'Intelligent' Authoring Systems, do they exist?

Dick J. Bierman
University of Amsterdam[1]

ABSTRACT

After an introduction into 'Intelligent' Tutoring systems two types of 'Intelligent' authoring environments are discussed. The first one produces 'Intelligent' Tutoring systems, that is a system with explicit representation of domain knowledge (it can solves problem in the domain), diagnostic knowledge (it maintains a cognitive student-model and knows about malrules, misconceptions and is able to reproduce incorrect solution paths of the pupil), and didactic knowledge (it knows about optimal teaching interventions given a diagnostic context).
The second type of 'intelligent' Authoring system produces traditional CAI-systems. The domain knowledge and didactic knowledge remains to a certain level obscured inside several presentation frames and the branching decisions. The 'Intelligence' of the authoring system would be that it knows about components of such a traditional CAI-program, about recurring structures and preferably how these structures might relate to a global teaching strategy (like mastery learning).
A few recent Authoring Systems for the Macintosh are discussed within these context. It is argued that Hypercard is not an adequate Authoring environment but that a few others exist that might eventually lead to 'Intelligent' Authoring Systems of the second type.

Introduction

Educational Technology has often become a goal in itself. It should be stressed that if a teacher wants to convey certain knowledge to a learner or if a teacher wants to correct false beliefs and misconceptions it is dependent on the educational context what kind of means should be choosen. Very often technological means are overkill (fig.1). This is not to say that Educational Technology is never useful but only to warn for the danger of using the technology because it appears to be fashionable.

[1] Dick J. Bierman, Psych. Lab., University of Amsterdam, Weesperplein 8, 1018 XA Amsterdam, The Netherlands, UUCP: .... mcvax!uvapsy!bierman

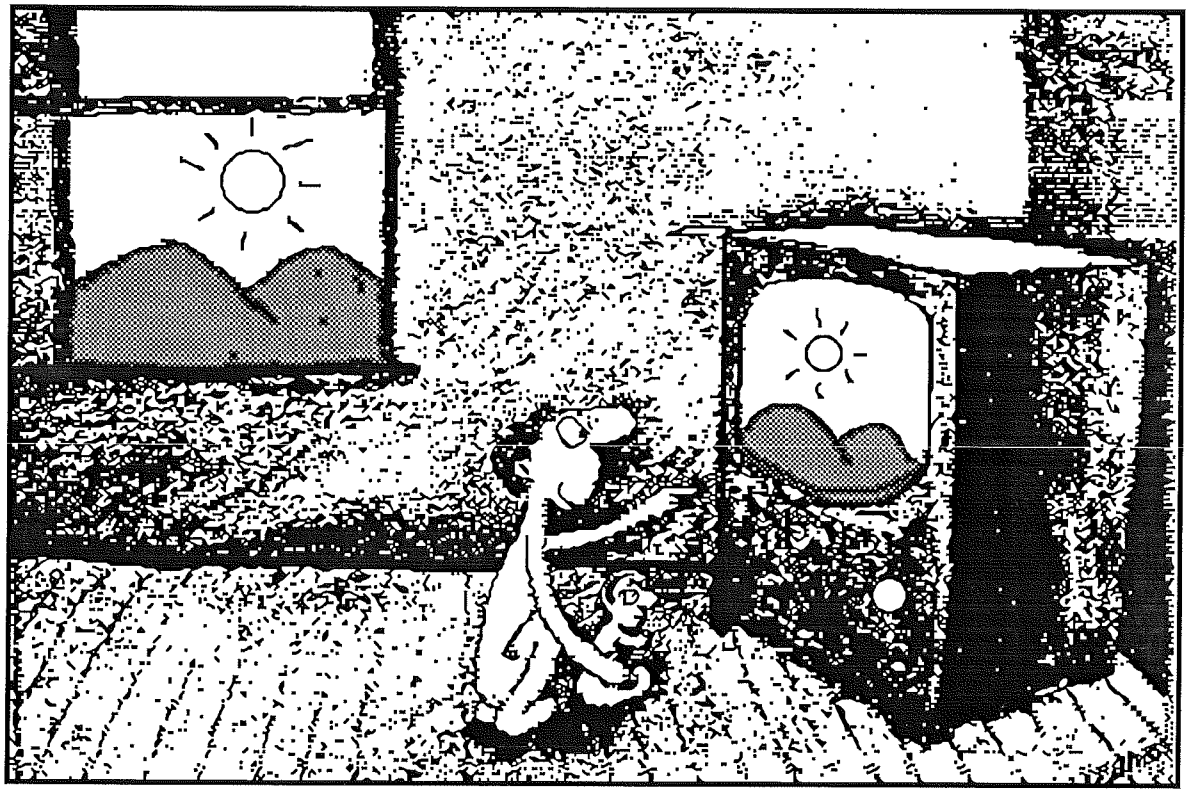

fig.1
Educational Technology ad absurdum

What is an 'Intelligent' Tutoring Systems?

The latest fashion emerging from the AI-community into the world of Computer Assisted Instruction is the appearance of so-called 'Intelligent' Coaching systems. Although there is no consensus on what actually constitutes the 'intelligence' of these systems there appear to be at least two recurrent criteria.

In the first place it is felt that the systems should be experts in the domain that they teach. So at least the system should be able to solve the problems that it requires the student to solve. For instance a system that teaches 'subtraction' should be capable of performing subtraction itself. Of course there exist no computer system that is not capable of subtraction. Therefore the requirement is not only that the system is capable of solving problems in the domain but that it does so in a 'human' way. A trace of the computer actions should resemble a trace of the cognitive actions of a human problem solver. In the case of subtraction one should observe the execution of procedures that correspond to the procedures a pupil would perform. For instance the procedure of 'borrowing'.

In the second place the systems are considered to be 'intelligent' if they can understand the pupils errors. Although 'understanding' is a rather vague concept it is very precisely defined in this context. A system is thought to understand the errors if it is capable of reproducing the error. For instance in the case of 'subtraction' the system could try to solve the problems by applying some wrong rules dealing with 'borrowing'. This could result in wrong answers that match the wrong answers of the pupils.

Due to these two aspects that give these 'Intelligent' Tutoring Systems their 'intelligence' the systems have a radically different architecture (fig.2) from the traditional CAI systems. To begin with , since they have to be experts in their domain, they incorporate a glass-box expert system. The transparency of the expert systems is reflected by the fact that the knowledge about the domain to be thaught and the didactic knowledge are in priciple explicitly represented and therefore inspectable and easy to maintain. This contrast with the traditional courseware where for instance the didactic knowledge is implicit in the branching schemes. Very often a traditional Courseware author is even not aware of the (intuitive) knowledge that is going into his or her courseware.

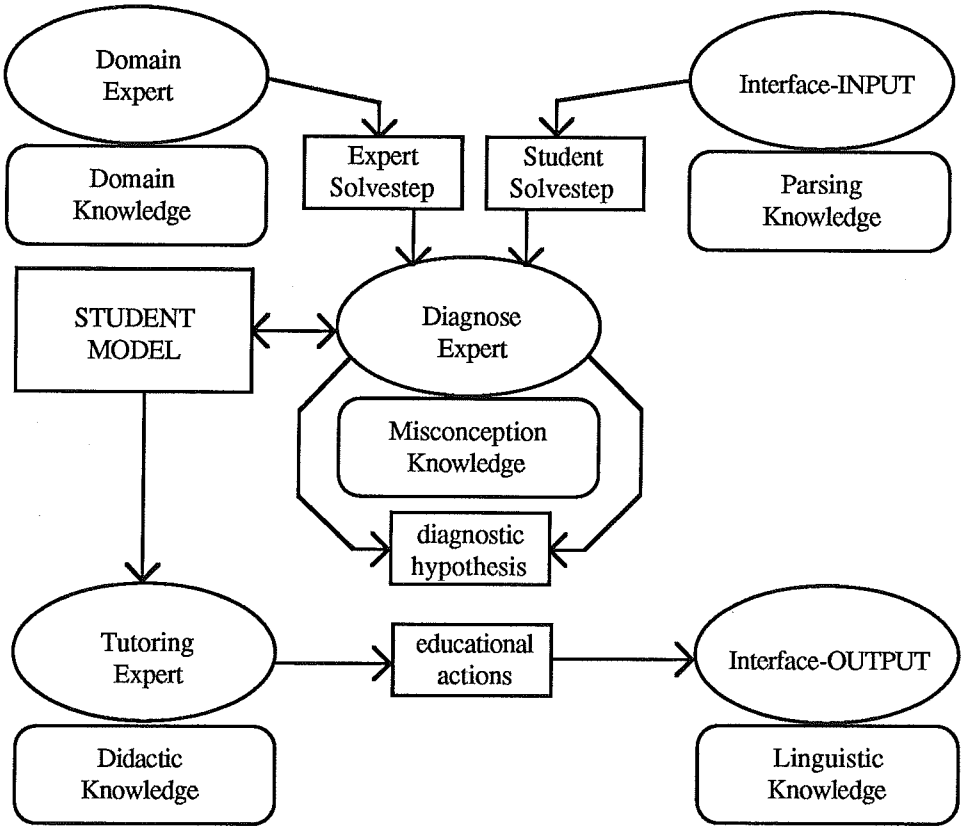

fig.2
Global architecture of the Amsterdam Thermodynamics Coach

Furthermore the core of the architecture is the so called student model. In 'intelligent' systems the student-model is a cognitive model and this model should be 'runnable' if the system is required to be able to reproduce the errors of the pupil. However a part for some simple domains like subtraction no systems with runnable student-models have been implemented yet (this is due to the combinatorial explosion of malrules). So according to this criterion the number of really implemented 'intelligent' CAI systems is quit small indeed.

On the other hand it has been shown that to take an expert system and insert some additional didactic component generally does not yield adequate teaching (Clancey,1982).

The compromise taken in the Amsterdam ThermoDynamics coach project. (Bierman & Kamsteeg, 1986) is to have an expert-system combined with some, not runnable, but certainly cognitive, student model.

Problems with 'Intelligent' Tutoring Systems

A major problem with this approach is the problem of the diagnosis of the student. Since a cognitive model is rather detailed it seems that detailed diagnosis is necessary which might strongly interfere with the ongoing interactions. We solved this problem with a special interaction feature called the 'electronic scratchpad'. A detailed description of this scratchpad and the underlying Object-oriented extension to Prolog which enables easy parsing of the symbolic contents of the scratchpad is given elsewhere (Bierman & Anjewierden, 1986).

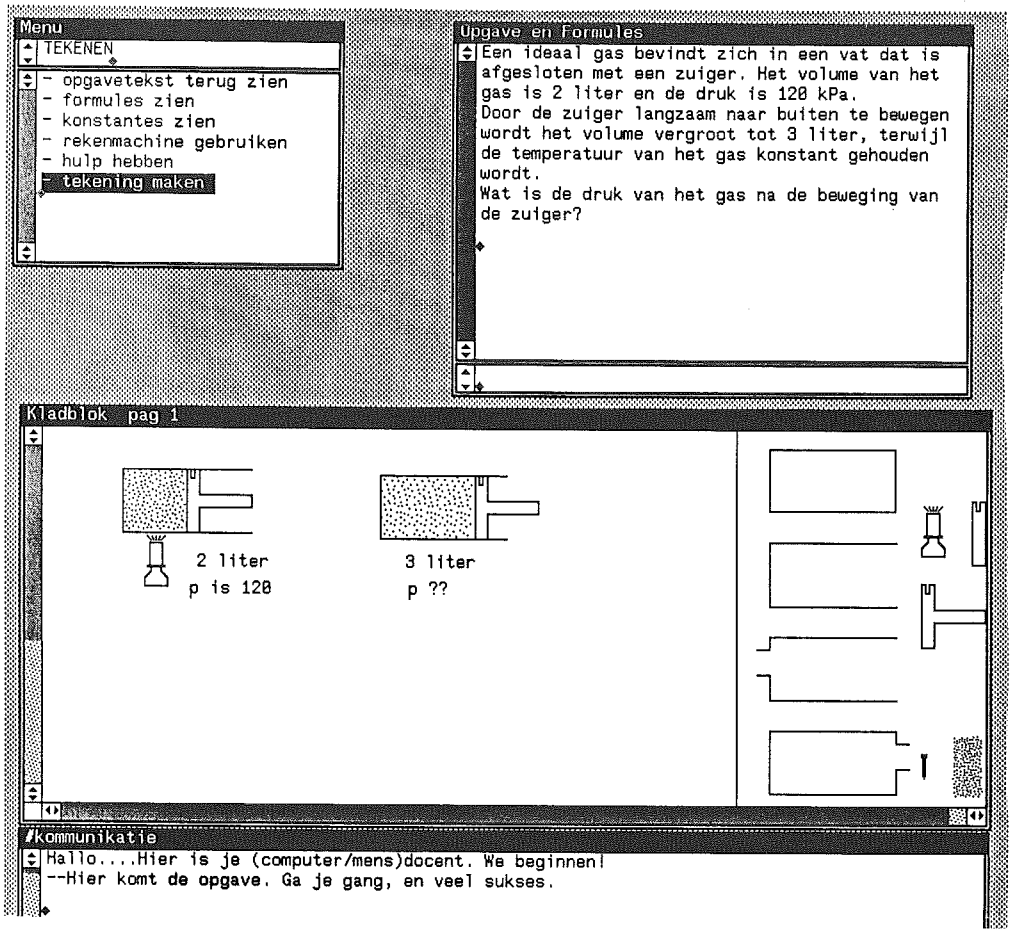

fig.3
'Electronic Scratchpad'

A second more fundamental problem is that the necessary didactic knowledge is not available. The current theories of learning and teaching do not have enough detail to be used as a basis for the implementation of an 'Intelligent' Tutoring Systems. In the Amsterdam Thermodynamics project this problem resulted in the development of a new methodology to elicit this knowledge from expert teachers (Kamsteeg & Bierman, 1986). However it turned out that the amount of implementable knowledge elicited from expert teachers was also rather dissapointing. Although this could be due to the techniuqe that was used used, it was felt that a more reasonable explanation was that teachers do not maintain such detailed cognitive student models. Therefore recent research focusses again on the student and his/her individual learning processes in a 'intelligent' simulation environment (Kamsteeg & Bierman, 1988). The hope is that this research will eventually result in a theory on individual learning which is specific enough to be implemented in an 'Intelligent' Tutoring System.

What is an 'Intelligent' Authoring System?

If there is already a considerable amount of confusion about what constitutes a 'Intelligent' Tutoring Systems this confusion becomes even larger if we talk about 'Intelligent' Authoring Systems ('intelligent' authoring systems).
In principle there are 2 main types to be considered.

a) The first one, let's called it 'Intelligent' Authoring Systems, type-1, helps the author to produce an 'Intelligent' Tutoring Systems.
b) The second one, type-2, helps the author to produce traditional courseware.

One can illustrate this by looking at the knowledge that will eventually be in the Courseware and where it originates.from. Traditionally we have an 'intelligent' author producing Courseware and supplying all the knowledge therein. Now consider the 'Intelligent' Authoring Systems-type1 system. This would produce 'intelligent' courseware.

'intelligent' courseware has the following explicit knowledge components:
1. Domain Knowledge
    a. factual
    b.procedural
2. Didactic knowledge
    a. Diagnostic (Cognitive)
    b. Tutorial
3. Interaction

In a system of the first type only the Domain knowledge originates with the author. The other knowledge components are thought to be more-or-less domain independent and should be a permanent part of the 'Intelligent' Authoring Systems. Back in 1985 a prototype of such a system called 'FITS' was under construction at the University of Essex (Woodroffe, 1985). However, FITS never became quite fit. And that does'nt come as a big surprise. In the first place for domains slightly more complex than 'subtraction' domain-dependent heuristics are necessary to give direction to the cognitive diagnosis. Secondly, as was mentioned before, even if a detailed cognitive diagnosis of the student is available it is just unknown what constitutes an optimal tutorial intervention. Currently a more modest approach for building a shell for 'intelligent' help-systems with limited coaching capabilities is sponsored by the european ESPRIT program (Breuker, Winkels & Sandberg, 1987).But even in this limited approach the requirement of a domain-independent shell creates tremendous problems.

For the second type of Intelligent Authoring systems the situation is rather complex. The system might supply some didactic knowledge in the form of commonly used control structures (branching schemes). This would be knowledge that is generally generated by an didactic expert. For instance a representation of a mastery-learning type of educational strategy. Ideally the system should be capable to derive from domain characteristics and the educationale goals the optimal control structures. This is what contitutes the 'intelligence' of the educational expert. Also the system might help the author in structuring the knowledge about the domain. In fact the system would act as a knowledge elicitation device, asking the user to supply relevant characteristics of the domain and of the educational goals and generating the appropriate control structures.

At the moment these systems do not exist yet. There are two major problems. In the first place the intuitive knowledge about the relation between domain characteristics and teaching strategies as reflected in global control structures is not available. In the second place the methodology to elicit knowledge from an expert is still in its childhood.

However the situation is not hopeless. At the University of Amsterdam a methodology has been developped (Breuker & Wielinga, 1987) which, in the process of building an expert system, enables knowledge engineers to get grips on a domain in a structured way. This methodology will become available in a computer assisted form. Such a computer assisted method for knowledge elicitiation could form the basis of an 'Intelligent' Authoring Systems system of type-2. The system will know a taxonomy of domains with specific characteristics. Research proposals have been submitted to apply this approach to the field of education.
Although there is hope for type-2 systems it will take at least another 4 years before they become available.

Present Situation

So with regard to 'Intelligent' Authoring Systems the message has to be rather pessimistic. However even nowadays new 'dumb' Authoring Systems are appearing which are major improvements over traditional Authoring Systems.

This does not apply to the Hypercard environment which is acclaimed to be the ultimate answer for the future education. The claim is reminiscent to the claim of the LOGO-community (Papert, 1980) back in the seventies. Often even the very same sociological arguments (like permanent education; learning to learn etc. etc) are given. Although

Hypercard is easy-to-use it is, in its rude form, not suitable for guided education. For those contexts where browsing is the optimal teaching strategy (eg. hyperintelligent kids in fact-based domains) it can be, and should be, used. But for nearly all other educational activities a shell should (and probably with an investment of about 1 manyear, will) be made to guide the author in **guiding** the student. One of the oldest principles in teaching is that the new material should be based upon what the learner already knows. Hypercard does not stimulate the diagnostic aspect of teaching. In fact, if one uses Hypercard in its simplest form, it is one super GO TO statement. And in that form it violates nearly every known educational principle (Ohlsson, 1986).

There are however two Authoring Systems on the market which are rather similar in their appearance namely CourseBuilder and Course of Action.(CoA; further reference will be to CoA because I am most familiar with that Authoring Systems)
These systems are not 'intelligent' at all. Not of type1- and not of type-2. What they do however is to stimulate the intelligence of the author who is using the system. In both environments the courseware is produced by directly building a flow-chart of the Lesson. Thus a structured approach.is stimulated in a natural way. One can save these structures in 'models' and re-use the same global branching scheme in other lessons. In this way the author is able to make a library of models each representing some teaching strategy. (Actually inspection of these models might teach us a lot about relations between domain characterisitics and teaching approaches, thus could be helpful in the development of 'Intelligent' Authoring Systems-type2). Interestingly there are no 'jump' possibilities in CoA. This might look at first sight as a real draw-back . However it might reflect the desire of the people who designed this Authoring Systems to make the author think about what he/she is doing If one allows to jump from any node to any node the overview easily gets lost (and so does the student). After some reflection one realizes that within CoA branching back is possible but only if the author has structured the domain in a thoughtful way (for instance as a tree of knowledge nodes). Of course the author has to do this on his own, there is no help offered by CoA like in 'Intelligent' Authoring Systems-type2.

Conclusion

In conclusion it appears that 'Intelligent' Authoring Systems-type2 will become reality and that the present generation of flow-chart based Authoring Systems are a natural way towards there more 'intelligent' systems. They do however require still a number of experts to be in the CAI-development team.

One could ask what about the 'Intelligent' Authoring Systems-type1. I personally doubt whether they will ever become a reality. Not because it is impossible but because it is very doubtful whether the increase of performance of 'Intelligent' Tutoring Systems over traditional Courseware is ever worth the extra effort that has to be spended in the development of these systems. There are strong doubts within the 'Intelligent' Tutoring Systems-community if a detailed student-model might enhance the teaching efficiency in a measurable way (Kelly et al, 1987). Of course these student models are instrumental to bring about understanding of the individual learner. Therefore research on 'Intelligent' Tutoring Systems will continue and in the end might yield spin-off for all educating systems not in the least the human teacher.

References

Bierman, D.J. & Kamsteeg, P.A (1986) *A computer Coach for thermodynamics*. Final report of SVO-project 1039, University of Amsterdam, Institute for Cognitive Studies (ICO), 1986.

Bierman, D.J. & A.A. Anjwierden, (1986), *The use of a graphic scratchpad for students in ICAI*. Proceedings of the 27th ADCIS Conference (pp.68-71), New Orleans.

Bierman, D.J. & Kamsteeg, P.A. (1987), *A methodology for the development of 'Intelligent' Educational Systems*. In (C.Schwarzer & B.Seipp, Eds.) Trends in European Educational Research. p.153-166. Braunschweiger Studieen Reihe, Dusseldorf.

Breuker, J., Winkels, R. & Sandberg, J. (1987). *Coaching Strategies and Tactics for Help systems: Empirical findings and model construction.* Proc. of the 3rd Int. Conf. on AI & Education, LRDC, University of Pittsburgh, May 8, 9, 10 1987. p. 16.

Breuker, J. & Wielinga, B.J. (1987). User models in the Interpretation of verbal data. In A.L. Gibbs (Ed.) *Knowledge Acquisition for Expert Systems, a practical handbook.* Plenum Press, London.

Clancey, W.J., (1982), Tutoring Rules for guiding a case method dialogue. In: D. Sleeman & J.S. Brown (Eds.), *Intelligent tutoring systems*. London: Academic Press.

Kamsteeg, P.A. & Bierman, D.J. (1988) *A Prolog-based Simulation Environment for Physics as a tool for fundamental Educational Research.* Submitted for presentation on the first European Conference on AI and Education.

Kelly, A., Martinak, R. & Sleeman, D. (1987) *How important is diagnosis for Remedition?* A concern for Intelligent Tutoring Systems. In: Heinz Mandl, Angelika Grauer & Emmanual Picard (Eds.) Abstract Volume of the second European Conference for Research on Learning and Instruction. Tübingen, FRG. Sept. 19-22,1987. p123.

Ohlsson, S., (1986), *Some principles of intelligent tutoring*. Instructional Science, **14/3-4,** 293-326.

Papert, S., (1980), *Mindstorms: Children, Computers and Powerful ideas*. New York: Basic Books.

Woodroffe, M.R. (1985). FITS, a framework for an Intelligent Tutoring Proc. of the 2nd Int. Conf. on AI & Education, University of Exeter, Sept. 2&3 1985.