The role of artificial intelligence techniques in training software[1]

Dr. Pierre Dillenbourg[2]
University of Geneva, Switzerland

Introduction

Different motivations led scientists to apply artificial intelligence (AI) techniques to educational software and training software. On one hand, courseware developers were seeking for more powerful techniques for building systems. On the other hand, researchers in computer science and in cognitive psychology found an opportunity to develop and test new techniques or new theoretical models. This second line has probably been the most influential during the eighties. It led to major scientific contributions. For instance, designers transformed the expert system design to develop systems which fulfil the educational functions (explanation, diagnosis, ...) expected in a training software. This work contributed to the elicitation of strategic levels in expertise (Clancey, 1987) and, later, to the emergence of second generation expert systems (Steels, 1990). In others words, research on educational applications helped to develop the methodology for analyzing expertise (knowledge engineering). Similar contributions have been produced in cognitive psychology. The work on learner modelling (trying to infer what the learner knows or misunderstands) has been central to the formalisation and evaluation of cognitive models (Anderson et al, 1989).

This paper does not attempt to review the large number of ideas, techniques or systems developed during the last 15 years of research in 'artificial intelligence and education (AI&Ed). The reader interested in this topic can read Wenger's synthesis (1987), which is not recent but gives an excellent overview of the ideas and principles developed in AI&Ed. We focus here on a body of work which is now rather stable and constitutes the core part of AI&Ed. It that can be summarized in three points:

1. The major contribution of AI to educational and training software is the possibility to model expertise. This expertise is the main feature of AI-based

1 Paper presented at LEARNTEC 1994, Karlsruhe, November 10th 1994.

2 Address: TECFA (Educational Technology Unit), Faculté de Psychologie et des Sciences de l'Education, Université de Genève, 9 route de rize, CH1227 Carouge. Phone: +41.22.705.96.93 Electronic mail: pdillen@divsun.unige.ch

courseware: **the system is able to solve the problems that the learner has to solve**. The system is knowledgeable in the domain to be taught. Of course, other computing techniques can produce a correct solution. The interest of AI techniques is less their ability to produce a correct solution than the way that this solution is constructed. For instance, some complex AI systems have been design to model the resolution of simple subtraction such as '234-98', while any computer language can produce the correct solution (Burton & Brown, 1982).

2. **This modelled expertise enables the system to conduct interactions** that could be not conducted if the system worked with pre-stored solutions. Since artificial intelligence was originally intended to reproduce human intelligence, the techniques available for modelling expertise are to some extent human-like. Actually, the educational use of AI techniques does not require that these techniques are the prefect image of human reasoning. More modestly, it requires that AI techniques support expert-learner interactions during problem solving. Some degree of similitude may be necessary if we want the expert to talk about its expertise in a way which can be understood by the learner. For instance, neural network techniques are considered as a more detailed account of human reasoning than the symbolic techniques used in expert systems. Nevertheless, the use of neural networks in courseware raises the interaction issue: how does the system communicate with the learner about the knowledge encompassed in each of its nodes? **From the courseware perspective, the quality of AI techniques is not their degree of psychological fidelity but the extent to which they support interactions which are interesting from a pedagogical viewpoint**.

3. The types of interactions supported by AI techniques are important for some learning objectives. **These interactions are especially relevant when the goal is to acquire complex problem solving skills.** Other learning objectives can be pursued with simpler interactions techniques, like multiple-choice questions. Since the development of an AI-based software is more costly that standard courseware (especially, those designed with advanced authoring tools), these techniques should be used only when they are really required.

This paper explains these three points, especially the link between the model of expertise and the types of interactions. This link is bi-directional: the model supports some interactions, but, conversely, the targeted interactions impact on the way expertise is represented in the system. We illustrate these principles with examples taken from an imaginary system that would train the employee of a travel agent to plan a trip. We could have selected examples from our own work (Dillenbourg et al, 94), but they are rather complex and require a good understanding of the domain.

Several names are used to refer to educational or training software using AI techniques: intelligent tutoring systems, intelligent educational systems, intelligent learning environments (these names are not synonymous). The term 'intelligent' concerns either the techniques used, either the performance of the system. The latter position being difficult to defend, we make the former position explicit by referring to all these systems as 'AI-based courseware'. These systems are not necessarily more intelligent than others, they simply use AI techniques to support some types of interactions that are interesting for some types of learning objectives

Modelling expertise for interaction

Let us imagine that the travel agency trainee has to solve the following problem:

<u>Problem 22</u>
Mr Hubert wants to go from Geneva to Igloolik, Canada.
His is member of the frequent flyer programme of Swissair
Departure: 1.04.95 Return: 10.05.95

The standard way to do support interactions about this problem would be to store the solution:

<u>Solution 22</u>

Flight SR434 Geneva - Zurich	1.04.94	9 H 40
Flight SR121 Zurich - Ottawa	1.04.94	11 H 45
Flight AC101 Ottawa - Frobisher	2.04.94	9 H 30
Flight AC123 Frobisher-Igloolik	2.04.94	16 H 20
...		

Another approach is to AI techniques to model the reasoning process which led to that solution. Knowledge modelling techniques decompose the reasoning process into numerous intermediate steps or decisions. The knowledge necessary for each step or decision is generally expressed as an 'IF-THEN' rule. Here are a few examples of rules that could be used for solving the problem presented above:

<u>Rule 12</u>
IF the destination is a remote and unknown place,
THEN plan the trip via a major city in the area

<u>Rule 33</u>
IF the local flight is done by a company
THEN plan the international flight with this company

<u>Rule 122</u>
IF the customer is part of the frequent flyer programme of a company and if this company has a flight to the customer's destination
THEN select a flight with this company

Each of these rules encodes a fragment of expertise. The system reasons by checking the 'IF' part of each rule. If the conditions described in the 'IF' part are matched, the system activates the 'THEN' part of the rules. The rules are general: the term 'a company', 'destination' or 'customer' are instantiated by the data of this problem. Some of these rules may be contradictory for some cases, as the 33 and the 122. The system selects the rules and activates them until the problem is solved. In real systems, the rule syntax is of course more formal than the examples above. The systems based on such a set of rules are generally named, knowledge-based, rule-based systems or expert systems. The system component which selects rules, makes the deductions and updates the problem state is called the inference engine.

By decomposing the expertise into fragments, we also decompose the reasoning into intermediate steps. These steps constitute the 'solution path'. The main advantage is that the system can interact about any of these steps: Zurich - Ottawa - Frobisher. For instance, if the learner selects Toronto instead of Ottawa, the system may ask the learner to justify his choice or may present the reasons why it selected Ottawa.

In other words, the first element to be understood regarding the link between AI techniques and interactions is the following: since the reasoning is not treated a black box, but decomposed into intermediate steps, **the expert-learner interactions 'enter' into the solution process**. Conversely, this implies that this approach is not suited to teach elementary skills or perception skills that require an intensive practice (automatization) rather than reflection.

Despite the fact that most expert systems decompose the problem into intermediate steps, they are not necessarily able to interact about each intermediate step. Such interactions require in addition that the representation of the intermediate problem state is accessible to the learner: she must be able to see it and to understand it. Our recent work focused on this aspect (Dillenbourg et al, 1994). The expert rules change the problem representation displayed on the screen. Hence, the learner can see what the expert does and continue on the point where the expert stops. Conversely, the changes that the learner performs on the problem representation are transmitted to the expert, which can as well 'see' what the learner does and continue on the point where she stopped. Our current research aims to improve the quality of these interactions.

Modelling expertise for explanation

The second advantage of AI techniques is that the system can explain its solution. In general, the explanation is the sequence of rules which have been used by the expert to reach the solution (the 'trace' of its reasoning). The system does not present the rule itself, which is often written in a very formal language, but some text comments associated to that rule:

Since Igloolik is a completely remote and unknown place, I plan the trip via a major city in the area: Ottawa (rule 12)

Since the local flight (Ottawa - Igloolik) is done by a Air Canada, I plan the international flight with Air Canada (rule 33)

...

The **delivery** of the expert's trace to the learner raises several issues, namely the question '**do we need to explain everything ?**'. When human beings provide an explanation, they adapt this explanation to the explainee. They especially vary the granularity of the explanation. If the explainer believes that the explainee knows some parts of the explanation, these parts may be presented very briefly or even skipped. Conversely, for the newer, more difficult, or more arguable parts of his explanation, the explainer may enter into more detailed explanations. Several solutions have been proposed to adapt the delivery of the explanation to the learner's level. The simplest one is to deliver the explanation under an hypertext format: the learner may hence decide to browse some parts and read carefully other sections, i.e. to adapt by herself the granularity level to her own needs.

Nevertheless, the problem of explanation goes beyond the simple delivery of text. Since the explanation is the trace of the expert's reasoning, the **quality of the explanation depends on how the rulebase has been designed**. It appeared that some implicit knowledge was often hidden in the rules, especially the knowledge that determines the order of rule selection. In the example above, there is no explicit knowledge that determines whether its is better to first consider the final destination and to work backwards (metarule 13) or to start from Geneva and to consider the flights forward untill Igloolik (metarule 14). In an expert system, this decision may simply result from the order in which the rules are entered in the system. For a human, such a decision may be critical to identify a real expert for this task. Clancey's work (1987) has shown that it is possible to extract this strategic knowledge from the rules and to make it explicit. This strategic knowledge is itself expressed as rules, called **metarules** since they determine the selection of lower level rules.

Metarule 13
IF the destination is a unknown and remote place
THEN select the rules that work from the destination

Metarule 14
IF the destination is a city with a major airport
THEN select the rules that work from the departure

This strategic layer of metarules encompasses the heuristic aspects of the expertise. This makes AI techniques especially relevant when these subtle aspects of expertise are important, and less relevant when the skill to be acquired is a rather trivial sequence of operations. Therefore, AI techniques are better suited for training advanced learners, who have already mastered the basic skills, but have to learn how to integrate these basic skills into an intelligent solution.

Some researchers have gone one step further: for instance, the system AMALIA (Vivet, 1988) uses metarules that prevent the expert to use rules which are unknown by the learner. In our example, we could for instance add a rule such as:

Pedagogical Metarule
IF the learner is a beginner
THEN do not select rules that consider week-end rates

It must be said that these techniques generate explanations which remain less adaptive than human-human explanations. Current researchers (Baker, 1992) considers that an explanation is not a product delivered by one agent to another agent, but as the result of a joint construction. In other words, the explanation functions does not simply influence the reasoning process: reasoning and explanation are viewed as two faces of a same process that must be as interactive as possible.

Modelling expertise for diagnosis

Most courseware includes some information about the learner, for instance her name and her score after the last serie of exercises. A challenge for research has been to improve this information: an 'intelligent' tutor should know what the learner knows or ignores. Learner modelling is the process of analyzing the learner's answers to infer her knowledge. This process is also referred to as 'cognitive diagnosis'. We will not review here the numerous diagnosis techniques that have been developed (Clancey , 1986; Dillenbourg and Self, 1991). We simply present the principles that have been most commonly used and which relys on rule-based models of expertise.

Let us imagine the following rules which determines the minimal time between two flights:

Rule 100
If we plan to consecutive flights
Then set the minimal transfer time to 20 minutes

Rule 101
If the transfer implies to change terminal
Then add 20 minutes to the minimal transfer time

Rule 102
If the transfer implies to pass through custom controls
Then add 40 minutes to the minimal transfer time

The system has to choose which fly to select from Los Angeles to San Francisco:

Flight SR434 arrives in Los Angeles at 5.00, international terminal
Flight Los Angeles - San Francisco leaves at national terminal B, at 5.30, 5.45, 6.00, 6.15 or 6.30

If we execute the rulebase normally, i.e. with the rules 100, 101 and 102 included, the system will set the minimal transfer time to 80 minutes (i.e. 20 + 20 + 40) and choose the flight at 6.30. Now, let us imagine what will happen if we execute an incomplete knowledge base:

- If we remove rule 101 from the rulebase, the system will ignore terminal change, set the transfer time to 60 minutes and select the flight at 6.15.

- If we remove rule 102 from the rulebase, the system will ignore the custom problem, set the transfer time to 40 minutes and select the flight at 5.45.

- If we remove the rules 101 and 102 from the rulebase, the system will ignore terminal change and the custom controls, set the transfer time to 20 minutes and select the flight at 5.30.

Let us summarise this:

rule 101 missing	->	suggests flight at 6.15
rule 102 missing	->	suggests flight at 5.45
rules 101 & 102 missing	->	suggests flight at 5.30

There three suggestions are wrong. **The principle of diagnosis is to compare these simulated wrong answers with the actual answer provided by the learner**. If the learner selected the flight at 5.45, the system will infer that the learner has forgotten to take the custom controls into account (rule 102 missing). If the learner's choice is the flight at 5.30, the system will deduce that she has forgotten both the terminal change and the custom controls (rules 101 et 102 missing). Other potential mistakes can be simulated by adding erroneous rules, called **malrules**:

<u>Malrule 102*</u>
If the transfer involve to pass through custom controls
Then add 90 minutes to the minimal transfer time

If we replace rule 102 by malrule 102* in the rulebase, the system will set the transfer time to 130 minutes and choose the flight at 7.15. If the learner select the plane at 7.15, we can infer that she overestimates the time necessary for custom controls, as expressed by malrule 102*.

These techniques do not imply that the learner's knowledge is stored as rules in their brain. Comparing the system answers to the learner's answers only generates hypothesis about which piece of knowledge might be wrong or missing. AI techniques are used here as diagnosis tools, i.e. as tools for reasoning about knowledge, not as a simulation of the learner real reasoning process.

Many problems have still to be faced. Some mistakes are due to non-systematic causes (distraction). In these cases, AI techniques may lead to an over-interpretation of the learner mistakes. In some cases, the learner misunderstands the domain in a way which is globally different from the expert, and cannot be reduced to a few malrules. Diagnosis techniques are relevant when the system designer knows the most classical mistakes done by learners and can hence write the malrules. They are less relevant in two cases:

- When the difficulty of the topic is its global conceptual understanding, instead of a set of precise concepts and principles, it is difficult to identify the precise source of errors.

- When the targeted skill does not include meaningful mistakes, i.e. when it includes only mistakes that can be pointed to the learner but not explained (e.g. identifying objects on a radar display), such a diagnosis is not useful and can only lead to over-interpretation.

Current research tends to consider that the difficulty of diagnosis can be decreased when one increases the interactions between the expert and the learner, namely if we give to the learner the possibility to correct the eventual misdiagnosis built by the expert.

Modelling expertise for adaptation

Adapting instruction to the learner characteristics has been a permanent concern in courseware design. Adapting the system involves two steps:

1. To identify the learner characteristics.

   In simplest cases, this process is limited to counting the number of correct answers or to recognize pre-defined mistake. In AI-based courseware, the learner characteristic that has received the largest attention is her knowledge regarding the task to be learned. Identifying the learner's knowledge is the diagnosis process described in the previous section. Other learner features, such as the learner motivation, have been neglected (del Soldato, 1993).

2. To take a pedagogical decision on the basis of the information acquired in 1.

   This decision may be choosing a feed-back, selecting an activity, changing a difficulty parameter,... In section 2, we presented some possibilities of adaptation performed in some AI-based courseware. If the system diagnoses that the learner has not acquired the knowledge corresponding to rule-X, the system may decide to provide more detailed explanations when it uses rule-X. Inversely, the system may prefer to avoid using rule-X when it reasons.

Both steps in adaptation are based on the **granularity of rulebases**: expertise is fragmented into small pieces that can combined and used in various ways. The high granularity of rulebases enable support **micro-adaptations**.

Some researchers have paid more attention to **macro-adaptations**. Micro-adaptations are not always sufficient. Sometimes, the learner understands the domain in a way which is globally different from the expert. This difference can not be covered by adding or removing a few rules. In this case, a macro-adaptation is necessary: **to use different rulebases**. For instance, the system WHY (Frederiksen & White, 1988) uses three models to teach how to compute the resistance or the intensity at various points of an electrical circuit:

- the qualitative model reasons in terms of mere presence or absence of current at each point;

- the semi-quantitative reasons in terms of increase or decrease of resistance or intensity at each point;

- the quantitative model applies the laws of physics to determine the exact resistance or intensity at each point.

These three models define a sequence of learning stages. The same principle has been applied to MEMOLAB (Dillenbourg et al, 1994) which is made of three microworlds. In each microworld, the expert and the problem representation are different. Some techniques are proposed (called the 'language shift') to smooth the transition between microworlds (Dillenbourg, 1992). Moving through a sequence of rulebases is useful when the learner progression is not a linear process, but includes qualitative jumps: the discontinuity between knowledge levels cannot be modelled by adding some rules, but by using different rulebases. Regarding to our example, we could design successive rulebases:

Rulebase 1: consider only timetable aspects
Rulebase 2: consider rates (charters, APEX rates, ...)
Rulebase 4: consider availability of seats
Rulebase 3: consider trains for short distances in Europe

In other cases, the different models can be considered as equivalent. They define different viewpoints of the problem which are equally valid (Moyse, 1989) or complementary. For instance, the heart can be studied from an electrical or from an hydraulic viewpoint. Regarding to our example, we could design alternative rulebases:

Rulebase A: view trip planning as a dialogue with the customer
Rulebase B: view trip planning as any kind of planning problem

However, writing a rulebase is a relatively expensive process. It is clear that defining multiple rulebases for a single courseware increases the development cost for this courseware. It is nevertheless useful when the multiplicity of viewpoints in considered as an important feature of the trainee final competence.

Modelling expertise for curriculum planning

In most traditional courseware, the sequences of goals and activities is pre-defined. This curriculum is theoretically based on a detailed analysis of the domain, in which the author identify which concepts or skills are pre-requisite to others. Actually, **writing a rulebase can be viewed as a technique for conducting a detailed analysis of the domain**. Moreover, since the output of this analysis is expressed in a formalism (rules or objects) that can be read by the computer, the scheduling of activities can be done by the system 'on the fly'.

A system able to decide which concept or skill to teach on the basis of the learner behaviour is potentially more adaptive than a system in which the curriculum is fixed once for all. We must acknowledge that the work done at this level has been less developed than for other areas presented above, despite a few significative contributions (e.g. Brecht, 1989). Here are some instances of planning rules that could be used in our travel agent example:

<u>Pedagogical Rule 21</u>
If the learner has shown that she has the knowledge equivalent to rules {a,b,c} and if she has not shown that she has the knowledge equivalent to rule x
Then select a problem that can be solved with rules {a,b,c,x}

<u>Pedagogical Rule 35</u>
If the learner has the knowledge equivalent to malrules m1 and m2, and if concept X appears both in rules m1 and m2
Then show the course unit in which concept X is introduced

As the reader may notice, the rules above are not specific to the training of travel agents. They could be reused to teach other skills in other domains. This enabled us to develop an authoring toolbox proto-type which includes some domain independent pedagogical knowledge (Dillenbourg et al, 1994).

We previously described the possibility to shift between domain rulebases. Some authors (Elsom-Cook et al, 1988; Dillenbourg et al, 1994) have also studied the possibility to shift between pedagogical rulebases. For instance, the two versions of pedagogical rule 10 (below) correspond to different **teaching styles**: rule 10a is very authoritarian, while rule 10b is more flexible. Rule 10a favours immediate feed-back, a basic principle in the behaviourist tradition of educational software, while rule10b implements a delayed feed-back, under the assumption that it is sometime useful to leave the (good) learner the opportunity to find out by himself that he made something wrong..

<u>Pedagogical Rule 10a</u>
If the learner and the expert disagree
Then ask the expert to continue the problem

Pedagogical Rule 10b
If the learner and the expert disagree
and if the learner has been successful before
Then leave the learner continue the problem

Other applications of AI techniques

The previous sections review the application of the knowledge systems, a part of AI which has been intensively studied and can be considered as part of the general computing techniques. Some other techniques have been applied:

- Techniques for **natural language processing** have been developed for building more robust interfaces. Courseware for teaching complex skills is often based on a **mixed initiative dialogue**: the system and the learner can take the initiative in dialogue. For instance, the learner can ask questions such as "what is the difference of voltage between point A and point D?".

  However, advances in the field of natural language understanding have been slower than expected twenty years ago. The techniques are useful for moderately open systems, i.e. when the semantic field is rather narrow and the format of learner's sentence is partly constrained. When communication needs are beyond the available techniques, various solutions are available such as using sentence openers (a list of pre-defined syntactic forms) or using direct manipulation gestures that are somewhat equivalent to natural language.

- A large area of study in AI has concerned **planning**, namely dynamic planning (changing plans during the course of action). Some researchers have applied advanced planning techniques (e.g. blackboard systems) to make the system to base its decision on multiple criteria (Murray, 1989). Other techniques have been used to attempt to infer the learner plans from his behaviour (Greer et al, 1993)

- Another branch of AI, called **machine learning**, has developed inductive, deductive and analogical learning algorithms (Kodratoff & Michalski, 1990). These techniques have been applied to develop self-improving systems, i.e. systems that learn from their teaching experience. For instance, in Proto-Teg (Dillenbourg, 1989), the system recorded when each teaching strategy was effective or not. Then, for each strategy, the system attempted to induce conditions which were met each time the strategy was efficient and were absent each time the strategy failed. As another application for machine learning techniques, we have the systems in which the learner does not interact with an expert, but with a collaborative learner (Chan and Baskin, 1988; Dillenbourg & Self, 1991).

Do we need AI in educational software?

There is no frontier between some courseware that would be intelligent and some that would not. AI-based courseware is not intrinsically better than other kind of courseware. The purpose of this paper is precisely to explain that some types of techniques are useful for some kind of learning activities. Namely, knowledge-based systems enable to interact about intermediate steps in the solution, to provide explanations, make diagnosis and so forth. These techniques are not useful for any kind of learning. We roughly discriminate three categories of goals and systems:

1. Goal : Automating skills
   Systems: Drill and practice

   When the goal is to automate elementary skills, the key factor to success is to provide immediate and accurate feed-back. Standard computing techniques can be used to develop systems in which the learner practices these skills and progressively compiles the associated knowledge.

2. Goal: Acquiring declarative knowledge
   Systems: Frame-based systems

   When the goal is to acquire declarative knowledge, traditional frame-based courseware can be very efficient. A frame-based courseware is a scenario made of frames. Each frame includes some piece of information and some activity, in general a question, a few expected answers and specific feed-back.

3. Goal: Acquiring complex problem solving skills
   Systems: Learning environments

   When the goal is to acquire complex problem solving skills, the key activity is that the learner integrates the various sub-skills by solving realistic problems. A learning environment is generally an open problem situation in which the learner explores the consequences of her actions, namely by observing the system reactions (the system generally includes some kind of simulation). We refer to intelligent learning environments when the learner interacts with some agents about the knowledge she uses in the solution.

If we get back to the travel agent problem, we can find these different types of learning. Some technical skills (computing rates, taking time zones into account, ...) can be practised through series of exercises (drill and practice). The terminology specific to this job could be presented, illustrated and tested by a standard frame-base courseware. Traditionally, the trainee would acquire this basic knowledge during her initial training, by reading books or following a course. Then, she would be asked to put progressively this knowledge into practice. Therefore, she would start working in an office with an experienced worker. This apprenticeship period is necessary because **turning declarative knowledge**

into operational skills and assembling sub-skills into a coherent behaviour constitute indeed a learning process. This is they type of learning which is supported by AI-based courseware, in which the interface imitates the office setting, the tasks to be done and so forth, and the system expert takes the role of the experienced worker.

Synthesis

The original goal of AI was to develop techniques which simulate human intelligence, i.e. which simulate the reasoning process itself or, more modestly, the outcome of this reasoning process . Now, with some distance with respect to this early days of AI, we can say that the role of AI techniques in courseware is to not to simulate human intelligence per se. The techniques are used to support interactions with the learner. Modelling expertise enables the system to 'enter' into the problem with the learner, discuss intermediate steps, explain its decisions, and reasons on the learner's knowledge (diagnosis). **The focus has moved from reasoning AS the learner to reasoning WITH the learner.** This evolution is not in contradiction with studies of human development which tend to consider intelligence not as the result of static knowledge structures, but as a capacity to interact with the our social and physical environment.

References

Anderson, J.R., Conrad, F.G. & Corbett, A.T. (1989) Skill aquisition and the Lisp Tutor. *Cognitive Science*, 13, 467-505.

Baker, M. (1992) The collaborative construction of explanations. Paper presented to "*Deuxièmes Journées Explication du PRC-GDR-IA du CNR*S", Sophia-Antipolis, June 17-19 1992.

Brecht, B. (1989) Planning the content of instruction. *Proceedings of the 4th AI & Education Conference* (pp. 32-41), May 24-26. Amsterdam, The Netherlands: IOS.

Burton, R.R. & Brown, J.S. (1982) An investigation of computer coaching for informal learning activities. In D. Sleeman & J.S. Brown (Eds), *Intelligent Tutoring Systems* (pp. 201-225). New York: Academic Press.

Chan, T.-W. & Baskin, A.B. (1988) "Studying with the prince", The computer as a learning companion. *Proceedings of the International Conference on Intelligent Tutoring Systems* (pp.194-200), June 1-3. Montreal, Canada.

Clancey, W.J. (1986). Qualitative student models, *Annual Review of Computer Science*, 1, 381-450.

Clancey, W.J. (1987) *Knowledge-based tutoring: the Guidon Program*. Cambridge, Massachusetts: MIT Press.

Del Soldato, T. (1993) Domain-based vs. Motivation-based instructional planning. In P. Brna, S. Ohlsson & H. Pain (Eds) *Proceedings of the World Conference on Artificial Intelligence in Education*, AI&Ed 93, Edinburgh, 23 - 27 August 1993. Published by AACE, Charlottesville, VA.

Dillenbourg P. (1989) Designing a self-improving tutor : PROTO-TEG. *Instructional Science*, 18, 193-216.

Dillenbourg, P. (1992) The Language Shift: a mechanism for triggering metacognitive activities. In P. Winne & M.Jones. *Adaptive Learning Environments.: foundations and frontiers*. Springer-Verlag. Hamburg

Dillenbourg, P., & Self, J.A. (1992) A computational approach to socially distributed cognition. *European Journal of Psychology of Education*, 3 (4), 353-372.

Dillenbourg, P., Mendelsohn, P., Schneider, D., & Borcic, B. (1994) Intelligent Learning Environments. In R. Bless (Ed), *Proceedings of the second NRP23 Symposium on Artificial Intelligence and Robotics* (pp. 57 - 74), September 29, Ecublens.

Elsom-Cook, M.T., Spensely, F., Byerly, P., Brooks, P., Mhende, M., Frederici, M. & Scaroni, C. (1988)*Using multiple strategies in an ITS. Proceedings of the International Conference on Intelligent Tutoring Systems* (pp. 286-298), June 1-3. Montreal, Canada.

Fredericksen, J.R. & White, B.Y. (1988) Intelligent Learning Environments for Science Education. *Proceedings of the International Conference on Intelligent Tutoring Systems* (pp. 250-257), June 1-3. Montreal, Canada.

Greer, J.E., Koehn, G.M, & Rodriguez-Gomez J. (1991) A system for exploring plan recognition. In P. Brna, S. Ohlsson & H. Pain (Eds) *Proceedings of the World Conference on Artificial Intelligence in Education*, AI&Ed 93, Edinburgh, pp. 465-472 August 1993. Published by AACE, Charlottesville, VA.

Kodratoff, Y. & Michalski, R.S. (1990) (Eds). *Machine Learning*. (Vol. III). Palo Alto, California: Morgan Kaufmann.

Moyse R. (1989) Knowledge Negotiation Implies Multiple Viewpoints. *Proceedings of the 4th AI & Education Conference* (pp.140-149), May 24-26. Amsterdam, The Netherlands: IOS.

Murray, W.R (1989) Control for Intelligent Tutoring Systems: A Blackboard-based dynamic instructional planner. Proceedings of the 4th AI & Education Conference (pp. 32-41), May 150-168. Amsterdam, The Netherlands: IOS.

Steels L (1990) Components of Expertise. *AI Magazine*, vol.11, 2,pp. 28-49.

Vivet, M. (1988) Reasoned explanations need reasoning on reasoning and reasoning on the student, Workshop IFIP-TC3, Fracati, May 1987, in B. Lewis (Ed), *Artificial intelligence tools in education*, Amsterdam: North Holland.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*, Los Altos: Morgan Kaufmann.