From Intelligent Tutoring to Computerized Psychotherapy

David Servan-Schreiber
Robotics Institute
Carnegie Mellon University
and
Western Psychiatric Institute and Clinic
Pittsburgh, PA 15213

Abstract

Building on the successes and shortcomings of previous experiences with computerized psychotherapy, we have attempted to extend the paradigm of intelligent tutoring systems to the domain of therapeutic interaction. Based on canonical examples, I present three dimensions of the task of tutoring systems: teaching problem-solving vs. domain knowledge; teaching isolated domains vs domains where students have prior misconceptions; teaching with the use of functional models of the domain vs no functional models. I then show how implications of these dimensions have helped us determine the specifications of a tutoring system for sexual therapy. Our approach has consisted of engaging patients in a tutoring dialogue driven by the identification of problem areas and their associated misconceptions. A diagnostic module, implemented as a traditional expert system, uses an extensive bug library to derive an internal model of patients. A dialogue driver relies on a hierarchy of dialogue plans and demons in order to preserve a logical grouping of related topics while remaining flexible to adapt itself, at each level of the dialogue hierarchy, to the unfolding case.

Introduction

Over the last decade, research on computer assisted instruction (CAI) has moved from frame-based systems (traditional CAI) to Intelligent Tutors. The purpose of intelligent tutors is to provide a learning environment that is more sensitive to a particular student's strengths, weaknesses and preferred style of learning, emulating the quality of a private human tutor. Computer-based tutors separate the subject matter they teach from the format of instruction. Their instructional actions are based on an internal model of the student and a set of teaching procedures-- strategies and tactics-- to select from on the basis of the model. Several programs have been developed, spanning areas such as computer programming, medical diagnosis and geography. While most systems are still experimental, some have been formally evaluated and compare very favorably to class-room instruction (e.g, the work of J. R. Anderson and his co-workers on the LISP tutor (1985a)).

In parallel, there have been several attempts to develop computer programs that could deliver psychotherapy. Most of recent research has focused on automating the presentation of psychotherapeutic techniques rather than on the process of therapeutic dialogue. For instance, Lang et al (1970) successfully used a computer to carry out systematic desensitization and to monitor progress in a group of female snake phobic undergraduates. A computerized "dilemma counseling" system, PLATO DCS, has been developed (Wagman, 1980) and been shown to be effective with university students. A portable calculator-size computer system designed to provide immediate feedback concerning caloric intake was found to promote weight loss in obese female volunteer subjects. Gosh et al. (1984) and Selmi (1983) have presented via computer relatively standard self help interventions for phobias and depression (cf. Servan-Schreiber (1986) for a more elaborate review). Although such computerized interventions appear promising, in terms of outcome, most also appear to require substantial additional therapist contact to promote compliance with treatment. This requirement parallels the results of many psychotherapeutic "bibliotherapy" studies carried out with a variety of psychological problems. None of these programs has attempted to base the format and content of therapeutic interventions on an internal model of the psychological situation of patients. While the absence of an internal model does not necessarily preclude treatment effectiveness, we believe that an intelligent and individualized dialogue is necessary to increase acceptability, motivation and compliance.

How can psychotherapy be cast in an intelligent tutoring system? In this paper, Ipropose to review some canonical examples of intelligent tutoring systems from which principles can be derived to guide application to a new domain area such as psychotherapy. I then present a prototype of a system designed to lead patients suffering from sexual dysfunctions through a therapeutic dialogue.

I. Lessons from Tutoring Systems

A. Historical Examples

1. SOPHIE

SOPHIE (Brown et al., 1982) is one of the earliest intelligent tutors. Its task is to monitor a student attempting to debug a simulated electronic circuit. The student can ask questions about circuit components, perform measures at different locations (voltage, intensity, etc...) and make hypotheses about the malfunction. SOPHIE contains a module that can identify the fault in the circuit based on functional specifications of electronic circuits and its own problem solving strategy. When the student performs measurements or proposes a hypothesis, SOPHIE can evaluate the student's strategy and critique it according to its own solution. The important characteristics of SOPHIE for our purposes is that it is attempting to teach *problem solving* in a *well defined, isolated domain*, and that it can rely on a *functional model* of the domain area.

2. GUIDON

GUIDON (Clancey, 1982) teaches a medical student attempting to solve a case of bacteremia or meningitis. The student is presented with some symptoms of a patient and can gather further data or make a diagnosis. He can also ask for help or for the relevance of particular information. To assess the student's knowledge, GUIDON uses as an "ideal student" model a specially designed version of the MYCIN expert system which can solve the case. Rules of the expert system are marked as known by the student or not, according to the student's questions and hypotheses; this results in an "overlay model".

Several "tutoring rules" use this model and the context of the lesson to decide on the appropriateness and content of instructional interventions. The expert system is not based on a functional model of the domain (the human body) but rather on judgments about empirical information such as the fact that a particular symptom is associated with a particular type of infection. It is the clear specification of an expert's knowledge and problem solving strategy that serves to assess the student's behavior. GUIDON's emphasis is on teaching *domain knowledge* as well as some general (not domain specific) strategies and the domain can be considered to be *isolated* in that students do not come to the task with a burden of misconceptions acquired from previous experience.

3. BUGGY, LMS and WHY

Rather than modeling the student's knowledge as a subset of an expert's knowledge --the premise of overlay models-- BUGGY (Brown & Burton, 1978), LMS (Sleeman, 1982) and WHY (Stevens et al., 1982) assume that students misrepresent the domain knowledge. They use false principles and procedures as well as incorrect facts. Stevens, Collins and Goldin, using protocols of human tutoring sessions recognized that tutoring activities often revolved around students' bugs. They claimed that:

"Much of a teacher's skill depends on knowledge about the types of conceptual bugs students are likely to have, the manifestations of these bugs and the methods for correcting them." (Stevens et al., 1982)

Accordingly, WHY was designed to teach students about causes of rainfall by systematically probing students' knowledge for misconceptions and missing reasoning steps. WHY engages the student in a Socratic dialogue guided by heuristics such as:

If the student gives as an explanation of causal dependence one or more factors that are not necessary
Then select a counter-example with the wrong value of the factor and ask the student why his causal dependence does not hold in that case

These heuristics rely on a functional model of the domain organized in a script-like fashion that lets the system evaluate questions and answers provided by the student. Thus, WHY does not need to have a "bug library" of students' misconceptions since bugs can be detected and (partially) understood by comparing the students' answers to the model of the domain (in fact, WHY does not even have an *explicit* representation of bugs). However, as the authors of WHY have stressed, this procedure identifies only extra and missing sub-steps in the scripts. Misconceptions which do not fall into these categories are not recognized by the system. This is the price that must be paid to avoid an extensive library of common bugs.

A key lesson that can be derived from WHY is that tutoring dialogues can be driven by bug identification and correction rather than by comparing the student to an ideal model. Experiments with WHY also made explicit the role of local and global strategies or goals to control the tutoring dialogue, that local management of the interaction based on recognition of a misconception is not sufficient and "discourse knowledge" is necessary to provide the system with a global perspective on the dialogue. To summarize, WHY attempts to teach primarily *domain knowledge*, in a domain that students approach with *a priori misconceptions*, and where a *functional model* of the domain can be used to evaluate students' questions and hypotheses.

4. Andersonian Tutors

John R. Anderson and his co-workers have developed several tutors that attempt to teach a new cognitive skill such as LISP programming or proving geometry theorems (Anderson et al., 1985a; Anderson et al., 1985b). The student is presented with a problem to solve and proposes a solution step by step (e.g, types a LISP expression on the keyboard). The tutor monitors every step and compares it to a production system that solves the problem in parallel with the student. If the student is found to be using an adequate production the tutor says nothing; if the production that matches the student's behavior is a "buggy production", a pre-stored intervention is generated to get the student back on the right track. This "model tracing" requires very precise models of the particular problem solving activity -- in the form of production rules-- and a considerable library of buggy rules in order to follow the student step by step. This level of precision is attainable in domains that are quite isolated from previous knowledge and where the bugs that occur most commonly stem from the student's experience with previous concepts of the domain itself (e.g, the confusion between "append" and "list" in LISP). As a result of this fine grained analysis, concerns for dialogue are minimal: since the tutor always knows where the student is and what her knowledge-state is, there is no need to carry on a dialogue that would yield a "cognitive diagnosis" of the student like WHY does. Also, it is interesting to note that these tutors do not rely on functional models of the domains even though such models are readily available (e.g, running LISP expressions). Rather, like GUIDON, they use a model of expert problem-solving. Thus, Andersonian tutors teach a *problem-solving skill*, in *isolated domains*, and do not use a functional model of the domain but rather a *functional model* of the expert problem-solving process.

B. Lessons for applications in other domains

The first obvious conclusion that one can derive from studying existing tutoring systems is that a very large effort is spent clarifying and formalizing the domain knowledge and problem solving strategies that ought to be taught. Whether these are represented as a production system like in GUIDON or as scripts and semantic nets like in WHY, more knowledge of the domain has to be implemented than what is traditionally considered to be adequate for expert systems.

A second and more puzzling conclusion is that tutors cannot rely on highly general, domain independent principles to teach their subject matter. It is not enough to recognize overgeneralizations, overdifferentiations and teach how to form and test hypotheses and collect enough information. In particular, as soon as prior misconceptions play an important role in the student's approach to the domain, what is required is a direct recognition of misconceptions and application of specific corrections. Bugs are not domain independent.

Finally, three dimensions of the task domain seem to have particular implications in terms of the style of interaction, modeling of the student and representation of domain knowledge that a tutor can use:

1. Teaching problem-solving vs. domain knowledge

As we have seen, systems such as SOPHIE and the LISP tutor focus primarily on teaching the student how to use tools or knowledge to complete a task. Conversely, GUIDON and WHY emphasize the acquisition of new knowledge. This dimension determines where the tutor should stand on a continuum from *monitoring*-- or "coaching"-- of the student involved in a problem solving task, to engaging the student in a *dialogue* about a case-- be it a patient or the presence of rain in Oregon. By extension, this also determines how much "discourse knowledge" a tutor should possess. The further the student is from active problem solving, the more the tutor should know about the structure of teaching dialogues.

2. Isolated domain vs prior misconceptions

GUIDON and SOPHIE are designed for task domains that are sufficiently different from students' previous experience for them to carry relatively few preconceptions to the new domain.

On the other hand, when analysing the knowledge states of students who claimed to know nothing about the causes of rainfall, Stevens and Collins found that each student harbored a host of beliefs and misconceptions. It is not that these students did not know about processes of evaporation, condensation etc..., but that they "knew" it incorrectly.

In fact, this distinction is more blurred than it might seem at first. Brown and Burton have stressed that students develop mental models of electronic circuits that can be incomplete or wrong and that SOPHIE is limited in its instructional capacity by its inability to deal with this phenomenon directly. What this dimension influences is whether models of students should be attempted in terms of a *sub-set of an expert* or "ideal student" model, or rather as a *collection of incorrect facts and erroneous procedures* that should be diagnosed and corrected. The more pre-conceptions might "infect" the domain area to be taught (e.g, causes of rainfall) the more active debugging of the student's approach is necessary. On the other hand, when misconceptions are less likely to influence the acquisition of new knowledge (e.g, facts about meningitis), the tutor can rely more heavily on an ideal student model.

3. Functional models

All the systems we have reviewed encompass a functional model of either the domain knowledge --electronic circuits for SOPHIE, scripts of rainfall for WHY-- or a functional model of the expert problem solving process --production systems of GUIDON and Andersonian tutors. Thus these tutors can evaluate the student's behavior -- and by extension her knowledge-- by comparison with a functional model. In particular, they can identify bugs in the student's conceptions by referring to a functioning or "debugged" model. The existence of such models avoids the difficult problem of having to compile extensive "bug libraries" associated with correction procedures. However, as we have seen, these models have their limits and Andersonian tutors are augmented to include an extensive library of typical errors.

II. Applications to Psychotherapy

A. Psychotherapy as Intelligent Tutoring

Not all forms of psychotherapy are equally amenable to the intelligent tutoring approach. However, one in particular, cognitive psychotherapy, insists on the logical scrutiny of cognitions and uses "error libraries" of common cognitive distortions which characterize particular forms of psychopathology. Interestingly, cognitive therapists stress their role as *teachers* and actively instruct their patients to recognize and overcome their maladaptive misconceptions. In addition, cognitive therapy sessions have a well defined and consistent format.

The purpose of cognitive psychotherapy is to work with a patient suffering from a circumscribed problem such as depression or marital difficulty by going over the patient's view of the domain in which the problem is rooted, looking for lack of information, misconceptions and maladaptive thoughts. In this sense the task of the therapist leans more toward reviewing domain knowledge rather than coaching a specific problem solving skill. A therapeutic tutoring system would thus more naturally fit in a dialogue framework than in a monitoring paradigm. In addition, the knowledge addressed by such as system is overwhelmingly not "isolated". Patients have typically lived with their problems for a significant amount of time before they come for consultation and they have developed their own model of the domain, most often out of partial information and misleading experiences. Their model is thus bug-ridden and patients need to *un*learn as much as they need to learn. As a result, a natural representation of the patient is to match his beliefs against an "error library" rather than attempting to develop an overlay model of an "ideal patient". Finally, and unfortunately, there are no extensive functional models of the type of social interactions in the context of which patients' problems arise, neither do functional models exist for human reasoning in the domains in which these problems occur.

This description of the task of psychotherapy, which reflects the theoretical commitment of cognitive psychotherapy, led to direct conclusions about the kind of tutoring system that we could plan to build: 1. the system had to be able to lead patients through a therapeutic dialogue and would thus require elaborate "discourse knowledge"; 2. the dialogue had to be driven by identification of misconceptions; 3. the absence of functional models would force the development of extensive error libraries associated with typical remedies to constitute the knowledge base of the tutor.

B. Sexpert: An Intelligent Tutor for Sexual Dysfunctions

Bearing this analysis in mind, we attempted the development of a tutoring system for the domain of sexual dysfunctions. We chose the domain of sexual dysfunctions for several reasons. First, the common cognitive distortions about sexual functioning have been well described. Second, sexual problems are a relatively well defined area of psychological difficulty for which well worked out therapeutic interventions exist. For some dysfunctions such as premature ejaculation or primary anorgasmia the appropriate interventions are also highly successful. Third, there is a strong tradition of self help among individuals suffering from sexual problems that is well accepted and even encouraged by sex therapists. We hoped that this would facilitate the acceptance of a new therapeutic modality. Finally, the relative anonymity and non-judgemental approach that could be offered by an intelligent computer-therapist has been shown to facilitate disclosure of personally sensitive information such as sexual problems.

Sexpert is organized around two components: a diagnostic module and a dialogue driver. We will discuss them in turn.

1. The Diagnostic Module

In Sexpert, the elementary step on which all instruction is based consists of the identification of a problem or misconception from a set of questions asked of the partners. We have compiled a large number of misconceptions from the literature and from our domain experts. Typical examples are: Misinformation: for example that anesthetic creams are useful to help delay ejaculation (in a sense they are, but they partially anesthetize the female partner too); False expectations: for example, males thinking that all women are turned on if their breasts are fondled, or females thinking that a partial loss of erection during intercourse indicates loss of interest.

Once such an error library is available to the program, the process of identifying bugs becomes one of traditional "classification problem solving" as defined by Clancey (1984). The program goes from data (answers to questions) to data-abstractions (internal representations) and performs a heuristic match from data-abstractions to bug categories. Finally it refines the bug category to a particular misconception or faulty procedure. Once a bug is identified, it is added to the model of the couple and can be used to diagnose bugs of a higher level of complexity which encompass several simple bugs. For example, after having determined that the couple suffers from a particular kind of premature ejaculation, the program might find out that the couple has reduced the duration of their foreplay. Sexpert may interpret this latter fact as an attempt on the part of the couple to keep the male arousal low prior to penetration (which in most cases does not work and results in both short foreplay *and* short intercourse).

This approach to diagnosis can readily be implemented with the methodology of expert systems and we are using a traditional rule-based inference mechanism to diagnose simple bugs of the kind illustrated. However, certain misconceptions only emerge when a therapist integrates a large number of related simple problems. Our approach to this problem has been to break down the analysis of the interaction pattern into core components that leave most of the details out and results in a gross, first-path representation. We then determined the most frequently occurring patterns at the detailed level and implemented them as possible add-ons to the coarse representation. Using this technique, if the couple falls into one of the common patterns, the analysis generated by Sexpert includes most of the details they have mentioned. If not, the program is still able to rely on a "general idea" of their situation.

2. The Dialogue Driver

We have seen how Sexpert relies on identification of problems, faulty procedures and bugs as its primary teaching step. However, teaching is a structured process and does not consist of sequences of unrelated actions. To quote Stellan Ohlsson's insightful analysis of tutoring systems:

> "A tutoring effort is structured; it coordinates the individual teaching actions, subsumes them under a plan for how to teach the relevant knowledge. The moment-to-moment behavior of the tutor originates in the execution of that plan, rather than in successive decisions about what to do next. If the student model is to be useful, it has to contribute in some way to the construction and execution of instructional plans." (Ohlsson, 1986)

Once we had convinced ourselves that it was possible to drive the psychotherapeutic process around identification and correction of bugs, we had to organize the interaction with patients in a meaningful way. To capture the gist of this type of therapeutic dialogue, we created a hierarchy of dialogue plans in which each level successively refines the actions of the system. Only abstract specifications of the topic to be discussed are implemented at the top level, while an intermediate level specifies the issues to be raised and their order and the lowest level determines the exact order and content of questions or explanations to be presented. This idea of hierarchical dialog plans is inspired by the concept of hierarchical planning developed by Sacerdoti (1974) and the structure of the MENO-TUTOR of Woolf and McDonald (1984).

For example, the main dialogue plan of the first session consists of the following goals: gather background data, get presenting complaints, identify and formulate problems, investigate contributing factors, relate contributing factors to symptoms, formulate and propose a treatment program. Within each of these categories, local dialogue plans are generated to structure and focus the discussion on the relevant topics. For instance, in order to investigate factors contributing to primary anorgasmia, Sexpert dynamically creates a plan to discuss physical health, sexual history, sexual fears and anxieties, sexual attitudes, etc... Within each of these categories, a more specific dialogue plan is again generated (e.g, selecting and ordering issues of sexual history) and so on until particular information is elicited (see figure 1.).

- Complete introduction (background data)
- Obtain presenting complaint
- Identify and formulate problem(s)
- Investigate contributing factors
  - Physical health
  - Sexual history
    - Familial influence
    - Sex education
    - Previous experiences
    - Sexual trauma
      - Nature of trauma
      - Age at that time
      - Partner's reaction
  - Sexual fears and anxieties
  - Sexual attitudes
  - Environmental factors
  - Relationship factors
  - Repertoire factors
- Relate contributing factors to symptoms
- Prepare and propose treatment plan

Figure 1.

This figure illustrates the hierarchical structure of dialog plans generated dynamically during the session. At first, the top level plan that organizes the entire session is generated. The session then proceeds by generating and refining plans under each heading in due time. For illustration, we have expanded only one such heading: the investigation of contributing factors. Note that in order to generate this plan, the program needs to have identified a dysfunction. This example assumes that primary anorgasmia has been identified. A plan is generated to discuss several categories of factors that may have contributed to cause or to maintain this particular dysfunction. Again, we have expanded only one of the categories asserted by this plan: sexual history. Another, more precise, plan is created that specifies issues of sexual history to be addressed. Finally, for each of these issues, a terminal level plan specifies questions to be asked directly of the users. When an issue or categoy has been fully investigated, the program backs up to the next higher level and proceeds with the next topic on the plan at that level. For example, after sexual trauma has been investigated, the program addresses sexual fears and anxieties.

While this hierarchical structure allows the program to group questions in relevant contexts, it also leaves a lot of freedom to reorganize plans to fit particular situations at run-time. More importantly, perhaps, is that each level of dialogue plan is independent of the specifics of lower levels (the lowest level specifics being particular questions and explanations). This provides the system with the ability to reason at a conceptual level on the dialogue itself. The system knows what concepts have been discussed or remain to be discussed independently of particular questions or answers. For example, it can tell whether a problem has been identified and formulated independently of what problem it is or of how many problems have been found. This ability becomes particularly useful when several dysfunctions are present simultaneously and the system has to alternate or mix the discussion of each of them throughout the session.

Example of dialogue on recent changes in a case of premature ejaculation:
The following two questions take place in the context of a diagnostic plan for premature ejaculation. They help determine whether the dysfunction is primary (the problem was always there) or secondary (there has been a period of normal functioning). They also discriminate between two forms of secondary premature ejaculation (better with previous partners or with current partner in the past).

John , were you able to exert better control over your ejaculation earlier in your relationship with Mary?
-->no
Did you have more control over your ejaculation with previous partners ?
-->yes
I see
(...)

Later, when the active goal is to inquire about contributing factors of premature ejaculation, a plan is generated to inquire about changes in the relationship which may help explain that the dysfunction is secondary and to give a summary of the relevant findings after all questions have been asked (rather than an explanation after each question). This triggers an even more specific plan which specifies the order and nature of questions to be asked and the following dialogue takes place:

Is the frequency of intercourse with Mary markedly lower than what you were used to with previous partners ?
-->no
Are you using intercourse positions with Mary you did not use with previous partners when you had more control ?
-->no
Are you generally more tired when you try to make love now than when you had better control ?
-->no
John , do you feel more anxious when you are having sex with Mary than with previous partners ?
-->yes
Was your non-sexual relationship better when you had better control ?
-->yes
(...)

While the implementation of this dialogue hierarchy gives a logical and adaptable structure to the interaction, the possibility remains that the system goes down a wrong path and that some backtracking is necessary. For example, at some point in the dialogue, information provided by one of the patients may be inconsistent with previous answers or with prior conclusions derived by the program. If the inconsistency is recognized, the patients are asked specific questions to clarify the situation and all previous answers and conclusions are reconsidered in light of the modifications. As a result the dialogue may take a completely different orientation with new plans and questions being generated while all other, still valid, previous answers remain available. Unfortunately, only the most predictable inconsistencies can be recognized by Sexpert. We have found that in most cases it is better to rely on the patients themselves to determine when the program is heading in the wrong direction. Thus, at all times, they have the option to change any of their previous answers that they feel were responsible for the current misled focus of the dialogue.

Finally, it is also important for the program to be able to follow through when a sensitive issue has been raised which requires immediate attention at the expense of the more general line of inquiry of the dialogue. For example, at different occasions during the dialogue, it might be discovered that the woman is pregnant (e.g, when discussing contraception). In that case, it is important to react immediately even though the rest of the information might not be relevant to what Sexpert wants to know at that point. To implement this "noticing" mechanism within the general hierarchical goal structure of the dialogue, we use demons that immediately trigger a dialogue plan which takes precedence over the current focus, any time their activation conditions are met. When the execution of the plan is completed, control returns to the last active goal.

3. Current Status of Sexpert

A functioning prototype of Sexpert consisting of twelve hundred rules and approximately one hundred and seventy pages of text currently operates on a personal computer. This prototype includes: an introductory section which explains the uses and limits of Sexpert and gathers background information concerning the users; a diagnostic section which makes decisions and gives individualized feedback concerning primary and secondary premature ejaculation, primary anorgasmia, and a variety of other orgasmic concerns; a contributing factors section which evaluates and discusses over fifty possible factors which may contribute to the above problems including an evaluation of sexual repertoire and the couple relationship; and a fifteen session treatment section for premature ejaculation. Prelimary results, based on the reaction of fifteen unscreened volunteer couples, are encouraging. All the couples thought that the dialogue was logical, appropriate and intelligent. Several spontaneously remarked that Sexpert was "smart" and appeared to really understand. None complained about the length of the session (60-90 minutes) or the amount of text to be read. Interestingly , almost all couples were highly sensitive to the wording of the texts, noticing and reacting strongly to differences such as: "your difficulty with duration of intercourse" rather than "your concern over duration of intercourse". We are currently systematically studying subjects' evaluation of the program and the degree of attitude change and belief revision related to the first session with Sexpert.

III. Conclusion

At this stage of the evaluation phase, it is difficult to draw definitive conclusions about the value of our approach. Some of the most salient limitations include the restriction of the interface to yes/no and multiple-choice modes, the pronounced domain-specific nature of the tutoring strategies and the absence of models for "deeper misconceptions (e.g, where do bugs come from?). Prior experiments with computer-based psychotherapy seem to suggest that a clinically significant therapeutic effect can take place in spite of these limitations. If this proved to be the case, the methodoly of intelligent tutoring systems understood and applied according to the framework we have explored promises to make psychotherapy of some well-defined emotional problems more accessible and affordable.

Acknowledgments

This project would not have been possible without the dedication and insight of Professor Irving Binik who provided virtually all the domain knowledge used by Sexpert, and of Simon Freiwald who designed the programming environment and inference engine on top of which Sexpert was built. I also wish to thank Benoit Mulsant for many stimulating discussions and his helpful review of this paper, and Ted Shortliffe for his comments and constructive criticisms of an earlier draft.

References

[Anderson et al., 1985a] Anderson JR, Boyle CF, Reiser BJ. Intelligent Tutoring Systems. *Science.* 228:456-462

[Anderson et al, 1985b] Anderson JR, Boyle CF, Yost G. The Geometry Tutor. In *Proceedings of the International Joint Conference on Artificial Intelligence. (Los Angeles, California)*

[Brown & Burton, 1978] Brown JS, Burton RR. Diagnostic Models for Procedural Bugs in Basic Mathematical Skills. *Cognitive Science* 2:155-192

[Brown et al., 1982] Brown JS, Burton RR and DeKleer J. Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II and III. In Sleeman D and Brown JS (Eds) *Intelligent Tutoring Systems* Academic Press: New York

[Clancey, 1982] Clancey WJ. Tutoring Rules for Guiding a Case Method Dialogue. In Sleeman D and Brown JS (Eds) *Intelligent Tutoring Systems* Academic Press: New York

[Clancey, 1984] Clancey WJ. Classification Problem Solving. In *Proceedings of the National Conference on Artificial Intelligence.* (Austin, Texas) pp. 49-55

[Gosh et al., 1984] Gosh A, Marks IM, Carr AC. Controlled Study of Self-Exposure Treatment for Phobics: Preliminary Communication. *J R Soc Med* 77:483-487

[Lang et al., 1970] Lang PJ, Melamed BG, Hart J, A. Psychophysiological Analysis of Fear Modification Using an Automated Desensitization Procedure. *Journal of Abnormal Psychology*. 76:220-234

[Ohlsson, 1986] Ohlsson S. Some Principles of Intelligent Tutoring. *Instructional Science* 14:293-326

[Sacerdoti, 1974] Sacerdoti ED. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence* 5:115-135

[Selmi, 1983] Selmi P. Computer-Assisted Cognitive-Behavior Therapy in the Treatment of Depression. Ph.D Thesis, Univ. of Wisconsin, Madison

[Servan-Schreiber, 1986] Servan-Schreiber D. Artificial Intelligence in Psychiatry. *Journal of Nervous and Mental Disease* 174:191-202

[Sleeman, 1982] Sleeman D. Assessing Aspects of Competence in Basic Algebra. In Sleeman D and Brown JS (eds) *Intelligent Tutoring Systems* Academic Press: New York

[Stevens et al., 1982] Stevens A, Collins A, Goldin SE. Misconceptions in Students' Understanding. In Sleeman D and Brown JS (Eds) *Intelligent Tutoring Systems* Academic Press: New York

[Wagman, 1980] Wagman M. PLATO DCS: An Interactive Computer System for Personal Counseling. *Journal of Counseling Psychology*.27,16-30.

[Woolf & McDonald, 1984] Woolf B, McDonald DD. Context-dependent transition in tutoring discourse. *Proceedings of the National Conference on Artificial Intelligence* (Austin, Texas) pp. 355-361