified by an inferred "model of the student's current understanding of the subject matter." (VanLehn, 1986) The CBT system that is less intelligent by this definition, I conceive of as CAI. Similarly, the system that is more intelligent, I conceive of as ICAI. Often, ICAI systems are referred to as Intelligent Tutoring Systems (Sleeman & Brown, 1982). In this paper, I will refer to a single ICAI system as an ITS, and to multiple ICAI systems as ITSs.

With respect to individualization, it is important to note that virtually all traditional CAI systems are individualized in the sense that they are self-paced, and many are further individualized by virtue of branching routines which allow different students to receive different instruction. CAI systems with branching routines are, in fact, more individualized than those without branching routines. Thus, they are more intelligent by the current definition (although in a weak sense, as we shall see). Nevertheless, in branched CAI the instructional developer must explicitly encode the actions generated by all possible branches, and there is a finite number of possible paths through these branches. As one moves further away from the CAI to the ICAI end of the continuum, one begins to see a very different and more powerful approach to individualization. This more powerful approach is touched on by Wenger (1987) when he refers to explicit encoding of knowledge rather than encoding of decisions (pg. 4). An ITS (which term I reserve for systems which are very far toward the ICAI end of the continuum) utilizes a diverse set of knowledge bases and inference routines to "compose instructional interactions dynamically, making decisions by reference to the

ORIGINAL PAGE IS
OF POOR QUALITY

knowledge with which they have been provided" (Wenger, 1987; pg. 5).

The ITS Anatomy. In an ITS, individualized instruction is an emergent property of several interacting components. ITSs often consist of four, sometimes five, distinct components. These are the expert module, the instructional module, the student model, the interface, and a device simulation when relevant.

The expert module is a programmed representation of expert knowledge in the target domain (that which is being taught). It is almost identical to what is commonly known as an expert system, except in this context it is often very articulate (able to generate some form of rationale for it's actions) and capable of generating alternative solution paths (rather than a single 'best' path). The expert module brings domain knowledge to the ITS. In some useful sense, the system 'knows' how to perform the task which it is seeking to teach, and can demonstrate that knowledge.

The instructional module is a programmed representation of expert knowledge on pedagogy in the target domain. It is generally not articulate but is invariably capable of generating alternative instructional approaches based on the current knowledge level of the current student. While the expert module invariably derives from knowledge engineering with an expert practitioner in the target domain; the instructional module may derive from knowledge engineering with an expert instructor in the target domain (which may or may not be the same person as the expert practitioner), with a general training specialist, or both.

The student model differs from the expert and instructional modules in that it is a mere shell at the beginning of a tutoring session, whereas the latter two are robust and complete when the development of the ITS is complete. At the beginning of a tutoring session the student model is merely a place to store specific kinds of information about students in particular formats that will be useful for the instructional module to access. The student model is dynamically updated during tutoring sessions to maintain current information about the student such as what the student knows, what the student does not know, and misconceptions the student may have. The student module brings situational awareness to the ITS. Thus, the system 'knows' who it is teaching to, and can make informed decisions about how to teach.

The interface provides the methods by which the student interacts with the ITS. The interface may include such output methods as computer generated graphics and text, recorded video images, or speech synthesizers; and such input devices as a mouse, keyboard, touchscreen, joystick, or voice recognition system. One important point about the interface is that it should be as simple as possible so that learning to use the ITS does not interfere with learning from the ITS.

Some of the ITSs developed at the Intelligent Systems Branch (e.g., MITT, MATIE), and many ITSs in general (e.g., STEAMER, IMTS/Bladefold, Sherlock) utilize an embedded computer simulation of an electrical or mechanical device, and thus provide device-specific instruction. The device simulations are used to teach operation or maintenance of a specific device in the context of an operating model of the device. Other ITSs teach a body of knowledge that is not specific to any particular device (e.g., SCHOLAR, LISP Tutor, Smithtown).

Knowledge Engineering. One of the bottlenecks in the development of an ITS is the process that has come to be known as knowledge engineering. The creation of any robust expert system (such as the

ORIGINAL PAGE IS
OF POOR QUALITY

expert module of an ITS) requires a great deal of front-end work before a single line of code is written. The knowledge engineer must first discover how the expert performs the target task. What knowledge is required? How are subgoals defined and achieved? What inferences are made and from what data? For complex tasks, the process is arduous even with a very articulate expert. One of the hallmarks of expertise, however, is a reduced ability to separate and articulate the small steps of a complex cognitive operation (Anderson, 1983). Thus, the process of knowledge engineering tends to be a successive approximation leading slowly toward a complete model of the task. The Human Resources Laboratory (HRL), Naval Training Systems Center, and the Army Research Institute are jointly pursuing a program with the goal of providing tools to support the iterative process of knowledge engineering. The KA (Knowledge Acquisition) toolkit is a software package which allows the user to easily create a flowchart representation of a procedural task. The system then prompts the user to break the chart into smaller and smaller substeps, and to specify the inferences underlying decision points. The end result of a session with the KA toolkit is a running simulation of target task performance that is suitable to support training. The goal of KA is to automatically generate a running expert system from computer-aided knowledge engineering.

ITS Domains. Traditionally (if the term applies to a technology less than twenty years old), ITSs have focused on knowledge-rich tasks such as electronic troubleshooting, physics, economics, and medical diagnosis.

The **Orbital Mechanics** (OM) tutor, currently being developed at HRL, teaches students the device-independent body of knowledge known as orbital mechanics. For example, the Ground Tracks Curriculum Module teaches the correspondence between orbital parameters and ground tracks of satellites. Ground tracks are displays which depict the changing relationship between the surface of the earth and the location of a satellite over time. Understanding this relationship is important for Satellite Operations Officers (2055 AFSC) who monitor and plan satellite missions. Because the OM curriculum is device-independent, the tutor provides appropriate instruction for any task requiring knowledge of orbital mechanics. It does not provide instruction on applying that knowledge in the context of a specific task using specific hardware.

The **Fuel Cell** (MITT) tutor, developed for HRL and NASA by Search Technologies, is an example of a device-specific tutor. MITT provides intelligent maintenance training for the fuel cells on-board the shuttle. This skill is important for Air Force Flight Controllers (20XX AFSC) and for space shuttle crew members. As a device-specific tutor, MITT is targeted for a specific point in the training curriculum of a specific group of students. That point lies midway between basic instruction and expensive simulation. For example, current NASA training for flight controllers involves a general systems course (Phase I), specialized Phase II instruction (e.g., Regency CAI and a workbook for fuel cell specialists), Single Systems Training (SST - individually tutored simulation time), and finally on-the-job training. The Phase II instruction is very inexpensive and very basic, whereas the SST component (also called SST malfunction class) is very expensive to provide at approximately $600 per hour. SST utilizes a shuttle cockpit mockup and allows instructors to introduce various kinds of system failures into the simulation. MITT is an example of a low fidelity simulation-based ITS that is targeted to fill the gap between

ORIGINAL PAGE IS
OF POOR QUALITY

Phase II and SST, so that students can learn a great deal about their task before moving on the the higher-fidelity, more expensive simulation. In this way they can make maximum use of their time on the more expensive, higher fidelity simulation.

Enabling Skills. In many cases, knowledge-rich domains involve components of expertise, sometimes called enabling skills, which can be characterized as high-performance or knowledge-lean components. For example, electronic troubleshooting involves schematic-tracing which is supported by the ability to immediately and accurately combine gate inputs to determine the output of a particular gate type as represented on the schematic. Similarly, expert performance in theoretical physics requires total facility with basic math and algebraic skills. Human instructors can recognize deficiencies in basic enabling skills (especially in one-on-one tutoring situations) and apply methods to correct these deficiencies.

ITSs as a rule are not sensitive to deficiencies in basic enabling skills, even though they are not difficult to identify. Moreover, computers are particularly well suited to providing the kind of drill-and-practice exercises that can correct the deficiencies. For example, in the Air Traffic Control (ATC) training regime for radar operators it is important to be able to visually estimate the angular heading of a radar blip within 5 degrees accuracy. This level of accuracy takes an average of 2000 training trials to achieve. Under normal training conditions, this many trials would require about 5.5 weeks of training time. In a computerized angle judgement module (Regian & Schneider, 1986), students perform a video-flash-card version of the task. In this form, students experience 2000 trials of the critical task in 3 hours.

In generating instruction for knowledge-rich domains, ITSs should be sensitive to the full range of performance determinants for the task, and have appropriate routines available for remediation. Furthermore, there may be a place for ITS technology even in relatively knowledge-lean domains, such as typing, air intercept control, and simple equipment operation. In these cases, knowledge engineering and subsequent knowledge representation for the target task would be relatively simple, since expert performance is defined more by skill than by knowledge. Conversely, knowledge engineering for the instructional module would require more emphasis.

In order to evaluate the utility of ITSs in knowledge-lean tasks, HRL is collaborating with the Southwest Research Institute in developing the **Console Operations** (COPS) tutor for NASA. COPS will be a prototype ITS to support the NASA Flight Control User Tools course, providing device-specific instruction for operators of the Propulsion Console. The first COPS module will teach the major components of the console, and console initialization procedures. Pedagogically, COPS focuses on the instructional principle of cognitive automaticity (Schneider, 1985). This principle describes the acquisition of of cognitive skill with consistent practice. The goal of the first COPS module is to instill facility with the Propulsion Console so that the operator can then focus on the more knowledge-rich aspects of the task. Later COPS modules will teach console reconfigurations that are associated with key mission events (e.g., main engine cutoff, external tank separation, etc.). Currently in the design phase, this system represents a significant departure from the knowledge-rich domains which have traditionally been the focus of ITS development.

Conclusion. The infusion of AI technologies into CBT has the

ORIGINAL PAGE IS
OF POOR QUALITY

potential of producing CBT systems which provide individualized instruction rivaling the quality of one-on-one human tutoring. However, the incorporation of intelligent routines in CBT systems is expensive and should only occur when there is sufficient enhancement of instructional efficiency and effectiveness to offset the additional development cost. HRL is conducting research to help define the role of AI in training, to develop usable and maintainable systems for users, and to advance the spectrum of ITS applicability.

REFERENCES

Anderson, J.R. (1983). The architecture of cognition. Cambridge, MA: Harvard University.

Dede, C., & Swigger, K. (1987). The evolution of instructional design principles for intelligent computer-assisted instruction. Proceedings of the American Educational Research Association.

Regian, J.W., & Schneider, W. (1986). Assessment procedures for predicting and optimizing skill acquisition. Paper presented at ETS conference on diagnostic monitoring, July.

Schneider, W. (1985). Toward a model of attention and the development of automaticity. In M. I. Posner & O. S. Marin (Eds.), Attention and performance XI (pp. 475-492). Hillsdale, NJ: Erlbaum.

Sleeman, D.H., & Brown, J.S. (Eds.) (1982). Intelligent tutoring systems. London: Academic Press.

VanLehn, K. (1986). Student modeling in intelligent teaching systems. Proceedings of the Research Planning Forum for Intelligent Tutoring Systems. San Antonio, TX: Air Force Human Resources Laboratory.

Wenger, E. (1987). Artificial intelligence and tutoring systems. Los Altos, CA: Morgan Kaufman.

Woolf, B. P. (1987). A survey of intelligent tutoring systems. Proceedings of the Northeast Artificial Intelligence Consortium. Blue Mountain Lake, NY. June.