CONFERENCE REPORT

The Third International Conference on Artificial Intelligence & Education

Blaise W. Liffick

The Third International Conference on Artificial Intelligence and Education attracted over 400 participants from all over the world who gathered to present project reports, exchange views, discuss common problems, and establish contacts concerning AI and education This article presents a synopsis of the major presentations and an overview of the conference as a whole

The Third International Conference on Artificial Intelligence and Education was held at the University of Pittsburgh, Pittsburgh, Penn., 8-10 May 1987. The welcome was given by University of Pittsburgh President Wesley Posvar. The conference cochairmen, Stellan Ohlsson and Jeff Bonar, also gave brief welcomes to the participants.

The relatively small size of the conference, about 425 participants, was undoubtedly in part responsible for the congenial ambiance of the meeting. In addition to the opportunity to reunite with old friends, it was easy to establish new relationships with nearly everyone at the conference. With so many attendees from abroad (The Netherlands, Japan, Canada, West Germany, England, Sweden, France, and Hong Kong were all represented by speakers), the international flavor of the conference was well established.

The conference did get off to a somewhat unfortunate start, however, when it was learned that the opening speaker, John Seely Brown, would be unable to attend because of illness. Through a message transmitted by way of Jeff Bonar, he assured the group that although his difficulty required hospitalization, his problem was not serious, and he would be up and around in no time. The obvious disappointment of the audience could be felt.

Other talks were quickly shuffled around, resulting in Elliot Soloway giving the opening address, "Programming as Artifact Design." This change worked out well because Soloway acted like a cheerleader, getting the crowd fired up about the subject of AI and education. Although he claimed that he is not a religious zealot about

programming per se, Soloway gave the impression of a preacher passionate about his beliefs. As Soloway described the changes in what he felt was important to study, from problem-solving skills three years ago to the construction of mechanisms and explanations last year to the design of artifacts today, he was clearly giving witness to his own enlightenment. Such confessions are rare, yet quite valuable even in the research world.

Soloway's main message was that the field has placed too much emphasis on the mythical transference of programming skills to other domains. Much conflicting evidence exists about transference, leading Soloway to conclude that transference is not the ultimate goal for teaching and tutoring programming. Instead, the concern should be for the development of synthesis skills and "high-order doing" skills because today's generation cannot avoid the inevitable interaction with computers.

Soloway suggested a five-stage model of design as part of his concerns about synthesis skills: (1) the understanding of the problem; (2) decomposition of the problem into goals, plans, and objects (Soloway's favorite mechanisms); (3) recomposition of the goals, plans, and objects; (4) implementation; and (5) reflection. This model does not vary significantly from standard software engineering principles and, indeed, is well known to anyone professing to be a scientist of any kind. However, instead of requiring these steps be followed in a strict order, Soloway contends that the way real programmers work best is to bounce from one stage to another as the need arises. This pronouncement delighted the audience, perhaps reflecting the general feeling that too

much rigidity has recently been imposed on programmers by the engineering approach.

Soloway's talk was not so much a call to anarchy (although watching him, one could easily mistake him for an anarchist), however, as it was a the differences between beginner and expert. Finally, Wender suggested that more attention should be paid to the difference between an expert in the subject domain versus an expert teacher.

The pessimism expressed by Wen-

... the panel discussion seemed to be concerned with whether the systems discussed were, in fact, doing AI yet. It was suggested by one participant that perhaps this emphasis is wrong. He asked if our concern shouldn't be with whether we are doing teaching yet.

challenge to the computer science community to develop higher-level constructs with which to tell the computer what is to be done. The emphasis should be on synthesis skills for designing, generating, and evaluating alternative artifacts that perform the desired functions. This emphasis will require a complete revamping of what we think of as a programming language.

Although Soloway's flamboyance and natural exuberance did not transfer to some of the other speakers during the conference (although it was easy to spot his former and current students among the speakers), his message clearly did. Karl Wender, in "Psychological Considerations for the Design of Tutorial Systems," paid tribute to Soloway in his opening by emulating one of Soloway's manic diagrams Wender went on to eschew the use of the infamous "boxes and arrows" so often used in diagrams during presentations.

Wender was a bit more pessimistic than Soloway, however, and bemoaned the fact that little progress has been made in the last 15 years in the development of predictive theories for constructing student models. However, he went on to state that what we do know about the act of programming at this point can be described in three levels: planning knowledge, algorithmic knowledge, and coding knowledge. With this format as a base, we can study strategies that combine the knowledge from these levels in various ways. In addition, it is certainly necessary to study der was echoed by Ben du Boulay in "What Should a Programming Environment for Novice Programmers Be Like?" Bonar's comment in his opening welcome that we are "on the verge of a breakthrough" in developing tutoring systems concerned du Boulay, who believed that such pronouncements raise expectations unreasonably high. He felt that researchers have forgotten students are more than "disembodied cognitive spirits"; they are human beings. In addition, he felt that tutoring systems to date have neglected the whole in favor of one part. Programming is much more than even Soloway's five-stage model suggests. It includes using debugging strategies, developing algorithm plans, testing, documenting, and proving software; in addition there are ancillary skills, such as editing and running programs, and pragmatics, such as dealing with the hardware.

He went on to enumerate what he feels are outstanding problems that must be taken into account when developing a system for novices. the educational setting; the goals of the learner, for example, grades; the learning styles of individuals; effective issues such as motivation; preconceptions and Pea's notion of "superbugs"; guidance versus exploration; the limited repertoire of actions available to the user; the complexity of the tutoring system itself; and the student's relationship with the tutoring system. Du Boulay went on to discuss some of these issues in the context of teaching PROLOG to novice programmers.

Andy di Sessa, in his talk "Social Niches for Future Software," focused on the need to provide a medium capable of supporting a broad range of activities that promote learning and intellectual development. He demonstrated his points using his Boxer system as an example of the types of tools he felt should be available. Some of the kinds of software he felt should be considered are knowledge spaces, transparent machines, throwaway tools, and mind modelers. He considers current applications to be "the acne on the future face of programming." He also suggested that "current programming is to synthesis as a hammer is to a thumb. Each is as likely to cause pain as [it is] to get the job done." His sentiments were well received by the audience.

Beyond the usual categories supplied by the conference structure, several themes linked many of the papers and presentations. The speeches noted earlier seemed to emphasize the importance of teaching knowledge; that is, where earlier intelligent tutoring system (ITS) work tended to focus on matters of curriculum representation and error diagnosis, this conference saw a great deal of emphasis on designing effective teaching.

Another theme which pervaded most of the presentations was that the scientific method works in the computer field just as it does in the standard sciences. Most projects that were discussed involved developing a theory for tutoring, creating an ITS based on the theory, evaluating the ITS, revising or refining the theory, and building a new ITS. This procedure demonstrates the scientific method perfectly.

Another important theme was that we still needed to extend our box of tools with which to build educational systems. Notable was a paper by Rich Epstein of West Chester University, West Chester, Penn., "A Formal Model for the Design and Implementation of Information Resource Systems." Epstein initially expressed concern that his research lacked an AI component, placing the emphasis instead on the user's initiative to manipulate Epstein's notion of an information resource. However, after hearing speakers such as those noted earlier,

he decided that such an exploration system should indeed be discussed at this conference. Based on the excited comments of many of the attendees both prior to and following Epstein's presentation, his concern was needless.

Many other presenters also had interesting comments on the state of the art of intelligent tutoring and our understanding of the student and the student's needs. Brian Reiser pointed out that novices apparently are not able to reliably predict the result of a particular operator, even after careful coaching to familiarize the student with the operation. In fact, he felt that novices might not even be able to identify whether the current state they are in is related to the goal state they are attempting to attain. Gerhard Fischer felt that AI in education means more than just tutoring systems. He suggested that systems must be able to support notions such as incremental learning and learning on demand and that they must support users involved in their own doing, not the system's. Beverly Woolf will undoubtedly be best remembered for her insight that "tricks used twice become methods; methods used for a year become a theory."

Bill Clancey's banquet presentation, "Qualitative Models and Instruction: An Overview of GUIDON2 Research," explored the notion of what an educational system ought to be like. He suggested that AI is the science of qualitative modeling of events and objects. Using the GUIDON and GUIDON2 projects as reference points, he described how this approach has influenced research. In his systems, the original idea was to instruct interns on how to give diagnoses such as most doctors would give. In other words, the interns were to use the evidence of symptoms and test results to narrow the range of possible diagnoses, eventually coming up with what appeared to be the most likely problem. However, this approach did not seem to account for how experienced doctors actually arrive at a diagnosis. In particular, experienced doctors seem to build a plausible story that accounts for all the symptoms. The doctors are not coming up with a single answer but a

whole explanatory structure. Even though the approach used in GUIDON produced an accurate answer, it was felt that the students were not learning the method of diagnosis as a human doctor actually functions. It is not sufficient that the student gives a right answer; the student must have a detailed sense of how to construct the answer.
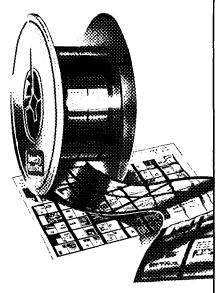
As a conclusion to the conference, Susan Chipman led a panel discussion on instructional principles for ICAI. Panel members were Albert Corbett, Lawrence Frase, Allen Munro, Walter Schneider, and Wallace Wulfeck. Each member gave a presentation, which unfortunately left little time for a discussion of the topics that were covered. Two particular concerns appeared to be central points of the ensuing discussion, however. First, there was concern over whether a cost versus benefit analysis is appropriate when dealing with tutoring systems. This notion led to the most lively debate of the session. Such a pragmatic issue was destined to be a hotly contested topic in a room full of researchers.

The second main issue addressed by the panel summarizes the overall climate of the conference. Many of the comments made by conference participants both during question periods following presentations and during the panel discussion seemed to be concerned with whether the systems discussed were, in fact, doing AI yet. It was suggested by one participant that perhaps this emphasis is wrong. He asked if our concern shouldn't be with whether we are doing teaching yet. Although meant to be rhetorical, his question would undoubtedly have led to a very lively discussion. Unfortunately, it didn't arise until the very end of the panel discussion, at the very end of the conference. It must, therefore, be reserved for discussion at the next conference on AI and education, to be held in two years, the location yet to be determined.

Acknowledgments