ASTUTE: AN ARCHITECTURE FOR INTELLIGENT TUTOR DEVELOPMENT

Brandt Dargue, Don Busch, Bruce Perrin
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

ABSTRACT

Intelligent Tutoring Systems (ITSs) have proven to be very effective at transferring complex knowledge to a student. They have also proven to be difficult to create and maintain. To create Intelligent Tutors in a cost-effective and maintainable manner, a set of tools and processes called Adaptive Student Tutoring Environment (ASTUTE) is under development. ASTUTE is a composition of commercial off-the-shelf (COTS) applications and specialized software that will enable instructional designers or subject matter experts to create computer-based training that will transfer the proficiency of experts to a learner in a way that is very stimulating to the learner. The resulting tutors, in effect, emulate on-the-job training (OJT) with an expert who is dedicated to being a private mentor for each learner.

This paper presents the ASTUTE architecture and describes how tutors using that architecture simulate mentored OJT and train apprentice students to perform at expert levels. Included are discussions on ASTUTE's methods of: (a) molding instruction to students learning habits, capabilities, and skills; (b) exercising skills in an environment that best reveals the intended use of the knowledge; (c) providing external coaching support that either fades as the student's skill builds or is under control of the student, giving advice only when asked; and (d) providing reflective follow-up that allows comparison of a student's solution with an expert's solution.

ABOUT THE AUTHORS

Brandt W. Dargue is a software engineer at McDonnell Douglas Training Systems, a Division of McDonnell Douglas Aerospace. He is a co-designer of ASTUTE and has been designing PC-based multimedia applications since 1985. He has also worked on projects that have designed and created simulators, neural networks, expert systems, configuration management, computer-managed instruction, and authoring tools. Mr. Dargue received a B.S. in Aerospace Engineering from Parks College of St. Louis University.

Donald C. Busch is a software engineer in the Automated Courseware Development research group at McDonnell Douglas Training Systems. He has worked on projects involving intelligent scheduling and automating portions of the software development life cycle. His current work focuses on developing tools to create Intelligent Tutoring Systems (ITSs) as one of the co-designers of ASTUTE. Mr. Busch has a M.S. in Computer Science from Michigan State University.

Bruce M. Perrin currently leads a multi-disciplinary team investigating intelligent tutoring methods at McDonnell Douglas Training Systems. He is also the principal investigator on Contributive Research and Development for the Training Decisions Systems for the Air Force, and is responsible for the design and conduct of training and decision-support studies and evaluations. His research interests and publications include work in the areas of cognitive analyses of performance and training environments, decision-making processes underlying adaptive training methods, and decision-making and -aiding under uncertainty. Dr. Perrin has a PhD in Industrial Psychology from Kansas State University.

ASTUTE: AN ARCHITECTURE FOR INTELLIGENT TUTOR DEVELOPMENT

Brandt Dargue, Don Busch, Bruce Perrin
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

INTRODUCTION

Intelligent Tutoring Systems (ITSs) are gaining recognition as a viable supplement to other forms of training (e.g., Kaplan & Rock; Psotka, Massey & Mutter, 1988), although considerable work remains to achieve all of the capabilities described by early proponents (Carbonell, 1970). General guidelines for ITS construction are difficult to identify because most have been hand-crafted using a variety of methods and architectures (Dillenbourg & Self, 1992).

This paper briefly discusses the evolution of ITSs, while describing some of the key features that appear to underlie successful systems. It then presents a particular architecture that implements those key features. The architecture is described by its functional goals. After each goal, the module(s) responsible for implementing that goal will be introduced. After all the goals have been described, the architecture is assembled and described as a whole.

THE EVOLUTION OF ITSs

As early as the mid-1960s, some intelligence was exhibited in computer-based drill and practice programs through their ability to select arithmetic problems appropriate to a student's prior performance (Suppes, 1967; Woods & Hartley, 1971). These "adaptive systems" achieved considerable success and saw extensive use over a number of years. Their adaptive capabilities were based on simple, parametric estimates of student mastery that were sufficient for the limited application domains in which they operated.

With the application of computer-based training (CBT) to more complex domains, more detailed representations of student skill and knowledge were required for the training to adapt to student needs. Researchers and developers in the emerging field of ITSs responded with a variety of student cognitive modeling approaches, designed to track understanding in domains where straightforward indications of mastery were unavailable. Building on advances in expert systems technology (e.g., Hart, 1986; Hayes-Roth, Waterman, & Lenat, 1983; Pearl, 1988; Waterman, 1986) and research on the nature of domain specific problem-solving skill and expertise (e.g., Gott, 1989), they responded with multidimensional models that tracked the development of mental models that paralleled those of the expert (e.g., Anderson & Reiser, 1985; White & Fredericksen, 1986).

In addition to providing the template for student mastery, a model of expert problem-solving in the domain also filled the dual roles of providing a means to demonstrate skilled task performance and interpreting and evaluating student actions. With the emphasis in ITSs on not only performing like an expert, but also thinking like one, considerable effort has been placed on deriving applicable techniques for obtaining this knowledge, including cognitive task analysis (Means & Gott, 1988). By linking expert behavior to the application of underlying decision-making processes in the expert model, ITSs are endowed with the basis for inferring student cognitive states from observable behavior (Van Lehn, 1988).

To complete the training environment, ITS researchers and developers have harnessed advances in simulation, modeling, and graphic visualization technologies to continue and expand the role of adaptive training systems for skill practice. Complex systems that could not be cost-effectively taken out of service for training could be simulated. Problems that have significant impacts, yet occur infrequently and therefore may not be encountered during on-the-job training (OJT), could be presented to students. Strategies for problem-solving that were largely ignored in formal training previously (Greeno, 1978) could become the focus of skill development programs using ITSs due to the skill practice environment they provided (e.g., Lesgold et al., 1988).

Taken together, the components of an ITS provide a rich environment for implementing a variety of instructional interventions and teaching tactics (Ohlsson, 1987). Demonstration, guided and annotated practice, hints, coaching, immediate feedback on errors, and other tactics become available through the capabilities of an ITS. While the evaluation of features and capabilities has been unfocused (Dillenbourg & Self, 1992), and much of the research has been on issues of the nature of expertise and the representation of knowledge (Frye, Littman & Solloway, 1988), ITSs have, nonetheless, achieved considerable success (Psotka, Massey & Mutter, 1988). The following sections are devoted to the identification and description of an architecture that embodies the central features and capabilities of an ITS as indicated in this section.

ASTUTE'S APPROACH TO ITS DELIVERY

ASTUTE is an acronym for:

Adaptive	What is taught and how it is taught changes to meet the student's needs.
Student	Student performance is used to alter the course of training.
Tutoring	Tutoring methods supplement training.
Environment	A suite of tools -- some commercial off-the-shelf (COTS) -- that work together to foster learning.

ASTUTE is an architecture being developed to help ease the development of ITSs. It is a modular architecture, comprised of COTS applications, along with custom tools and runtime modules.

The architecture of ASTUTE is comprised of:

- Student Model - a dynamic evaluation of the student;
- Plan of Instruction - a roadmap through the available lessons, examples, and exercises;
- Lesson Manager - a component which uses the Student Model to select activities from the Plan of Instruction;
- System Simulation - software that emulates significant portions of the problem-solving domain; and
- Expert System - a model of the subject matter expert's actions and reasoning.

The rest of the paper will be presented in the order of the acronym ASTUTE. First, a discussion of the adaptiveness of the instruction and how the student is important to the architecture. Next, a discussion of how the benefits of tutoring are emulated, and finally the interworkings of the complete environment are described.

ADAPTIVE, INDIVIDUALIZED TRAINING

Each student has unique capabilities -- one student might be better at understanding concepts, while another may be better at performing procedures. Some students may need to be reminded of basic skills or knowledge; other students may find these same reminders redundant and unnecessary. While one student may need to see several examples or have more practice at each step, another may not need to practice until after several concepts are presented. Since students learn at different rates and with different styles, the instruction, its pace, and its presentation should be tailored to the student to enhance the learning environment.

There are several ways to tailor instruction. For example, CBT generally is structured so that each lesson is appropriate for a student who has the expected background and has mastered previously presented information. Adaptive training systems take the next step in tailoring instruction to meet the student's needs by selecting each subsequent exercise based on an overall measure of student mastery. By using more detailed information on both student capabilities and the knowledge covered by exercises, skill-building exercises can be dynamically scheduled by an ITS to meet the student's requirements. Instruction can proceed toward mastery of the skills required in the domain when proficiency is shown, detouring to remediate weaknesses or correct misunderstandings when needed. Beyond modification of the sequence of topics, the general pace of instruction and level of practice can also be adjusted to match the student's proficiency in particular areas.

Various other pedagogical methods and teaching tactics may also have value, depending upon characteristics of the domain and the students. Therefore, we are making a variety of these methods available within ASTUTE. Some of these methods are established at the time of design of a specific ASTUTE tutor, such as options for exploration or "freeplay" simulation and determining the types of hints to be provided. Also, at design time, a plan for the development of skills, consistent with (a) the general abilities of students and (b) the difference between these skills and those of the expert's, is constructed. Other methods, such as the dynamic management of instructional topics and providing the proper type and level of hint, are inherent in the architecture. Still others represent optional courses of study that may be selected by the student. While the methods ASTUTE uses are more fully described in the sections that follow, some of the general forms of this adaptability are:

- The lessons chosen for presentation are selected to efficiently bring the student to expert levels of knowledge from his or her current state.
- The exercises are selected to gradually refine the student's skills. Often, the best problem is slightly beyond the student's current abilities.
- When the Instructional Designer thinks it is appropriate for freeplay and exploration, students can create their own problems and ask ASTUTE for advice or guidance toward the solution.
- If an instructional topic is not mastered, an alternate presentation of the topic is selected.
- Students can ask to repeat parts of any exercise.
- Students can ask for practical examples for any knowledge covered.

Emphasis is on helping the student become an expert -- not on progressing the student through a fixed set of materials (although this goal is accomplished as well). Nominally, there are one or more paths through instructional topics to transition students from their expected abilities to the point where their ways of viewing the state of a problem, the strategies they apply to select a course of action, and the actions they take in solving problems approximate those of an expert. However, instruction for individual students may need to vary from this path to address their unique capabilities or requirements. The primary mechanisms ASTUTE uses to provide this adaptability are the Plan of Instruction, the Student Model, and the Lesson Manager.

Plan of Instruction

In any CBT, instructional topics will be sequenced in a manner expected to facilitate skill development. Typically, the sequence of lessons is specified in advance, including points where a student may select one or more options or choose the order for one or more required lessons. The Plan of Instruction (POI) for ASTUTE differs mainly in the level of detail on instructional activities and the model of proficiency on which it is based. ASTUTE also uses this Plan of Instruction in a more proactive manner. The sequence is not fixed, nor is it based upon student selection; rather, ASTUTE selects appropriate topics and activities.

ASTUTE's Plan of Instruction is used to schedule topics that relate to specific aspects of expert performance. In the maintenance troubleshooting domain, for example, an expert's problem-solving skill appears to reside in the application of three types of knowledge: systems, strategic, and procedural (Gott, 1989). A Plan of Instruction should encompass all three types of knowledge explicitly in order to build these knowledge structures. Related research (e.g., White & Frederiksen, 1986) further suggests that acquisition of knowledge of these types may be facilitated by initially focusing on general, overview models (also known as qualitative or systems models) before instruction on detailed, procedural, strategic, or quantitative skills should be undertaken.

The capability to schedule activities designed to develop specific aspects of expert performance is integral to ASTUTE. A generic example is shown in Figure 1, although an actual Plan of Instruction would have to track specific skills and knowledge at a much greater level of detail.

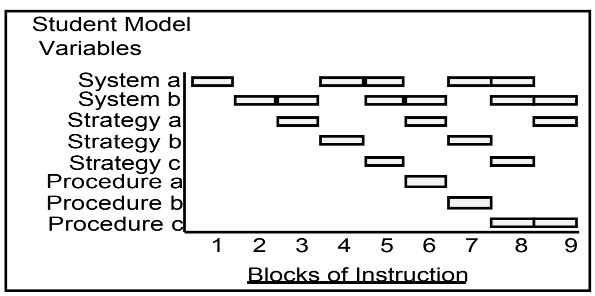

Figure 1. Generic Plan of Instruction

Student Model

The Student Model is a dynamic picture of a given student's developing expertise. Student models are generally used in ITSs for the following purposes:

1) determining when to advance the student through the lessons;
2) choosing or generating exercises for the student;
3) determining when to offer unsolicited advice to the student; and
4) adapting explanations to use concepts that the student understands (VanLehn, 1988).

In ASTUTE, the Student Model participates in each of these functions, but has sole responsibility for none of them. Rather, each function involves an interaction of the Plan of Instruction, Student Model, and Lesson Manager. In advancing the student through lessons and choosing exercises (rather than the next lesson or exercise in the sequence), ASTUTE's Student Model provides a current snapshot of the student's expertise, which is compared to an expected level of mastery based on a planned development of skills. In turn, advancement through the sequence of topics largely determines the types of explanations, advice, and other types of support ASTUTE provides. These types of adaptability of ASTUTE are described further in sections below.

ASTUTE uses a dynamic, hierarchical Student Model in which the lowest level nodes are relatively specific concepts and procedures, representing relatively concrete instructional objectives found in the Plan of Instruction. These nodes are closely associated with the actions or responses students make, and updates to them are propagated up the hierarchy to represent the student's understanding of more abstract knowledge or skills in the domain. This approach is similar to one implemented in an ITS called Sherlock (Lesgold et al., 1988), although methods to update the nodes (described below) are different. An example of a Student Model is shown in Figure 2. An actual Student Model would have one or more levels below that displayed.

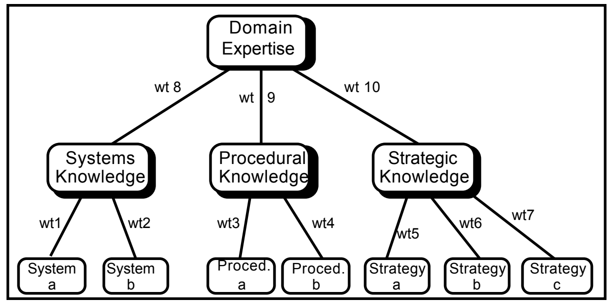

Figure 2. Generic Student Model

Each student's progress through ASTUTE instruction is tracked by a unique Student Model. As a student proceeds through instruction and exercises, his actions are interpreted and evaluated by testing procedures for textual information, or by an Expert System for performance skills. These evaluations, in turn, are used to update the lowest level node of that individual's Student Model. The higher level nodes are formed by a weighted combination of the nodes directly below them, and serve, at a courseware author's discretion, several purposes. For example, it may be preferable to display progress to a student in terms of his/her standing on nodes higher in the model since the lower level nodes are too detailed and show too much momentary variation to provide useful student feedback.

Lesson Manager

Since the Plan of Instruction is directly correlated with the Student Model, if the student shows any weakness, ASTUTE's Lesson Manager is able to determine what topics of instruction are to be used to strengthen that weakness. A pool of lessons, examples, and exercises is used as a database of learning materials. Each is linked to a specific topic in the Plan of Instruction. The materials in the pool of instruction may be exercises using simulation, existing or new CBT lessons or tests, problem-solving exercises, or even off-line assignments. Some materials may be remedial and not presented unless a specific need for the material is demonstrated. Also, the Lesson Manager is able to select from several different versions of presenting each main topic.

Finely tuning the exercises to the student may substantially enhance the learning experience. The student may be challenged to increase his limits and would not be discouraged by overwhelming problems. Students may also need reminders of some of the concepts that were understood earlier. This can be accomplished in ASTUTE by selecting exercises and skipping over steps in which the student has repeatedly shown expertise. The simulation can be brought to a specific step in the exercise so the student could complete the problem from that point. To present an example to the student, ASTUTE's built-in expert can demonstrate a particular method to complete a problem with the reasoning behind each action clearly noted to the student.

TUTORING

Apprenticeship Training

Learning is fostered when skills and knowledge are applied in a realistic problem-solving context (Glaser, 1990). Apprenticeship in many domains is successful for several reasons. Lesgold, Eggan, Katz, and Rao (1992) list four: (a) real tasks are more motivating than simply reading about them or doing practice exercises; (b) when knowledge is anchored in experience, it gains real meaning for the trainee because it is rooted in his or her personal experience; (c) the coach becomes a supportive external memory that shares attentional load; and (d) perhaps most importantly, apprenticeship affords opportunities for the master or coach to provide knowledge in the context of its use, so that the trainee learns both what to do and when to do it.

Hallmarks of apprenticeship training methods include:

- *situated learning*, where students learn by practice in an environment that reveals the intended uses of their acquired knowledge;
- *external support* or *scaffolding* from the tutor (or master) in the form of ideal modeling of the performance, hints, reminders, explanations;
- *fading* of external support as the apprentice's skill builds; and
- carefully *sequenced* learning activities that are both sensitive to changing student needs and promote integration and generalization of knowledge and skill (Collins, Brown, & Newman, 1987).

"In short, learning is assumed to occur through guided experience in instructional environments that provide progressive, explanation-based, and otherwise generally supported practice in the mechanics of solving problems" (Gott, 1989, p. 99).

Traditional apprenticeship training highlights methods for apprentices to learn through observation, individualized coaching, and practice (Collins, Brown, & Newman, 1987; Lave, 1977). Perhaps the most potent form of highlighting occurs as apprentices observe coaches make their knowledge structures and decision-making processes visible and explicit -- and thus knowable (Gott, 1989). These benefits have been the hallmark of apprenticeship training (Gott, 1989), but due to specialization and other organizational constraints, opportunities for apprenticeship are limited. An ITS provides many of these lost advantages, duplicating, as closely as possible, the trainee undergoing OJT in the task environment, while benefiting from the full attention of a task expert who is also an expert trainer (Loftin & Savely, 1992).

The ASTUTE architecture accomplishes simulated apprenticeship training by providing an environment in which skills can be practiced with expert feedback and assistance. The ASTUTE system functions as a non-threatening private tutor to the student. The system also enables the student to practice new skills in the work environment or in an environment that is similar to the actual work environment, which is the essence of today's definition of tutorial. This allows the student to correlate new knowledge with the real world and his or her experience. The components of ASTUTE that achieve the simulated apprenticeship are the Lesson Manager (described above) to provide fading of external support and sequencing of activities, the System Simulation to provide the situated learning, and the Expert System to provide external support.

System Simulation

The System Simulation of ASTUTE performs two main functions: first, it provides an environment to practice skills, and second, it provides a means of measuring skills during the student's demonstration of his understanding of the domain -- the system being simulated. The student can interact with the simulated equipment just as he would interact with real equipment on the job. He gains most of the benefits of using real equipment without incurring the inevitable costs. Based on the actions he chooses to perform on the simulation, ASTUTE can update the Student Model.

This practice, in itself, is valuable for training purposes; however, a student can only learn so much by exploring and manipulating equipment on his own. It is even more valuable when an expert is available to assist the student. Using a human expert has become prohibitively expensive. To fulfill this need for a mentor, ASTUTE, like most ITSs, provides an expert system.

Expert System

The Expert System of ASTUTE fills the role of mentor or coach. The Expert System evaluates the actions that the student takes in the System Simulation to solve a troubleshooting problem, it gives him feedback on the quality of his actions, and it can suggest an action to take next and why. The Expert System allows the student to learn and explore on his own, offers guidance if requested, and can interject if it is necessary. The objective is to offer instructional support to the student without explicitly telling him what to do. The student should be free to learn and discover, without being completely on his own.

The Expert System provides three types of support for the student. These include support for summarizing the status of a solution, advice on the next step in problem solving, and critiques of actions taken.

The Expert System can summarize the state of the problem solution in many different ways. The exact nature of these summaries varies with the domain, but a few specific examples for maintenance troubleshooting illustrate the alternatives. Perhaps the simplest summary is a description of the components whose functionality has been verified by direct test. This summary helps the student learn how particular procedures can be used to evaluate the status of individual components. A more complex summary of the problem is a depiction of the additional components eliminated by inference. Although not directly tested, components that are upstream of a verified component can usually be eliminated by inference. Summaries of this type lead the student toward general problem-solving strategies. A summary of the components that are still suspect can also be displayed, helping students identify the portion of the problem space which remains to be tested. Summarizing the current state of the problem may help the student identify the next step, but additional support may be required.

In one type of advice, the Expert System directly identifies the correct action for the student. This may be appropriate in the early stages of skill acquisition, but in general, it is prudent to let the student exercise as much initiative and independence as possible. This helps to solidify application of the knowledge and foster confidence. Thus, an additional type of support implemented in ASTUTE is a sequence of hints that help to identify the next step. When the student first asks for a hint, the result is a hint that recommends a strategy that might be used and the general area where it might be effective. Hints become increasingly specific with subsequent requests, until the specific action is identified.

A final type of support provided by the Expert System is a critique of troubleshooting actions as they are taken. Each action is evaluated as to whether it (a) matches one of the actions that would be recommended by the Expert System at that stage of the problem solution; (b) could lead to a solution, but is not one of the next actions that the Expert System would recommend; (c) is redundant with previous tests or inferences from them; or (d) is irrelevant to the problem at hand. Other considerations, such as cost, time, danger of injury to the individual, or damage to the equipment, may also be used as part of this critique.

Additionally, these types of support can be combined to achieve specific kinds of feedback. For example, if in early instruction the goal is to demonstrate the effect of particular problem-solving strategies, identification of the next step and the portion of the problem space resolved might be displayed together. Somewhat later, it might be advisable to help the student keep track of the status of the solution; consequently, the state of the problem solution might be displayed at all times, along with a general hint on a problem-solving strategy that might be used. This progression of support is linked to the Plan of Instruction, so that as the student moves through the sequence of topics, the appropriate level of support is provided. In addition to the support programmed in the Plan of Instruction, all implemented types of support are always available to the student on demand, at the discretion of the instructional designer.

Reflective Follow-up

The objective of ASTUTE is not only to teach a student specific problem-solving procedures, but also to teach him how the equipment actually works and strategies for troubleshooting that equipment. Solving problems with expert guidance is a critical part of the instruction necessary to acquire these skills, but students need additional opportunities to reflect on their solutions and to understand what they have done well, and what they need to learn.

ASTUTE gives the student the opportunity to reflectively review his solution -- to replay it step-by-step. The student not only self-reflects on actions taken, but also sees an evaluation of these actions, and compares his solution to the expert's. The student may request hints at each step to gain a better understanding of the effects of each action. He may even want to diverge from the first attempt in order to try to solve the problem in a different way. The goal is to let the student gain a better understanding of what his troubleshooting techniques accomplished and why, and where his troubleshooting techniques are different from those recommended by an expert.

The student can also replay the expert's solution one step at a time. Again, he can request hints, either as reasoning behind each step or as the status of the solution, to better understand the effects of the expert's troubleshooting choices. The major goal of stepping through the expert's solution is to understand the rationale behind the expert's troubleshooting technique. By stepping through his own solution and stepping through the expert's solution, the student should have a better idea of how his understanding of the system and his strategies differ from the expert's, and will be able to close the knowledge gap between them.

ENVIRONMENT - PUTTING THE PIECES TOGETHER

The methods and mechanisms described above are incorporated into the environment of the ASTUTE architecture. The architecture incorporates COTS applications, along with custom tools and runtime modules.

The ASTUTE Architecture

The architecture of ASTUTE is shown in Figure 3. It is comprised of:

- Student Model - a dynamic evaluation of the student
- Plan of Instruction - a roadmap through the available lessons, examples, and exercises
- Lesson Manager - a component which uses the Student Model to select lessons from the Plan of Instruction
- System Simulation - software that emulates significant portions of the problem-solving domain
- Expert System - a model of the subject matter expert's actions and reasoning used as a basis for comparison to the student's actions and to provide hints.

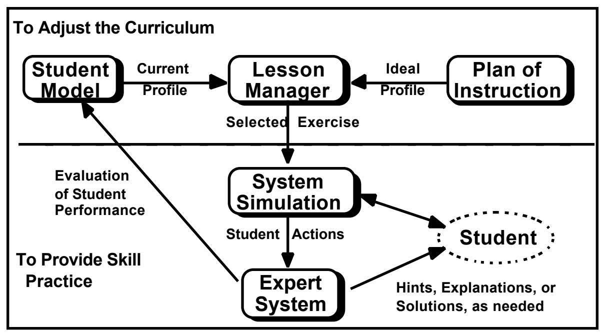

Figure 3. ASTUTE Architecture

Figure 3 illustrates the communication between the modules of ASTUTE and identifies the modules that are involved in the two main capabilities of ASTUTE. Above the horizontal line are the modules most involved with adjusting the curriculum, the adaptive feature of ASTUTE. At the curriculum level, ASTUTE has the capability to respond to student difficulties and misunderstandings as they occur by adjusting the sequence of topics. Below the line in Figure 3, the lesson level of ASTUTE is shown, indicating the modules most involved with providing skill practice in a supportive environment.

The components of ASTUTE communicate through the standard communications protocol of Microsoft Windows called Dynamic Data Exchange (DDE). This allows the ASTUTE modules to communicate dynamically with any Windows application that can act as a DDE client (e.g., most Windows authoring tools). Object Linking and Embedding (OLE) Automation will also be supported by ASTUTE modules for compatibility with future applications, although not many Windows applications support it at this time.

Using COTS Applications

Since ASTUTE is modular and the communication between modules is through standard Windows protocol, some of the modules are COTS applications. For example, there are several suitable tools for developing the System Simulation. The authors have successfully tried several of these tools. Any CBT lessons developed using COTS applications can be used by ASTUTE. There are many capable CBT and multi-media authoring tools and presentation tools available. For really close communication with ASTUTE and to provide immediate feedback to the student, the tools should support Windows DDE.

The Expert System is integral to ASTUTE. Features of it, such as the hint generator, do not exist commercially. Consequently, an authoring tool for it is being developed by the authors. Using some novel concepts, it is designed for ease of use by instructional designers and subject matter experts with limited assistance from programmers. With only the ASTUTE Expert System and a simulation, a stand-alone ASTUTE practice session or exercise can be created, as shown on the ASTUTE demonstration disk that the authors have developed. For simulation tools that do not support DDE, but do support Dynamic Link Libraries (DLLs), or to provide an easier access to ASTUTE's Expert System, a DLL has been developed to supply ASTUTE functions to communicate with the Expert System.

CONCLUSION

ASTUTE lessons meet the two main goals of the authors' project: (a) to adapt instruction dynamically to observed student needs; and (b) to provide an environment in which skills can be practiced with expert feedback and assistance. By using a detailed cognitive model of the student, the sequence, pace, and course of instruction are adjusted dynamically. The architecture also embodies the sound teaching strategies and coached practice of job skills found in apprenticeship or mentored OJT.

Specifically, the ASTUTE architecture offers:

- Tailored instruction based on the student's current and past performance
  - Teaching what the student needs to be taught
  - Not teaching what the student knows
  - Providing help that stimulates the student's abilities
- An on-line expert that can guide a student through problems
  - A mentor from which the student can learn
  - A coach that explains the rationale behind every step and provides encouraging critique
  - An expert to help explain the system

A strategy of integrating COTS applications to maintain flexibility and reduce cost has also been followed. ASTUTE has been developed as a supplement to other training. The authors have no doubt that this supplement can provide many of the same benefits of cost-prohibitive methods, such as apprenticeship, and provide training on complex systems that cannot easily be taken out of service.

REFERENCES

Anderson, J.R., & Reiser, B.J. (1985) The LISP tutor. Byte, 10(4), 159 - 175.

Carbonell, J.R. (1970). Artificial Intelligence in CAI: An artificial intelligence approach to computer assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4), 19-22.

Collins, A., Brown, J.S., & Newman, S. (1987). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L.B. Resnick (Ed.), Cognition and instruction: Issues and agendas. Hillsdale, NJ: Erlbaum.

Dillenbourg, P. & Self, J. (1992). A framework for learner modeling. *Interactive Learning Environments, 2(2)*, 111-137.

Frye, D., Littman, D.C., & Solloway, E. (1988). The next wave of problems in ITS: Confronting the "user issues" of interface design and system evaluation. In J. Psotka, L.D. Massey, & S.A. Mutter (eds.), *Intelligent* *tutoring systems: Lessons learned* (p 451 - 478). Hillsdale, NJ: Lawrence Erlbaum.

Glaser, R. (1990). The reemergence of learning theory within instructional research. *American Psychologist, 45(1)*, 29 - 39.

Gott, S.P. (1989). Apprenticeship instruction for real-world tasks: The coordination of procedures, mental models, and strategies. In E.Z. Rothkopf (ed.), *Review of Research in Education, 15*, Washington, DC: American Educational Research Association.

Greeno, J.G. (1978). A study of problem solving. In R. Glaser (ed.), *Advances in instructional psychology*, Vol. 1, 13 - 75. Hillsdale, NJ: Erlbaum.

Hart, A. (1986). *Knowledge acquisition for expert systems*. New York: McGraw-Hill.

Hayes-Roth, F., Waterman, D., & Lenat, D. (1983). *Building expert systems*. Reading, MAL Addison-Wesley.

Kaplan, R., & Rock, D. (1995). New directions for intelligent tutoring. AI Expert, 31 - 40.

Lave, J. (1977). Tailor-made experiments and evaluating the intellectual consequences of apprenticeship training. *Quarterly Newsletter of the Institute for Comparative Human Development*, pp. 1-3.

Lesgold, A.M., Lajoie, S.P., Bunzo, M., & Eggan, G.M. (1988). *Sherlock: A coached practice environment for an electronics troubleshooting job*. Paper presented at the J.S. MacDonald Foundation Conference on CAI and ITS, Carnegie-Mellon University, Pittsburgh, PA.

Lesgold, A., Eggan, G., Katz, S., & Rao, G. (1992). Possibilities for assessment using computer-based apprenticeship environments, pp 49-80. In J.W. Regian and V.J. Shute (Eds.), Cognitive approaches to automated instruction. Hillsdale, N.J.: Lawrence Erlbaum.

Loftin,R.B. & Savely,R.T. (1992). Advanced training systems for the next decade and beyond. *Proceedings of the AIAA Space Programs and Technologies Conference,* Huntsville, AL, March 24-27, 1992.

Means, B., & Gott, S.P. (1988). Cognitive task analysis as a basis for tutor development: Articulating abstract knowledge representations. In J. Psotka, L.D. Massey, & S.A. Mutter (eds.), *Intelligent tutoring systems: Lessons learned* (p 35 - 57). Hillsdale, NJ: Lawrence Erlbaum.

Ohlsson, S. (1987). Some principles of intelligent tutoring. In R.W. Lawler & M. Yazdani (eds.), *Artificial intelligence and education* (vol. 1), Norwood, NJ: Ablex.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* San Mateo, CA: Morgan Kaufmann.

Psotka, J., Massey, L.D., & Mutter, S.A. (1988). Introduction. In J. Psotka, L.D. Massey, & S.A. Mutter (eds.), *Intelligent tutoring systems: Lessons learned* (p 1 - 14). Hillsdale, NJ: Lawrence Erlbaum.

Suppes, P. (1967). Some theoretical models for mathematics learning. *Journal of Research and Development in Education, 1*, 5 - 22.

Van Lehn, K. (1988). Student modeling. In M. Polson & J. Richardson (eds.), Foundations of intelligent tutoring systems (p. 55 - 78). Hillsdale, NJ: Lawrence Erlbaum.

Waterman, D. (1986). *A guide to expert systems*. Reading, MA: Addison-Wesley.

White, B.Y. & Frederiksen, J.R. (1986). *Progressions of qualitative models as a foundation for intelligent learning environments*. (report no. 6277), Cambridge, MA: BBN Laboratories.

Woods, P. & Hartley, J.R. (1971). Some learning models for arithmetic tasks and their use in computer-based learning. *British Journal of Educational Psychology, 41 (1)*, 35 - 48.

ACKNOWLEDGMENTS