APPEAL: A multi-agent approach to interactive learning environments

APPEAL: A Multi-Agent Approach to Interactive Learning Environments

J. Masthoff & R.Van Hoe

Institute for Perception Research/IPO

PO Box 513, 5600 MB Eindhoven, The Netherlands

E-Mail: masthoff@prl.philips.nl, vanhoe@prl.philips.nl

Abstract

In this paper an agent-based approach to interactive learning environments (ILE) is proposed. It is argued that current interactive learning systems, especially the intelligent tutoring systems, do not satisfy the minimal requirements of an ILE. A specification is given of an agent-based approach to ILE in which situated agents are associated with different aspects of the teacher's behaviour. It is argued that the interaction between these teacher agents and the student agent results in a highly adaptive and interactive learning system that satisfies the requirements of an ILE.The most promising is perhaps that the behaviour of the teacher seems already fairly complex even with a very limited amount of simple behaviours of the agents.

1 Intelligent Tutoring Systems: Adaptive Interactive Learning Environments?

The main problem with traditional computer aided instruction (CAI) systems was that they were unable to provide adaptation or individualization. In response to this and other problems CAI faced, Self (1974) argued that an interactive learning program should contain knowledge of how to teach, knowledge of what is being taught, and knowledge of who is being taught. Carbonell (1970) also argued that the CAI problem could not be solved without the use of AI techniques. Consequently, CAI systems have evolved into what are now usually called "Intelligent Tutoring Systems" (ITS).

A central assumption of ITSs is that the cognitive diagnosis of a learner's errors is an essential step towards meaningful individualized tutoring; the student model is mandatory to the goal of cognitive diagnosis, and consequently to the goal of adaptation. More specifically it is assumed that: 1. student modelling allows feedback adequate to correct any misconceptions or missteps that result in error; and 2. beyond error correction, a dynamic representation of the developing knowledge of each student determines what to teach next and how to teach it (Snow & Swanson, 1992).

In ITSs the student is modelled in terms of the domain expert, either in terms of what knowledge units the expert has but the student does not (overlay model) or in terms of what knowledge units (i.e., buggy rules) the student has and the expert not (process model). In line with the expert system approach, it is assumed that both types of student model can be used with an inference engine.

Although the ITS approach to adaptive training has its merits, there are many problems associated with it. Since current interactive learning environments are developed according to an expert system methodology, the expert module is the central component of those systems.

A consequence of the expert system approach is that the student is modelled in terms of the representation model of the expert module. In the ITS literature (e.g., Wenger, 1987) it is often argued that the expert module should be transparent, so that each reasoning step can be inspected and interpreted by the student. However, it is not at all obvious that the representation model of the expert module corresponds to the cognitive representations of the student. There is indeed empirical evidence that expert knowledge structures do not provide the most useful models for teaching (McArthur et al., 1988; Roschelle, 1990). This implies that student modelling should be conducted in terms of the cognitive model of the student, and not in terms of the expert module.

ITS researchers are trying to develop an adaptive system based on a representation-driven approach rather than a situation-based (data-driven) approach. O'Shea (1979) has argued that one of the most important aspects of any CAI program is its response-sensitivity: an interactive learning program is more response-sensitive than another when it is more adaptive to the individual learning needs of the student than the other system. In a classical ITS system the student's performance is interpreted within the representation model of the expert. Hence the degree of response-sensitivity of the ITS is determined by the extent to which the set of all possible behaviours of the student fits within the representation framework of the ITS. As the student's performance is rarely consistent and it is in fact impossible to predict the full range of student behaviours, the response-sensitivity or adaptivity of ITSs is limited. (see, e.g. empirical research (Payne & Squibb, 1990)). In other words, classical ITSs are not robust as they cannot cope with unexpected, i.e., not predefined, behaviour of the student. Moreover, as further instruction is based on the hypothesized knowledge state of the student and not on the student's behaviour itself, problems can arise because the student's performance was misinterpreted within the expert module.

Last but not least, ITSs are very complex systems which require a lot of implementation time and consume a lot of computing resources. Consequently the performance of ITSs is very poor which makes on-line adaptive training quite difficult.

2 Minimal Requirements for Interactive Learning Environments

In this section, the minimal requirements for an ILE are discussed briefly:

- *Interactivity.* Tackling the problem of the design of an ILE means in the first place addressing the specific problems involved in the development and the use of interactive machines. The interaction in a computer-based learning environment can be considered as a dialogue between two equal and collaborative partners, i.e., the instructional system and the student, where special rules apply in order to promote learning and where the communication goes along a narrow channel, but where also the conventional behavioural repertoire can be exploited (Bouwhuis & Bunt, 1993).

- *Adaptive instruction.* One of the characteristics of the collaborative nature of an instructional system is its adaptivity. At the general level this implies that throughout the instruction session there is moment-to-moment adaptation of the instruction to the needs of the individual student.
- *Robustness.* The interactive system must cope appropriately and in a timely fashion with the always changing behaviour (i.e., learning) of the student, and unexpected behaviour or errors of the student should not lead to total collapse of the system.
- *Direct monitoring of the learning process.* The main goal of the ILE in the tutorial dialogue is to support or optimize the learning process of the student. This implies that the actions of the instructional system are based on a model of human learning.
- *Evaluation research: user-centred approach.* Empirical research is central to the design of ILEs. By analogy with Norman & Draper (1986) we could call this methodology the user-centred design of interactive learning environments. This approach implies that at *all* stages the design process of an interactive learning system should be based on (1) fundamental research on the learning process the system will support, and (2) continual user testing with the prototype.
- *Parsimony.* An ILE should have a simple, but efficient architecture.

3 The APPEAL System

APPEAL (A Pleasant Personal Environment for Adaptive Learning) is an interdisciplinary project which started in 1993 at the Institute for Perception Research (IPO). Persons with a background in computer science, computational linguistics, experimental psychology and phonetics are involved. The main goal of this project is to develop an appealing intelligent learning environment, which meets the requirements mentioned above. Interaction should be more varied and flexible, i.e. highly adaptive to the student's performance. A prototype, also called APPEAL, has been developed. This prototype will function as a research carrier.

The APPEAL system has been designed in a modular way. Its architecture is shown in figure 1. The instruction network contains the course material. Its structure provides information about the relative level of difficulty of the various lessons and exercises. The student history contains a history of the student's performance. The dialogue with the student is determined by the teaching expert. This component represents the general didactic side of teaching. We believe that it is independent of the chosen domain, and that it can be used in various applications ranging from the teaching of language to that of mathematics. The domain model contains information specific to the domain, which, in the system we have implemented, happens to be the language Dutch. The domain expert generates, on the basis of information from the teaching expert, exercises and examples, and evaluates the student's answers. This component represents the domain specific side of teaching. The interface manager provides the interface of the system to the student.

In this paper we focus on the teaching expert which we have implemented using agents. For more detailed information about the other components of the APPEAL system see (Van Hoe et al., 1995; Appelo et al., 1994).

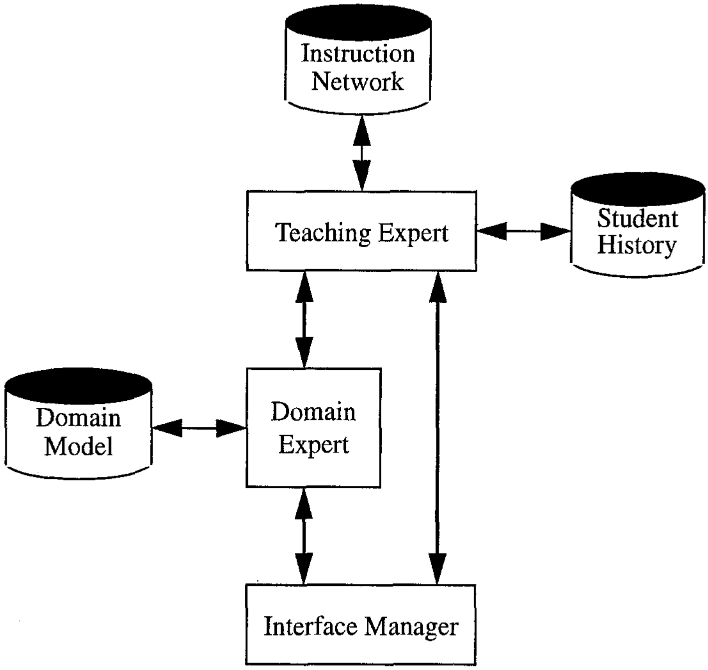

Fig. 1. Architecture of the Appeal system

4 The Situated Agent Approach

The agents in APPEAL are situated agents as in the work of Connah and Wavish (Wavish & Connah, 1990; Connah, 1993) and Chapman and Agre (Agre & Chapman, 1987; Chapman, 1991), and are inspired by the work of Brooks (1986, 1991) and Steels (1989) on autonomous robots, and the work of Suchman (1987) on situated action.

In this section we describe briefly this situated agents approach. For a more detailed overview of the relevance of the situated agent approach for cognitive science in general, and cognitive psychology in particular, we refer to the special 1993 issue of Cognitive Science on situated action.

We can take Shoham's (1993, p. 52) description of agents as a starting point for our discussion of situated agents: "an entity that functions continuously and autonomously in an environment in which other processes take place and other agents exist".

This description needs to be explained and extended as follows:
Situatedness. An agent's most important resource in determining what to do is its immediate situation, rather than internal plans of action;
Interactivity and emergent functionality. The activity pattern emerges from the dynamical interaction between an agent and its environment, rather than being explicitly designed as a property of an agent or the environment. In general, the relation between agents can either be cooperative, to solve a common goal or share a common resource, or competitive.
Autonomy and self-organization. On a micro-level, autonomy implies that the activities of the individual agents do not require constant external guidance. On a macro-level, autonomy implies self-organization, i.e., there is no central authority to control the overall interactions between the agents.
Cognitive economy. An immediate implication of the situatedness principle is the cognitive economy principle. Certain facts relevant to the explanation of the agent's behaviour are not represented by that agent, but are to be found only in the presenting external situation. Agents do not maintain a symbolic model of the world: the world is its own best representation. This implies that perception is central in the functioning of an agent, as it is the agent's only access to the concrete situation.
Parallelism. Agents function continuously, i.e., concurrently with each other, in a multi-agent environment.
Predictability. The environment in which the agents are functioning is partially unpredictable.
Behaviour-based. Agents are described in terms of their behaviour at different levels of granularity in response to other behaviour in their environment rather than in terms of their knowledge.

5 The Teacher Analysed

From an observer's point of view, a human teacher seems to be very complex. That is why we have decided to decompose teaching into smaller components which are easier to think about. Concentrating on the different aspects of a teacher's behaviour which are adapted to the student, we can distinguish the following:

- *Navigation*
  The teacher navigates through the course material, determining at any time which topic to discuss. The speed and order in which the material is taught depend on the performance and interests of the student.
- *Practice*
  Within a given subject, those exercises and examples are given that fit best to the performance of the student.

- *Instruction*
  In the case of bad performance or lack of performance, the student is instructed and helped with his task. The amount of instruction needed depends on the student.
- *Feedback*
  The teacher reacts in different ways to the student. He or she tries to distinguish between errors and slips, and is more or less enthusiastic or disappointed.
- *Presentation*
  Material can be presented in many different ways. For instance in mathematics, depending on the level of the student, additions are presented using formulas, pictures, or even real apples (say).
- *Student model*
  A lot of the adaptations we have just mentioned depend on some knowledge about the history of the student's performance. So, another aspect of the teacher's behaviour is to keep a record of this.

We have associated an agent with each of these different aspects of the teacher's behaviour.

6 Behaviours of the Agents

As a basis for the behaviours of the various agents we drew on the literature and on certain cognitive models. We will now discuss some of the behaviours of the agents and the theories they are based upon.

6.1 Navigation Agent

The behaviour of the navigation agent is based on the notion of controlled complexity as described in (Wood, Wood, & Middleton, 1978). The student should be confronted with problems that lie beyond his current level of competence but not so far beyond that he or she is unable to master the problem being presented. Problems should be neither too easy nor too difficult.

We use rules like

> *If the student's performance is not improving any more and he or she performs sufficiently well then go on to a more difficult kind of exercise or topic.*

and

> *If the student's performance is not improving and he or she does not perform sufficiently well then return to a less difficult type of exercise or topic.*

A history of the student's performance is maintained in the student history to determine whether the preconditions of these rules hold. We will come back to this in more detail when discussing the behaviour of the student model agent.

To determine which new topic or exercise to address or to which to return, information from the instruction network and student history is used. In the instruction network information can be found about which topics or kinds of exercise are considered beforehand to be the more difficult. The student history (as maintained by the student model agent) provides information about which topics and exercises a student has already encountered and his or her performance on these.

6.2 Practice Agent

The navigation agent determines which topic and kind of exercise will be addressed. In most cases however, the topic to be taught consists of a number of items of which the relative difficulty level is not necessarily known beforehand. For instance, with the conjugation of the verb 'to be' as topic, there could be an item for each person. The practice agent decides at any time which item will be presented in an exercise or example.

The most efficient way of teaching seems to be to confront a student with those items he does not know yet. The difficulty is that it is not always obvious what a student knows. As a basis for the behaviour of the practice agent we took quite trivial hypotheses. In the first place, we are convinced that performance of the student in the past predicts future performance. To be more precise, we believe that:

$$P\,(\,incorrect(b,n) \mid incorrect(b,n\text{-}1)\,) > P\,(\,incorrect(b,n) \mid correct(b,n\text{-}1)\,)^{1}$$

In natural language this comes down to: 'the chance that a student answers incorrectly to an item, is greater when he or she answered incorrectly the last time it was presented than when he or she answered correctly'. This assumption led to the idea of using two sets to represent the items to be taught, a 'good' set and a 'bad' set. The good set contains the items correctly answered the last time they were presented. The bad set contains the incorrectly answered ones. Initially all the items to be taught are in the bad set. The assumption implies that the chance that a student does not know an item is bigger when that item is in the bad set than when it is in the good set. So, an item should have a greater chance of being presented to the student when it is in the bad set than when it is in the good set. This is expressed in the following formula.

$$P\,(\,present(b) \mid b \text{ in bad set}\,) = k\,P\,(\,present(b) \mid b \text{ in good set}\,), \quad k > 1$$

1. where incorrect(b,n) stands for an incorrect answer to the n-th presentation of item b

In natural language: 'the chance that an item is presented to the student is k times as great when that item is in the bad set than when it is in the good set, with k greater than one'. When k is very large, an item answered correctly (so, in the good set) is only presented to the student again when the bad set is empty (i.e. all other items have been answered correctly the last time they were presented). If a student never forgot an item that has been answered correctly once, this would be a very good strategy. We have however as a second assumption that a student tends to forget. This is expressed by the formula:

$$P(\ incorrect(b,n) \ \wedge \ \neg slip(b,n) \mid correct(b,n\text{-}1)\ ) > 0$$

The chance that a student answers an item incorrectly, i.e. makes a real error not a slip, is greater than zero even when he or she answered correctly the last time it was presented. This made us choose k not too large. We have chosen k equals 10. Notice that our task sequencing strategy only has one parameter. This makes it far simpler than those existing strategies which also try to take the student's performance into account. We are currently evaluating this strategy and model and comparing it with existing ones.

6.3 Instruction Agent

The instruction agent is an important agent in the system as it not only determines the adaptivity, but also the interactivity level of the system. The behaviour of the instruction agent is determined by the rule "If the student succeeds, when next intervening offer less help. If the student fails, when next intervening take over more control." (Wood, Wood, & Middleton, 1978, p. 133). Furthermore, the instruction agent is based on a layered model of intervention varying between the "general verbal encouragement" level and the "demonstration" level, whereby each layer represents a different level of control from the side of the teacher agent. This implies that the behaviour of the instruction agent is contingent on the performance of the student (cf., situatedness principle); the most appropriate intervention is chosen on the basis of the student's success.

Currently our instruction agent uses a quite simple form of layered intervention, with only two levels. It has, among others, the following rules:

If the student does not act then try to encourage him or her by saying something like 'you must do something to proceed'.

and

If the student even after encouragement does not act then he or she should be helped by making the exercise some what easier.

For instance, in a kind of jigsaw exercise in which the student has to construct a sentence from words, a word is put in the correct place. We intend to extent this behaviour with more layers of intervention.

Essential for the behaviour of the teacher is that out of the cooperation between practice agent (i.e., controlled complexity) and instruction agent (i.e., contingency rule and layered intervention model) a tutorial dialogue emerges with a mixed, varying locus of control: as long as the student cannot master the problems he or she is confronted with, the teacher is more or less in control, once the student succeeds in solving the problem, he or she takes over more control. The layered intervention model of the instruction agent also encompasses different types of ILEs. The 'general encouragement' level corresponds with microworld systems which are user-controlled systems, whereas the 'demonstration' level corresponds with classical ITSs in which control is in the (virtual) hands of the system.

6.4 Feedback Agent

As a first rule we have that feedback should be immediate. For instance, in the before mentioned jigsaw exercise the student gets immediate feedback when he or she tries to put a word where it does not belong. The word bounces back and a sound is produced.

As described in (Norman, 1981) a distinction can be made between real errors and slips, which are errors that occur when someone does an action that is not intended. We have as hypotheses that:

$$P(\, slip(b,n) \,) > 0$$

$$P(\, slip(b,n) \mid correct\,(b,n\text{-}1) \,) >> P(\, slip(b,n) \mid incorrect\,(b,n\text{-}1) \,)$$

Slips occur and most frequently after the same item has been answered correctly before.
These hypotheses led to the following rule:

If an incorrectly answered item belonged to the good set, there is a fair possibility that the student has made a slip. In that case the student will get the opportunity to correct his answer.

The feedback agent also determines the degree of enthusiasm or lack of it with which the student's answer will be greeted. It uses amongst others the following rules:

If the student answers correctly and his or her last answer was correct then give weak positive feedback like 'okay'.

If the student answers correctly and his or her last answer was incorrect then give very positive feedback like 'excellent'.

This last rule is meant to keep the student motivated.

6.5 Presentation Agent

We have not yet implemented the presentation agent. We will briefly describe our views of the behaviour of this agent. The behaviour of the presentation agent can be compared with the task of the director of a television broadcast of a soccer game.[1] The default rule of the director is that as long as the game goes up and down the field, the television viewer gets a general overview of the whole soccer field. However, when a goal is scored the director focuses in on the goal-scorer and replays the goal-scoring in slow motion. As in the case of the broadcast director, the presentation agent tries to choose at each moment in the learning session the optimal representation for the student.

The presentation agent is based on a layered model, whereby each layer represents a different level of abstraction. The general rule determining the behaviour of the presentation agent is that information is presented at the abstract level (cf., the default rule of the director). However, when a student is having difficulties with a particular exercise or is asking for more explanation, the presentation agent can decide to focus in on the problem in another, more concrete representation format. It is obvious that efficient behaviour of the presentation agent will depend on the collaboration with the other agents, especially the student model agent (for information on the performance level of the student) and the feedback and instruction agents (those are the agents requesting a specific representation format).

6.6 Student Model Agent

All the agents mentioned so far, need some information about the history of the student's performance. The student model agent maintains this kind of information. We decided what exactly to record by looking at which information the other agents need. So, student modelling did not become a goal per se, but is only used to supply the other agents with the information they need.
In the first place, the student model agent maintains the good set and bad set needed by the practice agent. The following rules are used for this purpose:

> *If the student answered incorrectly to the item presented, it is added to the bad set and removed from the good set.*
>
> *If the student answered correctly, the item is added to the good set and removed from the bad set.*

For the navigation agent, information regarding which topics and exercises the student has encountered and his or her performance on these is maintained. This means that

---

1. This comparison is due to Mike Graham, Philips Research Laboratories, England.

not only a good set and bad set for the current exercise is kept but for all exercises the student has done so far. The expression 'the student performs sufficiently well', in the precondition of the rules of the navigation agent, has been formalised as 'at least 95% of the items is contained in the good set'. Whether the student's performance is improving is determined by comparing the size of the current good set with the size of that set before a certain number of items has been presented. For this purpose, a history of the number of items in the good set is maintained for the current exercise.

6.7 Postman Agent

An extra agent, called the postman agent, is added to transfer incoming messages from the other modules of the system into a format the other agents can understand.

6.8 Interaction

There is no control component. The behaviour of the system emerges from the interaction between the agents. To give an example of this interaction: the navigation agent, practice agent and instruction agent compete sometimes. When the student is performing badly, the practice agent tries to present more exercises of the same kind to the student. The navigation agent tries to return to an easier kind of exercise or topic, and the instruction agent tries to give extra instruction. Which agent wins eventually depends on the internal states of the agents and the history of the student's performance.

7 Concluding Remarks

The first prototype of our system was ready in december 1993. It is argued that our multi-agent architecture satisfies the minimal requirements of ILEs discussed in section 2. According to an agent-based approach, interactivity and adaptivity are emergent properties of the interaction between the student and the teacher.The persons, with various backgrounds, to which we have shown APPEAL were surprised by the variety of the system's responses and its adequate reactions to the student's behaviour.

As robustness is concerned: in the course of more than 50 demonstrations the system has never behaved in an undesirable way. The system coped even with very unexpected behaviour from the visitors.

The definition of the agents has been and will be based on user-centred research into the interactive learning process. Based on empirical research the behaviours of the agents will be fine-tuned. The system as a whole will be evaluated when the course material has been extended.

It is our experience that the use of agents (the RTA programming language in particular) results in a simple, but efficient architecture. Response times are very acceptable, and the amount of storage space needed for the agents and the student model is limited. It is quite simple to extend the behaviour of the agents, and we plan to do this. To prove the domain independence of the teaching expert, another domain, namely reading, has been implemented.

There is still a lot to be done: the behaviour of the instruction agent should be extended with more layers of intervention, the presentation agent still has to be implemented, the behaviours of the other agents have to be fine-tuned and extended. But it is promising that the behaviour of the teacher seems already fairly complex even with a very limited amount of simple behaviours of the agents. So, we are only at the beginning but we seem to be on the right track.

Acknowledgements

The implementation of the agents has been done in the RTA programming language. RTA is a concurrent and declarative programming language for developing agent-based systems. The language was developed by Wavish and Connah (1990) at Philips Research Laboratories, England.

We would like to thank David Connah for his comments on earlier versions of this paper.

References

Agre, P.E., & Chapman, D. (1987). *Pengi: An implementation of a theory of activity.* Proceedings of the AAAI Conference (pp. 268-272). Seattle, Washington.

Van Hoe, R., Masthoff, J., Appelo, L., Harkema, H., De Pijper, J.R., & Rous, J. (1995). APPEAL: a multimedia environment for adaptive learning. In: *IPO annual progress report 29.*

Appelo, L., Leermakers, M., & Rous, J. (1994). *Language exercise generation in an interactive learning environment.* IPO Manuscript 1012.

Bouwhuis, D., & Bunt, H. (in press). Interactive instructional systems as dialogue systems. In L. Verhoeven (Ed.), *Training for literacy.* Dordrecht: Foris.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation, 2,* 14-23.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence, 47,* 139-159.

Carbonell, J.R. (1970). *Mixed initiative man-computer instructional dialogues.* Doctoral dissertation, MIT, Cambridge, Massachusetts.

Chapman, D. (1991). *Vision, instruction, and action.* Cambridge, MA: The MIT Press.

Connah, D.M. (1993). The design of interacting agents for use in interfaces. In M.D. Brouwer-Janse & T.L. Harrington (Eds.), *Human-machine communication for educational systems design* (pp. 197-206). Berlin: Springer-Verlag.

Green, J.G., Chi, M.T.H., Clancey, W.J., & Elman, J. (Eds.) (1993). Situated Agents [Special issue]. *Cognitive Science, 17.*

McArthur, D., Statz, C., Hotta, J., Peter, O., & Burdorf, C. (1988). Skill-oriented task-sequencing in an intelligent tutor for basic algebra. *Instructional Science, 17,* 281-307.

Norman, D.A. (1981). Categorization of Action Slips. *Psychological Review*, *88*, 1-15.

Norman, D.A., & Draper, S.W. (Eds.) (1986). *User centered system design*. Hillsdale, NJ: Lawrence Erlbaum.

Payne, S., & Squibb, H. (1990). Algebra mal-rules and cognitive accounts of error. *Cognitive Science*, *14*, 445-481.

Roschelle, J. (1990). *Designing for conversations*. Paper presented at the Annual Meeting of the American Education Research Association, Boston.

Self, J.A. (1974). Student models in computer-aided instruction. *International Journal of Man-Machine studies*, *6*, 261-276.

O'Shea, T. (1979). A self-improving quadratic tutor. *International Journal of Man-Machine studies*, *11*, 97-124.

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, *60*, 51-92.

Snow, R.E., & Swanson, J. (1992). Instructional psychology. *Annual Review of Psychology*, *43*, 583-626.

Suchman, L.A. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge: Cambridge University Press.

Steels, L. (1989). Cooperation between distributed agents through self-organisation. *Journal on Robotics and Autonomous systems*.

Wavish, P.R., & Connah, D.M. (1990). *Representing multi-agent worlds in ABLE*. Technical Note, Philips Research Laboratories.

Wood, D., Wood, H., & Middleton, D. (1978). An experimental evaluation of four face-to-face teaching strategies. *International Journal of behavioral Development*, *1*, 131-147.

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