NON-DIAGNOSTIC INTELLIGENT TUTORING SYSTEMS

Leo Gugerty and Kimberly Hicks
Mei Technology Corporation
San Antonio, Texas

ABSTRACT

The keystones of traditional intelligent tutoring systems (ITSs) have been complex procedures for student diagnosis and adaptive instruction based on diagnostic data. While some of these systems have been shown to be effective, they are also very expensive to develop. This paper describes another class of ITSs, non-diagnostic ITSs, which do little or no student diagnosis, and concentrate their intelligence in other areas. Intelligent features of non-diagnostic ITSs include: modeling of experts' reasoning processes and cognitive representations (often using graphic displays), comparison of student and expert performance, and replays and summaries of student performance. While traditional, diagnostic ITSs are usually intended to be used in a stand-alone fashion, non-diagnostic tutors are designed to facilitate collaborative learning among students and between teachers and students.

The non-diagnostic approach to ITS development offers either a low-cost alternative to traditional ITSs or a way to expand the educational capabilities of traditional systems. This paper presents a framework for comparing the features of non-diagnostic and diagnostic tutors. A number of non-diagnostic and diagnostic ITSs are described, and data on the costs and educational effectiveness of each type of ITS is presented. Finally, obstacles to wider use of non-diagnostic ITSs are discussed.

ABOUT THE AUTHORS

Dr. Leo Gugerty is a cognitive psychologist in the Training Technology Division of Mei Technology Corporation. His research focuses on complex problem-solving skills (such as troubleshooting) and effective interventions to train these skills. He has a Ph.D. in Cognitive/Experimental Psychology from the University of Michigan, Ann Arbor, and has held research positions at Educational Testing Service, Lockheed Engineering and Sciences Co., and the University of Houston.

Kimberly Hicks is an Industrial Psychology Specialist in the Training Technology Division of Mei Technology Corporation. Her work is concentrated in the areas of training systems analysis, and survey development and analysis. She has an M.A. in Industrial Psychology from St. Mary's University, San Antonio, TX.

NON-DIAGNOSTIC INTELLIGENT TUTORING SYSTEMS

Leo Gugerty and Kimberly Hicks
Mei Technology Corporation
San Antonio, Texas

INTRODUCTION

There is a sense in which the goals of traditional intelligent tutoring systems (ITSs) are both too ambitious and too narrow. Most "classical" (or traditional) ITSs, such as the LISP Tutor (Anderson & Reiser, 1985), are designed to provide tutoring in a stand-alone setting (i.e., without a human teacher present). This ambitious goal requires that the ITS handle all aspects of the very difficult task of tutoring, including expert problem solving, student diagnosis, tailoring instruction to changing student needs, and providing an instructional environment (e.g., a simulation). On the other hand, the goal of developing very intelligent stand-alone ITSs is narrow in the sense that it limits our conception of how intelligence can be incorporated into computer-based training and education. One key problem with focusing on stand-alone ITSs is that we may overlook intelligent computer-based systems that include the teacher as part of the tutorial interaction.

ITSs are currently being developed that break with the pattern of traditional ITSs. An example is the Intelligent Conduct of Fire Trainer (INCOFT), an ITS to train the skill of identifying aircraft from radar displays (Newman, Grignetti, Gross & Massey, 1989). INCOFT does little student modeling and relies on a teacher to provide much of the instruction. Its intelligence lies in its ability to advise students when their performance differs from an expert's, to model experts' reasoning for the student, and to provide useful summaries and replays of student performance that can be discussed by the student and the teacher. Thus, INCOFT acts as an intelligent teacher's aid, and facilitates collaborative learning.

The goal of this paper is to describe ITSs like INCOFT, and compare these to traditional ITSs like the LISP Tutor. The key features that differentiate INCOFT and the LISP Tutor are student diagnosis and adaptive instruction. The LISP Tutor performs student (or cognitive) diagnosis; that is, it makes inferences about the knowledge and misconceptions underlying student performance. Having a detailed student diagnosis enables the LISP Tutor to adapt its instruction to small changes in student knowledge during a tutoring session. INCOFT, on the other hand, simply records student performance without making inferences about it. Therefore, INCOFT must rely on the teacher to adapt instruction to fine-grained changes in student knowledge. Whether or not an ITS performs student diagnosis has a large effect on how it can be used in instruction. Therefore, we will refer to systems like the LISP Tutor as *diagnostic* ITSs. This term is intended as shorthand for a system that performs both detailed diagnosis and adaptive instruction based on the diagnosis. Systems like INCOFT will be referred to as *non-diagnostic* ITSs (and sometimes as intelligent teacher's aids).

There are a number of reasons for exploring non-diagnostic ITSs. The first reason concerns the difficulty of the tutoring task, as described above. Recently, some ITS researchers have suggested that some tutoring tasks, such as student diagnosis, will require long-term basic research before solutions are found (Burger & DeSoi, 1992). A second reason is that augmenting a teacher's knowledge with a non-diagnostic intelligent teacher's aid may provide just as much and perhaps more educational benefit as replacing a teacher (or at least part of a teacher's task) with a stand-alone, diagnostic ITS. A third reason is that traditional diagnostic ITSs are very expensive to develop and are applicable only in narrow domains. Non-diagnostic tutors can cut the cost of tutor development by eliminating the need for some of the complex components of traditional ITSs.

In this paper, we will describe specific features, advantages, and disadvantages of both non-diagnostic and diagnostic ITSs, and estimate the cost of each approach in terms of type and level of development work. This overview should allow someone considering developing or using an ITS to understand the costs and benefits of each approach.

COMPARISON OF DIAGNOSTIC AND NON-DIAGNOSTIC ITSs

Table 1 contains a list of some of the key capabilities or features of ITSs. The table is organized in terms of the four components of traditional ITSs, the expert, diagnostic, instructional, and interface modules. For each capability in the table, a range of options is presented, from "high-tech" options that rely on the computer to perform the pedagogical function (e.g., diagnosis), to "low-tech" options that rely on the teacher (or other students) to perform the function. The table shows the capabilities of two diagnostic ITSs, the LISP Tutor and Sherlock I (Lajoie & Lesgold, 1989), and two non-diagnostic ITSs, INCOFT and the Maintenance Aid Computer Hawk Intelligent Institutional Instructor (MACH III) (Kurland, Granville & MacLaughlin, 1992). The LISP Tutor has primarily high-tech features. Sherlock I is less sophisticated than the LISP Tutor, but still possesses the essential capabilities of a diagnostic ITS. The two non-diagnostic ITSs have primarily low-tech features, except in the case of their interfaces, which use high-tech features such as realistic simulation and modeling expert reasoning and representations.

A few points should be made about Table 1. First, the terms high-tech and low-tech are not meant to connote a value judgment. Complex technology is not always the best technology. Second, there is not a strict correspondence between diagnostic ITSs and high-tech features, on the one hand, and non-diagnostic ITSs and low-tech features, on the other. The two non-diagnostic ITSs use some high-tech features, as the table shows. Also, diagnostic ITSs can incorporate low-tech features that facilitate teacher involvement, as do non-diagnostic systems. An example of this is the Sherlock II maintenance skills tutor, which provides precise student diagnosis and adaptive instruction as well as feedback (such as replays and summaries of student performance) intended to foster collaborative learning (Katz & Lesgold, in press). Sherlock II can be considered a hybrid of a diagnostic and a non-diagnostic ITS.

In the following, we first describe each of the ITS capabilities in the table and explain the differences between low-tech and high-tech options. Then, some of the diagnostic and non-diagnostic ITSs in Table 1 are compared in terms of the table features.

Comparison of ITS Features

Expert Module - As Table 1 notes, an important question concerning the expert module is whether it simulates human thought processes. *Black-box* expert modules solve problems using methods completely unlike humans, while *glass-box* experts attempt to simulate the important human thought processes used in the task being instructed (Burton & Brown, 1982). The LISP tutor is an example of a glass-box expert module. This tutor uses hundreds of production (if-then) rules to represent the knowledge and strategies used in LISP programming in a detailed manner. INCOFT uses a black-box expert (mathematical equations) to solve aircraft identification problems.

The main advantage of a glass-box model is that its detailed model of human thought processes allows it to more specifically and accurately diagnose student knowledge and misconceptions, and then base instruction (e.g., explanations) on specific student weaknesses. The main disadvantage of glass-box models is their cost. The expert module for the LISP tutor is based on Anderson's ACT* theory of human learning and problem-solving, which is based on years of research and theoretical work (Anderson, 1983).

A second important question characterizing expert modules is whether they generate the steps to solving problems online, when presented with a brief problem description, or have the specific solution steps pre-stored in their memory. A system that generates problem solutions online usually can solve a wider variety of problems than a system that relies on "canned" (pre-stored)

Table 1. Capabilities of Diagnostic and Non-Diagnostic Intelligent Tutoring Systems

Key: Diagnostic Systems: LISP Tutor (L), Sherlock I (S); Non-Diagnostic Systems: INCOFT (I), MACH III (M)

<u>ITS Module</u> Capabilities	Low-Tech					High-Tech	
<u>Expert Module</u>							
Simulates human thought processes?	no expert module			black-box expert	I,M,S	glass-box expert	L
Generates problem solutions online?	no expert module			canned solutions for pre-selected problems, or generated solutions based on a simple algorithm	I,M,S	solutions generated for a wide variety of domain problems	L
<u>Diagnostic Module</u>							
Info in student model	no student model	performance history	I,M	specific knowledge & skills based on a black-box expert (e.g., issue-based tutoring)	S	specific knowledge, skills, and misconceptions based on a glass-box expert (e.g., model tracing)	L
Use of student model	no *student* model	performance *replays and* summaries	I,M			to choose problem/topic sequence, *knowledge/skills to coach, and level* of coaching	L,S
<u>Instructional Module</u>							
How is the content and sequence of topics/problems determined?	*by teacher*		I,M	*preset by tutor developers*		*generated online*	L,S
How is the method of instruction determined?	by teacher		I,M	preset by tutor developers	L,S	generated online	
How is the content and timing of instructional interventions determined?	by teacher		I,M	preset by tutor developers	L,S	generated online	L,S
Group or individual use?	collaborative		I,M	stand alone	L,S	both	
<u>Interface</u>							
Simulates real-world task context?	not at all			to some extent	M,S	closely	I,L
Models expert reasoning and representations?	very little			somewhat	L,S	extensively	I,M

problem solutions. The LISP Tutor can generate problem solutions online, while Sherlock I and MACH III use pre-stored problem solutions. INCOFT generates solutions online using a very simple algorithm.

To use pre-stored problem solutions, one must conduct a task analysis, which is time consuming and requires some specialized knowledge. However, this is a less difficult task than developing a model of problem solving that can generate solutions to arbitrary problems in a domain. Thus, the use of task analysis and pre-stored problem solutions is less expensive and more widely applicable than developing a system that can generate solutions online.

Diagnostic Module - The second major component of an ITS, the diagnostic module, allows the system to create student models that record aspects of individual students' performance and knowledge. The ITS then uses information in the student model to tailor its instruction to the needs of individual students.

The most advanced diagnostic modules use performance data concerning the actions students take during problem solving and/or the final results of their problem solving to make inferences about the knowledge and skills behind individual students' performance. A powerful method for making these inferences, called *model tracing,* can be used if an ITS has a glass-box expert module that models human thinking. Model tracing is used in the LISP tutor. As the student uses the computer to plan and write computer programs, the diagnostic module matches each problem solving action taken by the student with the specific knowledge (i.e., production rules) that the expert module would use to produce that action. The diagnostic module also contains production rules to represent specific student misconceptions, so that when students make errors, it can match them with the underlying misconception. The diagnostic module can then record in the student model production rules that a student knows well, rules the student knows less well, and misconceptions.

A slightly less sophisticated method of student diagnosis is *issue-based tutoring* (Burton & Brown, 1982). An issue-based tutor makes inferences about the knowledge underlying student performance, like a model-tracing tutor. However, issue-based diagnosis can be accomplished with a black-box expert module, whereas model-tracing requires a glass-box expert. Sherlock I uses a complex version of issue-based tutoring.

The information in a student model created by model tracing or issue-based diagnosis can be used by the instructional module in a number of ways, such as in determining the contents of hints and explanations and in selecting problems for students. For example, if the diagnostic module infers that a student mistake is based on knowledge the student knows fairly well, the instruction module can give only a general hint to the student. On the other hand, if the student mistake is based on knowledge the student knows poorly, the instruction module can give a detailed explanation of the mistake and the correct move.

The least sophisticated diagnostic modules record only data about student performance in a student model, making no inferences about the knowledge underlying this performance. An example is INCOFT, which monitors and records the aircraft-identification actions students take while they observe radar displays. It also records the timing of students' actions. These data are used by the instructional module in two ways: to provide replays of a problem in which differences between the student's and an expert's performance are pointed out; and to create summaries of student and expert performance on a problem. INCOFT does not use its limited student model to adjust the instruction based on a student's performance or to select problems. These tasks are left up to the teacher.

In the extreme case, some computer-based training systems record no student performance data, and do no student diagnosis.

Since the student model created by model tracing relies on a glass-box expert that closely simulates human thought procedures, this kind of diagnostic capability is expensive and time consuming to develop. The minimal student model used by INCOFT is obviously much easier to develop. The effort required to develop issue-based student models varies widely depending on the complexity of the issues and the complexity of the schemes by which issues are updated.

Instructional Module - The third major component of an ITS is the instructional module. Planning and delivering tutorial instruction is a complex, interactive task. The decisions a tutor must make include: 1) curricular decisions regarding the content and sequencing of topics or problems, and 2) instructional decisions regarding the type of instructional intervention, the content and timing of instructional interventions, and the overall method of instruction. The tutor must choose from a variety of instructional interventions, such as exposition (e.g., explanations, examples of concepts, modeling of procedures), coaching (e.g., hints and explanations during problem solving), and asking and answering questions. Methods of instruction also vary widely, including direct instruction, guided discovery learning, and Socratic dialog. In addition, the advantage, and challenge, of tutorial interaction is that all of these decisions can be changed frequently based on the tutor's assessment of the student's progress, motivation, and learning style.

As Table 1 shows, an ITS can make these curricular and instructional decisions online (during a tutorial interaction) using a comparison of the student's and the expert's knowledge states. Alternatively, an ITS's developers could make some or all of these decisions on a one-time basis and hardwire these decisions into the ITS's algorithm. Finally the ITS could leave curricular/instructional decisions up to the teacher.

The first curricular/instructional decision shown in Table 1 focuses on curricular decisions, such as problem sequencing. The LISP Tutor and Sherlock I choose problems online based on diagnosis of individual students' knowledge. At the other extreme, INCOFT requires the teacher to make these decisions. The next instructional decision shown in the table concerns the overall methods of instruction. Almost all existing diagnostic ITSs have a single method of instruction that is used consistently. For example, the LISP tutor uses a directive, problem-based method of instruction, with immediate feedback after errors. Non-diagnostic ITSs like INCOFT and MACH III rely on the teacher to determine the method of instruction.

In terms of more specific decisions about the content and timing of instructional interventions, most diagnostic ITSs make some decisions online, while other decisions are preset by the developers, as is shown in the table. For example, in the LISP Tutor, the content of specific hints and explanations was preset by the developers. However, the tutor makes a number of instructional decisions online, such as when to intervene (based on student errors), whether to provide a general hint or a detailed explanation (based on the number of student errors or the student's request), and the topic of the hint or explanation (based on the diagnostic module's assessment of the missing knowledge or misconception underlying the student's error). Again, with non-diagnostic ITSs, the teacher must decide what instructional interventions to use, for example, how to use the replays and summaries of student performance. Finally, Table 1 also characterizes the instructional modules of ITSs according to whether they focus on collaborative or stand-alone use.

To summarize this subsection, even though the LISP tutor is one of the most intelligent of ITSs, the instructional decisions that it generates online are based on fairly simple algorithms. This is typical of other diagnostic ITSs. Much of the intelligence of the LISP tutor, and most diagnostic ITSs, lies in the diagnostic and expert modules. For most ITSs, many important curricular/ instructional decisions, such as the overall instructional method and the content of explanations, are made on a one time basis by the system developer and cannot be changed by the tutor itself during operation or by the teacher. Developing ITSs that can flexibly make difficult curricular/instructional decisions is a long term research goal. The approach taken by non-diagnostic ITSs is to leave these difficult decisions up to the teacher.

Human-Computer Interface - The final ITS component in Table 1 is the interface. Two factors that distinguish low-tech and high-tech interfaces are whether the tutor simulates the real-world task context, and whether the interface allows students to use experts' reasoning and knowledge representations while using the tutor.

Non-diagnostic tutors concentrate their intelligence in the interface.

A realistic simulation environment can help students transfer knowledge from the tutorial to a job situation. INCOFT uses a realistic, simulated radar display that allows students to solve aircraft identification problems in real time. The artificial intelligence and psychological expertise required to build a realistic simulation often is less extensive than that needed to create glass-box expert and diagnostic modules. On the other hand, expertise in computer graphics and video is needed, and a thorough task analysis must be done. The LISP Tutor also uses a realistic interface.

Bonar (1991) has suggested that the effectiveness of an ITS will be greatly improved if the interface allows students to see and work with experts' reasoning and representations while solving problems. For example, MACH III represents expert troubleshooting knowledge in terms of "troubleshooting trees", which show all the general and specific faults that could cause a particular symptom in a radar system, as well as the troubleshooting tests to conduct for each specific fault. Students can view the appropriate tree on the computer in order to understand why a particular test was recommended. As in constructing a simulation environment, extensive artificial-intelligence knowledge is not required to build an interface like this. Rather, one needs a careful analysis of experts' problem solving processes for the task to be tutored.

Comparison of Particular Diagnostic and Non-Diagnostic ITSs

To summarize the discussion of the capabilities of diagnostic and non-diagnostic ITSs, we will compare the capabilities and the effectiveness of the LISP Tutor with those of INCOFT and MACH III. As the table shows, the LISP Tutor is a diagnostic ITS. It uses high-tech approaches in its expert and diagnostic module, that is, a glass-box expert and model tracing. Its instructional module uses a mixture of high-tech techniques (e.g., choosing the topics and level of detail of hints and explanations online) and some less sophisticated ones (using only a single, preset method of instruction). The LISP Tutor's interface simulates real-world programming interfaces closely, and sometimes allows students to use expert task representations (e.g., by showing students templates of LISP functions to fill in).

INCOFT takes a non-diagnostic approach. Although its expert module generates problem solutions online, it does this using a simple algorithm. INCOFT's diagnostic module records only performance data about the nature and timing of student responses. The tutor's instructional output consists of replays and summaries of students' performance, and demonstrations of expert performance. These were designed to be used more as informational aids for teachers and students than as stand-alone instructional interventions. INCOFT leaves the decision about how to use these aids, and most other curricular/instructional decisions, up to the teacher. The strength of INCOFT lies in allowing students to practice a real-time task on a realistic interface, and then, via replays and summaries, providing students with comparisons of their performance and that of experts. The important instruction with INCOFT occurs when the teacher and student (or groups of students) discuss the student's replayed problems. While using the replays and summaries, students do not have the pressure of real-time performance, and can evaluate and discuss their performance.

So far in this paper, we have examined the differing capabilities of diagnostic and non-diagnostic ITSs, and given a rough indication of the cost or level of effort these capabilities require to develop. Diagnostic tutors are more sophisticated in how they model students' knowledge states and adapt instruction to students' needs. Non-diagnostic tutors focus their intelligence on modeling experts' task knowledge and providing replays and summaries that can facilitate collaborative instruction and learning. Because of the difficulty of developing student diagnosis schemes, diagnostic tutors are usually more costly to develop. A key question for someone who is contemplating developing an ITS is whether the added sophistication of diagnostic ITSs is worth the cost. To begin to answer this, we will present some data on the effectiveness of diagnostic and non-diagnostic tutors.

The example we have used for a diagnostic tutor has been the LISP tutor. The model-tracing approach used in this tutor has also been used in tutors for geometry, algebra, and calculus (Merrill, Reiser, Ranney, & Trafton, 1992). Anderson and Reiser (1985) found that students using the LISP tutor took 15.0 hours to complete a set of programming exercises, much faster than students who completed the exercises on their own (26.5 hours), and almost as fast as human-tutored students (11.4 hours). Each group performed equally well on posttests of their programming knowledge. Other model-tracing ITSs, such as Anderson's Geometry Tutor and the Graphical-Instruction-In-LISP (GIL) Tutor, have also been found to be more effective than traditional instruction (Merrill et al., 1992).

Two points should be made about these findings. First, in all of these evaluation studies, students using ITSs also received classroom instruction from a teacher. Thus these studies suggest that model-tracing tutors are effective in outside-the-classroom situations. The studies do not suggest that these tutors can replace human teachers altogether.

The second point concerning the effectiveness of model-tracing tutors is that some of this effectiveness may be due to other aspects of the tutors besides model-tracing, such as the structured editor in the LISP tutor and the graphic interfaces in the Geometry Tutor and GIL. However, studies have shown that when both model-tracing diagnosis and its associated instructional guidance are removed from the LISP tutor and GIL, students learn slower and sometimes perform worse than with the full versions of these tutors (Corbett & Anderson, 1991; Merrill et al., 1992). Another study compared a version of GIL that provided very little instructional feedback (that is, where model tracing diagnoses were used only to point out when students made errors) to versions that gave more detailed explanations of the locations of and reasons for errors (Merrill et al., 1992). The versions with more detailed instructional feedback resulted in faster and better student learning. These studies suggest that each of the key aspects of intelligence in a model-tracing ITS -- model tracing diagnoses and adaptive instructional feedback based on these diagnoses -- can lead to increments in students' learning.

On the other hand, the intelligent capabilities of model-tracing tutors do not always lead to better learning. For example, students using the version of GIL without model-tracing diagnoses or instructional feedback performed better on a debugging posttest than students using the full-fledged GIL (Merrill et al., 1992). Thus, the detailed and immediate feedback characteristic of model-tracing tutors may deprive students of the opportunity to make, and learn to correct, errors. This issue deserves further investigation, since debugging is an important aspect of programming skill.

Few studies have been conducted on the effectiveness of non-diagnostic tutors, as these tutors have been developed more recently than diagnostic tutors. INCOFT was not evaluated formally. However, a controlled study of the effectiveness of MACH III has been completed (Acchione-Noel, Saia, Williams & Sarli, 1990). MACH III was developed by the same company as INCOFT, and shares its focus on classroom-based learning via replays, summaries, and graphic representations of expert knowledge (e.g., troubleshooting trees). In keeping with MACH III's intended use as a classroom teaching aid, the study compared the use of this ITS with the traditional classroom methods of practicing troubleshooting in a radar maintenance class. The traditional methods involved using procedure manuals and schematics (paper-based practice). Both the MACH III and the "paper-based" group also received lectures and practice on the actual radar equipment.

Although the MACH III students did not perform any better than the paper-based group on practical and written troubleshooting posttests, the tutor students did perform more consistently (i.e., with lower variability). Also, the MACH III group solved significantly more, and more difficult, troubleshooting problems during the class than the paper-based group.

The lack of significant differences in student posttest performance in this initial study should not be taken as a general criticism of non-diagnostic tutors, for a number of reasons. First, the instructors felt they needed more training on how to use MACH III in the classroom. Second, the instructors tended not to use some of the more advanced features of MACH III, such as the troubleshooting trees, because they thought these gave students too much help. The Army school where MACH III was tested (Ft. Bliss) has continued to use the tutor in classes following the tests (Kurland et al., 1992).

Because MACH III was used in a different instructional context, the evaluation of this ITS cannot be compared easily to the evaluations of diagnostic tutors. The MACH III evaluation studied tutor use in the classroom and used a control group that received extensive classroom instruction. The studies of diagnostic ITSs (e.g., Anderson & Reiser, 1985; Lajoie & Lesgold, 1989) looked at stand-alone ITS use, and found that students using these ITSs performed better than students who received no additional instruction.

CONCLUSION

Non-diagnostic ITSs offer a potentially fruitful approach to computer-based education and training that complements the approach taken by traditional diagnostic ITSs. The lack of student diagnosis in non-diagnostic tutors will likely result in lower tutor development costs. In addition, the non-diagnostic approach promises to have positive educational value. Non-diagnostic features such as modeling experts' representations, replaying and summarizing students' performance, and focusing on collaborative learning implement some of the key aspects of the successful cognitive-apprenticeship approach to training and education (Collins, Brown & Newman, 1989).

The non-diagnostic approach can be applied in the development of new ITSs and in converting existing systems (e.g., expert systems) to ITSs. Many expert systems use black-box expert modules. Black-box modules are difficult to convert to a traditional ITS because the expert problem-solving knowledge in the expert system does not mimic human knowledge and thus is not in a form that can be easily used for student diagnosis (Anderson, 1988). However, if one is developing a non-diagnostic ITS that does little or no student modeling, then the conversion task will be much easier. For example, the Air Force's Armstrong Laboratory is developing a computer-based training system based on the Integrated Maintenance Information System (IMIS) (Link, Von Holle & Mason, 1987). IMIS is a job aid that will assist flightline maintenance technicians repair aircraft. IMIS contains an expert system that gives troubleshooting advice to technicians. Since this expert system is closer to a black-box than a glass-box system, non-diagnostic ITSs are being used as models in developing the training version of IMIS. Preliminary design work suggests that non-diagnostic features can be added to IMIS to create a relatively low-cost training system, because extensive task analysis will not be necessary beyond that performed for developing the expert system.

Before non-diagnostic ITSs can become widely used, however, a number of obstacles must be overcome. The first obstacle concerns how to integrate these ITSs into the classroom and train teachers to use them. The importance of considering these issues was highlighted by the instructors in the MACH III study, who asked for more ITS training and tailored the ITS use to their instructional goals in ways not intended by the developers.

The second obstacle to widespread use of non-diagnostic ITSs is the lack of empirical validation of their effectiveness. Conducting rigorous research that tests these systems in their intended educational settings (i.e., classrooms and other collaborative learning situations) should be a high priority.

Acknowledgments: - We would like to thank Valerie Shute, Patricia Hsieh, Ron Sheffer, William Walsh, Kevin Singley, and Elizabeth Gibson for comments on the paper.

REFERENCES

Acchione-Noel, S., Saia, F.E., Williams, L. & Sarli, G. (1990). Maintenance aid computer hawk intelligent institutional instructor: Training developments study. (Report No. TRAC-WSMR-TEA-90-051). White Sands Missile Range, NM: U.S. Army TRADOC Analysis Command.

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1988). The expert module. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 21-54). Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Reiser, B. J. (April, 1985). The LISP tutor. *Byte*, pp. 159-175.

Bonar, J. G. (1991). Interface architectures for intelligent tutoring systems. In H. Burns, J. W. Parlett, & C. L. Redfield (Eds.), *Intelligent* *tutoring systems: Evolutions in design.* Hillsdale, NJ: Erlbaum.

Burger, M. L., & DeSoi, J. F. (1992). The cognitive apprenticeship analogue: A strategy for using ITS technology for the delivery of instruction and as a research tool for the study of teaching and learning. *International Journal of Man-Machine Studies, 36*, 775-795.

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.* New York, NY: Academic Press.

Collins, A., Brown, J. S. & Newman, S. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning and instruction.* Hillsdale, NJ: Erlbaum.

Corbett, A. & Anderson, J. (1991). Feedback control and learning to program with the CMU Lisp Tutor. Paper presented at the 72nd Annual Meeting of the American Educational Reseach Association, Chicago, IL.

Kurland, L., Granville, R. & MacLaughlin, D. (1992). Design, development, and implementation of an intelligent tutoring system for training radar mechanics to troubleshoot. In M. Farr & J. Psotka (Eds.), *Intelligent instruction by computer.* Philadelphia, PA: Taylor & Francis.

Lajoie, S. P & Lesgold, A. (1989). Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship. *Machine-Mediated Learning, 3*, 7-28.

Link, W. R., Von Holle, J. C., & Mason, D. (1987). Integrated Maintenance Information System (IMIS): A maintenance information delivery concept. (Report No. AFHRL-TP-87-27). Dayton, Ohio: Wright-Patterson AFB.

Merrill, D., Reiser, B., Ranney, M. & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences.*

Newman, D., Grignetti, M., Gross, M., & Massey, L. D. (1989). Intelligent Conduct of Fire Trainer: Intelligent technology applied to simulator-based training. *Machine-Mediated Learning, 3*, 29-39.