Can We Learn From ITSs?

Benedict du Boulay

School of Cognitive and Computing Sciences
University of Sussex
Brighton, BN1 9QH,
U.K.
bend@cogs.susx.ac.uk

Abstract. With the rise of VR, the internet, and mobile technologies and the shifts in educational focus from teaching to learning and from solitary to collaborative work, it's easy (but mistaken) to regard Artificial Intelligence in Education, in general, and Intelligent Tutoring Systems, in particular, as a technology that has had its day — an old solution looking for a new problem. The issues of modeling the student, the domain or the interaction are still very much to the fore, and we can learn much from the development of ITSs.

Despite the changes in technology and in educational focus there is still an ongoing desire for educational and training systems to tailor their interactions to suit the individual learner or group of learners: for example, by being able to deal appropriately with a wider range of background knowledge and abilities; by helpfully limiting the scope for the learner to tailor the system; by being better able to help learners reflect productively on the experience they have had or are about to have; by being able to select and operate effectively over a wider range of problems within the domain of interest; by being able to monitor collaborative interchanges and intervene where necessary; or, most tellingly, by being able to react sensibly to learners when the task they are engaged on is inherently complex and involves many coordinated steps or stages at different levels of granularity. Individualising instruction in an effective manner is the Holy Grail of ITS work and it is taken as an article of faith that this is a sensible educational goal.

This paper explores the question of how much educational difference the "AI" in an ITS system makes compared either to conventional classroom teaching or to conventional CAI methods. One criterion of educational effectiveness might be the amount of time it takes students to reach a particular level of achievement. Another might be an improvement in achievement levels, given the same time on task. So the paper surveys the recent past for ITS systems that have been evaluated against unintelligent versions or against traditional classroom practice and finds cause for optimism in that some of the techniques and solutions found can be applied in the present and the future.[1]

[1] This paper is an edited version of [6].

1 Introduction

In many ways Artificial Intelligence in Education is in a state of flux. People sometimes talk of one of its subfields, Intelligent Tutoring Systems, as an outmoded technology that has, in some sense, "failed" [5]. The emphasis today has shifted to exploring the possibilities of newer technologies such as virtual reality and the Internet, and is particularly concerned with learning environments and collaboration. However most of the traditional hard problems still remain — adjusting the environment to meet the needs of the learner(s), determining what to say to learners and when to say it, and so on.

One aspect of the issue of teaching *vs* learning crystalised into the issue of whether the educational system should attempt to model the student [10]. Modelling the student allows, at least in principle, the system to adjust its behaviour or to react to that student as an individual, or at least as a representative of a class of individuals (see [17]). The argument for not modelling the student arises because it is hard — indeed some regard it as inherently impossible — or because it is thought unnecessary. The argument goes that if a learning environment is well-designed and operated by the students within a supportive educational environment, we can rely on the students themselves to manage their own learning without having the system individualise its reactions in any way.

In some ways the heat has gone out of the debate between the modellers and the non-modellers. Although both camps have coexisted throughout the history of Artificial Intelligence in Education, there is a stronger realisation that both approaches have something useful to offer. Indeed both approaches are now sometimes to be found inside a single system, where an ITS of a traditional architecture may be but a single component of a more general, possibly distributed, system offering the learner a variety of learning experiences from which they can choose [14].

This paper examines what has been shown to be of value in ITS work by briefly exploring the question of how much educational difference ITSs make compared either to conventional classroom teaching or to conventional CAI methods (for more detailed reviews see, e.g. [15, 16]). One criterion of educational effectiveness might be the amount of time it takes students to reach a particular level of achievement. Another might be an improvement in achievement levels, given the same time on task.

A problem for computers and education in general is that it gets hijacked from time to time by particular technologies claiming to produce wonderful educational results simply by virtue of that technology — examples include LOGO, hypertext, and now we have the World Wide Web, hypermedia and virtual reality. It is important to separate reasonable from unreasonable claims and expectations.

To the sceptical eye the evidence for the value of ITSs is not yet overwhelming, though the positive trends are clearly visible. The extra individualisation enabled by an intelligent system does indeed produce educational benefits either through faster learning or through better learning.

This paper starts by exploring the issue of the difference between an intelligently designed system and an intelligent system. It goes on to review criteria by which the educational success of an intelligent educational system could be measured. The paper then examines a number of evaluations of actual systems. Finally it briefly surveys some of the educational issues with which ITS research is grappling.

2 Educational value

It is important to acknowledge that non-intelligent but well-designed systems can be educationally excellent. For example, Dugdale [7] offers a telling account of how quite simple programs can generate authentic mathematical activity, discussion and insight, in particular getting students to reflect on strategy and plans rather than simply following procedures. Her examples have simple interfaces and are not internally complex. They essentially invite users to engage in a problem-solving process that involves only a single step at a time and the systems are able to react to the success or failure of that step immediately. For example, Green Globs, displays coordinate axes and a number of points where the task for the student is to provide a function which intersects and then "explodes" as many of the points as possible. In each case the programs provide visual feedback of success or failure and can adjust, within limited parameters, the difficulty of the tasks that they present, e.g. the Green Globs program can choose locations for the points that can be intersected by simple formulae. However the degree of possible individualisation is slight and one would not regard the programs as "intelligent" no matter how educationally successful they are. It is worth stressing that quite small changes in the way problems are presented and represented can make a big difference in the students' success rates, see e.g. [1]. Such findings suggest that intelligent design on its own is inlikely to get it right for all the students in a target population, and that ideally the system itself needs to have some way of adjusting to the background knowledge and learning preferences of the particular student under instruction.

2.1 Criteria for success

Bloom and his colleagues investigated how various factors, such as cues and explanations, reinforcement and feedback, affect student learning taking conventional classroom teaching as the baseline [2]. They found that highly individualised expert teaching, shifts the distribution of achievement scores of students by about two standard deviations compared to the more usual situation where one teacher deals with a classroom of students. They also found that the range of individual differences reduced.

This two standard deviation improvement, or Two Sigma shift, has become a goal at which designers of ITSs aim. A standard method of evaluation of such a system is to compare it with conventional non-computer-based teaching on the same topic, though there have been some comparisons of "smart" and "dumb" versions of the same software.

2.2 Reducing time on task

Smithtown is a discovery environment with which students can explore problem-solving and inductive learning in the domain of microeconomics [20]. The goals of the system are to help students grasp specific economics concepts, such as the notion of "market equilibrium", as well as more general problem-solving skills such as adjusting only one variable at a time when undertaking an experiment.

Shute and Glaser [20] undertook two kinds of evaluation of the system. One was a comparison with a non-computer-based exploration of the same material; the other was an examination of the particular cognitive and learning-style factors that lead to success with this kind of discovery environment. The comparison study was quite small (N = 30) but found that the group using Smithtown improved their pre/post-test scores as much as the classroom based group despite spending about half the time on the material (5 hours vs. 11 hours).

Over a number of years Anderson and his colleagues have produced a variety of tutoring systems for programming and for mathematics in the heart of the ITS tradition (for an overview, see [4]). Their systems attempt to model the student in detail as s/he undertakes complex problem solving so as to be in a position to offer assistance focussed on the point of difficulty and at the most helpful level of generality ("model tracing"), as well as being able to select problems for the student to solve that move him or her optimally through the curriculum ("knowledge tracing").

One such tutor (LISPITS) taught LISP and has been extensively evaluated in terms of its specific educational interaction methodology (e.g. immediate or delayed feedback) as well as in terms of its overall effect on learning gains. For example, novice programmers using LISPITS were compared to a group working on their own with a textbook and to a group working with a teacher in a conventional classroom manner. While all three groups did equivalently well on the post-test, the group who worked with the human teacher finished in about 12 hours, the group who worked with LISPITS finished in 15 hours and the group who worked with the textbook took 28 hours. The authors argue that the intelligent computer-based system was able to produce similar results to a human teacher and achieved this with in only slightly greater time. In another study with slightly more experienced students, there were two groups both of whom took a conventional LISP course. The control group did the exercises with a textbook and a LISP system whereas the experimental group used LISPITS to do the exercises. As before the LISPITS group finished faster, and this time did better on the post-test compared to the non LISPITS group.

2.3 Improving achievement scores

One of Anderson's more recent evaluations concerns a system designed to be used in Pittsburgh High Schools [8]. The Practical Algebra Tutor (PAT) is designed to teach a novel applications-orientated mathematics curriculum (PUMP — Pittsburgh Urban Mathematics Project) through a series of realistic problems.

The system provides support for problem-solving and for the use of a number of tools such as a spreadsheet, grapher and symbolic calculator.

The intelligence of the system is deployed in several ways. Model Tracing, based on representing knowledge of how to do the task in terms of production-rules, is used to keep close track of all the student's actions as the problem is solved and flag errors as they occur, such as misplotting a point or entering a value in an incorrect cell in the spreadsheet. It also adjusts the help feedback according to the specific problem-solving context in which it is requested. Knowledge Tracing is used to choose the next appropriate problem so as to move the students in a timely but effective manner through the curriculum.

Of special note is the way that attention was paid to the use of the Tutor within the classroom. The system was used not on a one-to-one basis but by teams of students who were also expected to carry out activities related to the use of PAT, but not involving PAT, such as making presentations to their peers.

An evaluation was carried out in three Pittsburgh Public High Schools (N > 100). We should note that the evaluation was of the tutor plus the new curriculum against a more traditional curriculum delivered in the traditional manner. Two standardised and two specially prepared tests were used.

The experimental group performed significantly better than the control group on all four tests but did not achieve Bloom's [2] criterion of improving outcomes by two sigma above normal classroom instruction. However they did perform 1.2 standard deviations better than the control on the specially written Representations Test which was designed "to assess students' abilities to translate between representations of algebraic content including verbal descriptions, graphs and symbolic equations".

Table 1. Comparison of PUMP curriculum plus PAT tutor with traditional curriculum and no tutor. Each cell in the first and second columns contains proportion of the post-test correct (standard deviation) and N. The F values in the third column are derived from a between-subjects ANOVA.

	Control Group	Experimental Group	F value and significance	sigma
Iowa Algebra Aptitude	.46 (.17) 80	.52 (.19) 287	$F(2,398) = 17.0$ $p < .0001$	0.3
Math SAT Subset	.27 (.14) 44	.32 (.16) 127	$F(2,205) = 5.1$ $p < .01$	0.3
Problem Situation Test	.22 (.22) 42	.39 (.33) 127	$F(2,186) = 5.3$ $p < .01$	0.7
Representations Test	.15 (.18) 44	.37 (.32) 124	$F(2,183) = 13.4$ $p < .0001$	1.2

(adapted from [8], page 40).

Lesgold, Lajoie and their colleagues have taken a slightly different approach to individualisation in their work on SHERLOCK 1, a tutor designed to teach to airforce technicians the electronic debugging skills needed to operate a complex piece of testgear. In their system all users worked through the same set of problems but the help and other feedback was adjusted to the expertise of user. Various evaluations of this system are cited by Lajoie [9]. For example, the Air Force evaluation was that "technicians who spent 20–25 hours working with Sherlock 1 were as proficient in troubleshooting the test station as technicians who had 4 more years of job experience". In another evaluation a pre/post comparison was made between a group using the tutor and a control group who carried out their normal troubleshooting duties using the real testgear over a twelve day period. The experimental group solved significantly more problems in the post-test than the control group and the quality of their problem-solving methods was more like those of experts.

3 Smart vs. Dumb

Several studies have compared the effectiveness of intelligent and non-intelligent versions of the same program. For instance, Mark and Greer [13] compared the effects of four versions of the same tutor designed to teach the operation of a simulated Video Recorder. The least intelligent version gave simple prompting and allowed the user only a single way of carrying out a task, such as setting the simulated VCR to record for a particular period at a particular time on a particular channel. The most intelligent, and the one providing the most "knowledgeable" teaching offered conceptual as well as procedural feedback, undertook model-tracing to allow flexible ways of carrying out tasks and could recognise and tutor for certain misconceptions. In a comparative evaluation (N = 76), Mark and Greer [13] found that increasing the knowledgeability of the tutor produced a decreasing number of steps, decreasing number of errors and a decreasing time needed for students to complete the post-test. They also found that these gains were not the result of greater time on task in the case of the most knowledgeable tutor.

Shute [17] evaluated a particular method of student modelling (SMART) which forms the individualising component of a tutor named Stat Lady designed to teach elementary statistics, such as data organisation and plotting. Two versions of the tutor were produced. The non-intelligent version worked through the same curriculum for all learners, with fixed thresholds for progress through areas of increasing difficulty and a fixed regime of increasingly specific feedback when repeated mistakes were made. The intelligent version had a more detailed symbolic, procedural and conceptual knowledge representation which enabled it to provide much more focussed remediation as well as to individualise the sequence of problems for the learner to solve by a more careful analysis of the students' degree of mastery of individual elements of the curriculum.

As with Smithtown described above, Shute [17] was interested not just in the comparative performance of the system but also in aptitude-treatment in-

teractions. The unintelligent version of Stat Lady improved students' scores (N = 103) by more than two standard deviations compared to their pre-test scores. Other studies with the unintelligent version did not produce such high learning gains, but did produce as good outcomes as an experienced lecturer [19] or a workbook [18], though Stat Lady subjects showed a significant gain in declarative knowledge compared to workbook subjects. In another study (N = 168) Shute and her colleagues [19] compared the unintelligent version of Stat Lady to a traditional lecture approach. Stat Lady improved pre-post test score differences by about the same margin as the traditional lecture approach (i.e. about one standard deviation) and over the same time on task (about 3 hours). In a similar study (N = 311) Stat Lady was compared with use of a workbook on the same material [18]. Learning gains were generally similar though Stat Lady subjects showed a significant gain in declarative knowledge compared to workbook studies.

A further study [17] was conducted (N = 100) using the intelligent version of Stat Lady. Pre-post test gains were significantly greater than for the unintelligent version, which themselves were high. However there was a cost in that students spent quite a lot more time working with the intelligent version of the system (mean = 7.6 hours) compared the the unintelligent (mean = 4.4 hours). In general high aptitude subjects gained more from Stat Lady than low aptitude subjects.

In a somewhat similar but smaller (N = 26) study, Luckin compared learning outcomes for versions of a tutor for simple ecology covering topics such as food chains and webs [11, 12]. An unintelligent version (NIS) of her system ECOLAB provided a range of activities, perspectives on the domain, traversal through the curriculum and levels of help wholly under the control of the pupils themselves. The intelligent version (VIS) made decisions in all four of these areas for the pupils based on a student model. As with Stat Lady, the intelligent version produced higher pre-post gains than the unintelligent version, with high ability students gaining more than those of low ability. Time on task was the same for both groups; the gains for both groups were maintained at a delayed (10 week) post-test.

4 Conclusions

ITSs have been designed to individualise the educational experience of students according to their level of knowledge and skill. This paper has described briefly some of the evaluations that have been conducted into the educational benefits of this investment in the capability to individualise. Although the evidence is not definitive, there are indications that the extra individualisation enabled by an intelligent system does indeed produce educational benefits either through faster learning or through better learning. There are also indications that high ability subjects are better suited to this kind of treatment. By contrast, it really would be a surprising finding if attempting to match teaching to the learners capability produced *poorer* results than not so matching. However what has not

been discussed is whether, in practical terms, the effort needed to produce these intelligent systems is sufficiently paid back through their superior performance.

Acknowledgements

I thank Rosemary Luckin for commenting on a draft of this paper.

References

1. S. Ainsworth, D. Wood, and P. Bibby. Co-ordinating multiple representations in computer based learning environments. In Brna et al. [3], pages 336–342.
2. B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6):4–16, 1984.
3. P. Brna, A. Paiva, and J. Self, editors. *Euroaied: European Conference on Artificial Intelligence in Education*, Lisbon, 1996. Edicoes Colibri.
4. A. T. Corbett and J. R. Anderson. LISP intelligent tutoring system: Research in skill acquisition. In J. H. Larkin and R. W. Chabay, editors, *Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*, pages 73–109. Lawrence Erlbaum, 1992.
5. F. M. de Oliveira and R. M. Viccari. Are learning systems distributed or social systems? In Brna et al. [3], pages 247–253.
6. B. du Boulay. What does the AI in AIED buy? In *Colloquium on Artificial Intelligence in Educational Software*, pages 3/1–3/4. IEE Digest No: 98/313, 1998.
7. S. Dugdale. The design of computer-based mathematics education. In J. H. Larkin and R. W. Chabay, editors, *Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*, pages 11–45. Lawrence Erlbaum, 1992.
8. K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8(1):30–43, 1997.
9. S. P. Lajoie. Computer environments as cognitive tools for enhancing learning. In S. P. Lajoie and S. J. Derry, editors, *Computers as Cognitive Tools*, pages 261–288. Lawrence Erlbaum, 1993.
10. S. P. Lajoie and S. J. Derry, editors. *Computers as Cognitive Tools*. Lawrence Erlbaum, Hillsdale, New Jersey, 1993.
11. R. Luckin. 'ECOLAB': Explorations in the zone of proximal development. Technical Report CSRP 386, School of Cognitive and Computing Sciences Research Paper, University of Sussex, 1998.
12. R. Luckin and B. du Boulay. Ecolab: The development and evaluation of a vygotskian design framework. *International Journal of Artificial Intelligence in Education*, 10(2):198–220, 1999.
13. M. A. Mark and J. E. Greer. The VCR tutor: Effective instruction for device operation. *Journal of the Learning Sciences*, 4(2):209–246, 1995.
14. J. Mitchell, J. Liddle, K. Brown, and R. Leitch. Integrating simulations into intelligent tutoring systems. In Brna et al. [3], pages 80–86.
15. J. Self. Special issue on evaluation. *Journal of Artificial Intelligence in Education*, 4(2/3), 1993.

16. V. J. Shute. Rose garden promises of intelligent tutoring systems: Blossom or thorn? In *Space Operations, Applications and Research (SOAR) Symposium*, Albuquerque, New Mexico, 1990.
17. V. J. Shute. SMART: Student modelling approach for responsive tutoring. *User Modelling and User-Adapted Interaction*, 5(1):1–44, 1995.
18. V. J. Shute and L. A. Gawlick-Grendell. What does the computer contribute to learning? *Computers and Education*, 23(3):177–186, 1994.
19. V. J. Shute, L. A. Gawlick-Grendell, R. K. Young, and C. A. Burnham. An experiential system for learning probability: Stat Lady description and evaluation. *Instructional Science*, 24(1):25–46, 1996.
20. V. J. Shute and R. Glaser. A large-scale evaluation of an intelligent discovery world: Smithtown. *Interactive Learning Environments*, 1(1):51–77, 1990.