Technical Report 1125

Automated Tutoring Environment for Command (ATEC): Using an Intelligent Tutor To Model Expert Mentor Interactions

Joan M. Ryder, Jennifer L. DePaul, Wayne W. Zachary, and Vassil Iordanov
CHI Systems, Inc.

February 2002

United States Army Research Institute for the Behavioral and Social Sciences

REPORT DOCUMENTATION PAGE

1. REPORT DATE (dd-mm-yy) February 2002	2. REPORT TYPE Final	3. DATES COVERED (from. . . to) December 2000 to June 2001

4. TITLE AND SUBTITLE Automated Tutoring Environment for Command (ATEC): Using an Intelligent Tutor to Model Expert Mentor Interactions	5a. CONTRACT OR GRANT NUMBER DASW01-01-P-0354
	5b. PROGRAM ELEMENT NUMBER 0602785A
6. AUTHOR(S) Joan M. Ryder, Jennifer L. DePaul, Wayne W. Zachary, and Vassil Iordanov (CHI Systems, Inc.)	5c. PROJECT NUMBER A790
	5d. TASK NUMBER 211
	5e. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) CHI Systems, Inc. 716 N. Bethlehem Pike, Suite 300 Lower Gwynedd, PA 19002	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences ATTN: TAPC-ARI-IK 5001 Eisenhower Avenue Alexandria, VA 22333-5600	10. MONITOR ACRONYM ARI
	11. MONITOR REPORT NUMBER TR1125

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

This report documents Phase I efforts and Phase II plans to develop an Intelligent Tutoring System for Teaching Battlefield Command Reasoning Skills.

14. ABSTRACT *(Maximum 200 words)*: Report developed under a Small Business Innovation Research Program 2000.2 contract for topic OSD00-CR02. The report discusses the feasibility of developing an intelligent tutoring system (ITS) for the interactive training of thinking skills, such as battlefield command reasoning. This ITS will operate within the deliberate practice framework. An "Automated Tutoring Environment for Command" (ATEC) system was designed and a limited prototype was developed to automate the Think Like a Commander (TLAC) materials. The ATEC system is comprised of : (a) a dialog management capability from the AutoTutor system, (b) an instructional agent that replicates the knowledge and role of the human TLAC tutor, and (c) a web-based personalized interface that manages the interaction between instructional agent and trainee. The Phase I research effort reported here has defined the architecture for ATEC. This effort has also demonstrated a proof-of-concept prototype, and has provided a detailed design for a full-scale Phase II system development by September 2003.

15. SUBJECT TERMS

instructional agents, cognitive model, intelligent tutoring system, deliberate practice, Think Like a Commander, web-based instruction, tutorial dialogue, SBIR Report

SECURITY CLASSIFICATION OF			19. LIMITATION OF ABSTRACT	20. NUMBER OF PAGES	21. RESPONSIBLE PERSON (Name and Telephone Number)
16. REPORT Unclassified	17. ABSTRACT Unclassified	18. THIS PAGE Unclassified	Unlimited		Dr. James W. Lussier DSN 464-6928

Technical Report 1125

Automated Tutoring Environment for Command (ATEC): Using an Intelligent Tutor To Model Expert Mentor Interactions

Joan M. Ryder, Jennifer L. DePaul, Wayne W. Zachary, and Vassil Iordanov
CHI Systems, Inc.

Armored Forces Research Unit
Barbara A. Black, Chief

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

February 2002

Army Project Number
20262785A790

Personnel Performance and
Training Technology

FOREWORD

Improvements in Army training and evaluation are an enduring concern of the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI). A related concern addressed in this effort is that military officers may have insufficient opportunity to apply their battle command reasoning skills in realistic battle command situations. One way to provide sufficient opportunity might be through an intelligent tutoring system (ITS) for training battlefield command reasoning skills.

This Phase I Small Business Innovation Research Program (SBIR) effort targeted the design and limited demonstration of an ITS for high-level battlefield command reasoning skills. The research goal was to develop innovative training methods for conceptual skills. Particularly, new ITS techniques and technology for teaching skills that cannot be taught as simple methods and procedures to be followed. The product goal was an ITS for training battlefield command reasoning skills that could be hosted on an interactive web site. Overall, the objectives for developing such an ITS prototype include: anytime, anywhere tutoring; deliberate practice opportunities; standardized instructional procedures; and at least a partial answer to the growing problem of limited expert human tutors.

This research was part of ARI's Future Battlefield Conditions (FBC) team efforts to enhance soldier preparedness through development of training and evaluation methods to meet future battlefield conditions. This report represents efforts for Work Package 211, Techniques and Tools for Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance (C^4ISR) Training of Future Brigade Combat Team Commanders and Staffs (FUTURETRAIN). Results of this effort were briefed to the Director of Bio-Systems, Office of the Director, Defense Research and Engineering. As a result of the Phase I success, the Phase II effort was awarded and an ITS for conceptual skill training should be available for commercial application by September 2003.

MICHAEL G. RUMSEY
Acting Technical Director

AUTOMATED TUTORING ENVIRONMENT FOR COMMAND (ATEC): USING AN INTELLIGENT TUTOR TO MODEL EXPERT MENTOR INTERACTIONS

EXECUTIVE SUMMARY

Research Requirement:

In the 21st century Army, battle commanders will need exceptional and highly adaptive battlefield thinking skills. Today, Army officers' knowledge of battlefield tactics appears to outpace their ability to use that knowledge in an expert manner. Current limitations in skill development include a lack of live instructors or mentors with required thinking and tutorial skills, and the difficulties associated with developing more automated approaches to tutoring battlefield thinking skills. Research directed at improving intelligent tutoring systems (ITS) might help meet this critical training requirement.

Procedure:

Prior to this effort, the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) developed a body of training materials, *Think Like a Commander* (TLAC), that supports human instructor efforts to help train battle command reasoning skills. The TLAC materials deal with battlefield thinking habits that are characteristic of expert tactical thinkers. Though tactical concepts are known and understood conceptually by most officers, often they are not applied in the realistic tactical problem solving of less experienced commanders.

This research and development (R&D) effort focused on design and development of a prototype ITS to automate the human training approach used in the TLAC materials. This Phase I study involved review and adaptation of the TLAC instructional materials to a limited prototype ITS. Key procedural steps in this adaptation included design and partial development of: a dialog management capability, an instructional agent that replicates the knowledge and role of the human TLAC tutor, and a web-based personalized interface that manages the interaction between agent and trainee.

Findings:

Overall, the Phase I effort reported here successfully demonstrated an approach for developing a prototype ITS for battle command thinking skills. Notably, prior ARI research and training development on TLAC provided a solid base for Phase I efforts. The TLAC research base enabled the Phase I team to focus more on "how" an ITS for battle command reasoning might be developed, than on "what" training content was required. Key findings include a better understanding of the research challenges that must be addressed in Phase II efforts. Particularly, the Phase I findings on design and development of dialog management and instructional agent capabilities underscore and direct future research requirements.

Examples of the research issues to be addressed during Phase II include advances in dialog management and tutorial strategies. Typically, a dialog management system determines the interactions between trainee and instructional agent in a well structured ITS training domain. However, the Phase I prototype used a very simple keyword-spotting algorithm, as a Phase II placeholder for a more fully developed dialog management system. Moreover, the dialog management modules for Phase II must operate in a relatively unstructured domain of conceptual reasoning and include language analyzers, a dialog move generator, and an animated agent. Ideally, the natural language analyzers for Phase II will consist of a set of interconnected tools for analyzing and generating individual natural language acts, such as written responses to questions. The Phase II system will also require a dialog turn generation strategy (e.g., questions need answers, assertions need evaluative responses). Similarly, the Phase II system will require a more adaptive tutorial strategy (e.g., when and how to provide hints, prompts, change of topic, ordering of topics), as well as an automated performance assessment of student responses.

Utilization of Findings:

The most immediate use of the Phase I findings were to prepare the Phase II Plan for continued development and funding. These findings were briefed to the Director of Bio-Systems, Office of the Director, Defense Research and Engineering. As a result of the Phase I success, the Phase II effort was awarded, and an ITS for conceptual skill training should be available for government and commercial application by September 2003. Successful completion of the Phase II effort should help meet the Army requirement for more efficient and effective methods for instilling battle command thinking skills. Successful Phase III commercialization should extend similar benefits to a wide range of private and government sector training audiences. Moreover, the research entailed in Phase II development should increase our understanding of basic and applied research issues related to the training and assessment of conceptual reasoning skills.

AUTOMATED TUTORING ENVIRONMENT FOR COMMAND (ATEC): USING AN INTELLIGENT TUTOR TO MODEL EXPERT MENTOR INTERACTIONS

CONTENTS

LIST OF FIGURES

Page

AUTOMATED TUTORING ENVIRONMENT FOR COMMAND (ATEC): USING AN INTELLIGENT TUTOR TO MODEL EXPERT MENTOR INTERACTIONS

Introduction

In the 21st century Army, battle commanders will need exceptional and highly adaptive battlefield thinking skills. Today, Army officers' knowledge of battlefield tactics appears to outpace their ability to use that knowledge in an expert manner. To meet this training requirement, the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) developed an innovative set of instructional materials for helping prospective battlefield commanders further develop their tactical thinking skills. These *Think Like A Commander* (TLAC) instructional materials were designed to support *human* instructors or tutors to train battle command reasoning skills (U.S. Army Research Institute, 2001). This Phase I Small Business Innovation Research Program (SBIR) adapted the TLAC materials to design and partially demonstrate an *automated* intelligent tutoring system (ITS) for high-level battle command reasoning skills. Successful completion of the subsequent Phase II effort should help meet the Army requirement for more efficient and effective methods for instilling battle command thinking skills.

Background

Schoolhouse learning involves primarily declarative knowledge about command principles and tactics, and training in task-specific procedures (in declarative form) based on these principles and tactics. Full-scale exercises and real command situations require integrated 'proceduralized' expertise, including determining what facts and principles are applicable to the problem, retrieving them, mapping the situation to the appropriate parts of the principles, and drawing inferences about the problem situation and its solution (e.g., VanLehn, 1996; Zachary & Ryder, 1997). At first this process is slow and effortful, and the principles are applied one at a time. However, real problem situations require coordinated application of multiple facts and principles. Repeated real-time practice allows for proceduralization and chunking of the skill (deriving domain-specific problem-solving strategies integrating separate pieces of declarative knowledge) and development of automaticity of component skills (Fisk & Rogers, 1992). It takes about 10 years to develop truly expert levels of performance and understanding, such as Klein's level of recognition-primed decision-making (Klein, 1989) where the expertise is such that the appropriate interpretation and course of action can be derived directly (and almost instantaneously) from recognition of key problem-instance features. The TLAC materials address the transition from schoolhouse learning to adaptive expertise by providing deliberate practice experience in a form most likely to facilitate the development of command expertise more rapidly and consistently.

TLAC

The TLAC materials are currently being used with brigade (BDE) command designees attending the School for Command Preparation of the Command and General Staff College at Fort Leavenworth, Kansas, and are being adapted for use at the battalion level and below. The TLAC materials involve presenting tactical situations, or vignettes, as PowerPoint presentations in a classroom setting. Following presentation of a vignette, there is a classroom discussion of the vignette led by an instructor acting as tutor or mentor. The instructor begins by asking indirect general questions to stimulate thinking, then asks increasingly more directed questions to probe for themes that have not been addressed. The discussion is organized around eight themes: (a) focus on mission and higher commander's intent, (b) model a thinking enemy, (c) consider the effects of terrain, (d) use all assets available, (e) consider timing, (f) see the bigger picture, (g) visualize the battlefield, and (h) consider contingencies and remain flexible. A number of vignettes have been created, all within a specific fictitious situational context.

However, the limited availability of live instructors to serve as mentors requires ways of augmenting the TLAC approach, such as an intelligent tutoring system (ITS). The present effort has thus been designed to determine the feasibility of developing an ITS for the TLAC materials. This issue is discussed in the ensuing section.

Feasibility of Developing an ITS for the TLAC Materials

An ITS involves the application of artificial intelligence (AI) technology to computer-based instructional systems. Since the late 1970's, many ITSs have been built, some of which have proven very successful. A set of tutors developed by John Anderson and his colleagues using a model-tracing approach (Anderson, Boyle & Reiser, 1985) are used productively today for such well-structured learning domains as geometry, programming, and physics. However, many more ITSs remain purely research projects, because they deal with restricted aspects of real training problems or are not able to demonstrate instructional value.

The canonical ITS architecture includes, at a minimum, the following three components: (a) an *expert module* containing a representation of the knowledge to be presented and a standard for evaluating trainee performance, (b) a *trainee (student) module* representing the trainee's current understanding of the domain, and (c) an *instructional module* containing pedagogical strategies and guiding the presentation of instructional material (e.g., Polson & Richardson, 1988; Wenger, 1987). These three aspects of an ITS need not be separate components. Current thinking is that the key to intelligent training is that the ITS should behave intelligently in terms of providing adaptive instruction based on diagnosing the trainee's knowledge structure or skills (Shute & Psotka, 1995). The indeterminacy and complexity of domains such as battlefield thinking preclude the use of model tracing approaches to trainee (student) modeling, which are only applicable to procedural learning in well-structured domains. Furthermore, recent theorizing on pedagogical approaches focuses on collaborative learning, situated learning, deliberate practice, constructive learning, distributed interactive simulation, etc., all of which essentially call for adaptations of the traditional ITS paradigm, or creation of alternative types of interactive learning environments.

Another approach to the problem has been to use cognitive modeling technology to create a model of an instructor that can be embedded in an interactive learning environment to provide feedback or instruction in these more complex, indeterminate domains. These models, called *instructional agents*, embody the reasoning of an expert human instructor and include all three aspects of tutoring intelligence—domain knowledge, diagnostic reasoning, and pedagogical reasoning—in one model. The difficulty of diagnosing deficiencies in knowledge and skill or of selecting appropriate pedagogical strategies is not diminished using instructional agents. However, by analyzing the expertise of an instructor using cognitive task analysis methods, and creating an executable model of whatever type of tutorial knowledge is applicable in the domain and instructional problem, the problem becomes more tractable. Cognitive modeling provides a more natural language for representing human expertise than other AI formalisms, and associated cognitive task analysis provides a richer method for acquiring that knowledge than other knowledge engineering techniques. Furthermore, the instructional agent can observe and diagnose selected aspects of the trainee performance, allowing the system to address tractable performance assessment and training problems, while leaving the most difficult aspects for a human instructor (at least making the human instructor more efficient or reducing the number of human instructors needed).

The approach for this problem is to develop an instructional agent that reasons about trainee inputs as a human instructor would, using CHI Systems' COGNET cognitive modeling framework (Zachary, Ryder, Ross & Weiland, 1992) and associated agent development software, iGEN ™. The COGNET provides a description language that can be used to capture and formalize expertise, and iGEN ™ provides a software simulation of the underlying cognitive architecture. The iGEN ™ also provides various tools to support the creation and modification of the cognitive model itself, as well as an Application Program Interface (API), called the communication shell, for communicating with the tutorial interface through which the trainee interacts.

Phase I: Developing the ATEC Approach, Architecture, and Operational Concept

Research Objectives

This research and development effort evaluated the feasibility of developing an ITS for interactive training of thinking skills, such as battlefield command reasoning, within the deliberate practice framework. The corresponding objectives were as follows:

- Create an architecture and operational concept for an approach;
- Implement, integrate, and demonstrate portions of the three key capabilities of the ATEC system;
- Develop a plan for Phase II development.

The following three sections of this report address these objectives in turn.

The first objective of this research was to create an architecture and operational concept for ATEC. The initial proposal for this architecture involved CHI Systems' basic approach to ITS. However, after reviewing the TLAC program and requirements, the present authors realized that the proposed approach required significant change.

The first part of this section deals with the initially proposed approach. This section includes a discussion of the problem analysis that led to a revision in the architecture proposed.

After that, the revised approach is discussed. The final part of this section describes the operational concept accompanying the revised architecture and approach.

Proposed Approach and Architecture

The proposed ATEC approach was to create a deliberate-practice intelligent training system for the TLAC curriculum, integrating the following three technologies:

- CHI Systems' iGEN cognitive agent technology to create an *instructional agent* that subsumes the role of the human mentor/tutor in the TLAC curriculum;

- An *interactive learning environment* adapted from CHI Systems' C3-Core command and control software toolkit into which the agent would be embedded;

- A web-based *personalized interface* using Provant's instructional development tools and expertise to deliver pedagogical interactions between agent and trainee.

Based on the present authors' previous experience with constructing tutoring, this approach seemed to be appropriate for the battlefield-reasoning domain. The key technical challenges, as these authors saw them, were capturing the diagnostic reasoning processes of the instructor and determining how to assess trainee cognitive performance, as well as integrating independent software technologies.

Initial Problem Analysis

An initial analysis from review of the TLAC program, TLAC materials, discussion of ARI goals for the target system, and observation of TLAC courses at Fort Leavenworth led to the following assumptions as to the necessary characteristics of the ATEC: (a) there are no right or wrong answers, (b) the focus should be on knowledge rather than behavior (c) there needs to be a way to assess trainee performance, and (d) there should **not** be a simulated environment. The last point is based on several factors, such as officers do not use complex systems directly, and should not be expected to learn to use such a system for command thinking training.

Based on this initial analysis, the ATEC architecture was revised and expanded. This analysis, most notably, revealed that the planned action-based interactive practice environment was not feasible for this instructional problem. This realization has led to eliminating the C3-Core command and control software as an interactive practice environment. Instead, ATEC would need alternative interaction mechanisms, and methods for assessing performance would need to be primarily language-based rather than action-based. A number of mechanisms for tutorial interaction were then investigated. These mechanisms included: (a) allowing trainees to make information requests, (b) sequencing of increasingly constrained probes beginning with general questions, and (c) constraining responses (e.g., menu selections, multiple choice questions).

Allowing information requests to be made without providing responses seemed unacceptable, and providing the ability to actually respond to information requests was not practical; thus, the first option listed above was eliminated. Sequences of increasingly constrained probes follows the TLAC concept and provides a method for developing an evaluation component, but would require a language-processing component in the system. There are potential methods that could constrain the tutor-trainee interaction to allow an instructional agent to monitor the trainee thought processes without using unconstrained natural language. Constrained interactions eliminate the natural language processing (NLP) requirement but are too leading for general questions, and could only be used when the trainee did not offer much in an area. Form fill-in constrains the language-processing requirement somewhat, but not enough to eliminate it. Thus, it was determined that the only way to develop a system for training battlefield thinking skills was by incorporating language processing into the system and using a tutorial dialog-based approach.

Tutorial dialog systems are more complex than other ITSs, because in addition to standard ITS components, they also need to have a dialog management subsystem with some or all of the following components: a parser, semantic analyzer, dialog planner, language generation, speech recognition, speech generation, and animated agents. Although there has been considerable success in constrained interaction ITSs (Anderson, Corbett, Koedinger & Pelletier, 1995; Wenger, 1987; Shute & Psotka, 1995), natural language-based tutoring systems are in their infancy (other than a few limited NLP tutors built in the 1970's). One dialogue-based tutor, AutoTutor (Graesser, Wiemer-Hastings, K., Wiemer-Hastings, Kreuz, & the Tutoring Research Group [TRG], 1999) has been implemented and used successfully in an introductory course in computer literacy.

The state-of-the-art in computational linguistics has advanced considerably in the last 10 years (Jurafsky & Martin, 2000), as has understanding of tutorial dialog (e.g., Moore, 1995; Person, Graesser, Kreuz, Pomeroy & the TRG, [in preparation]). In computational linguistics, the key recent advances deal with robust methods for natural language understanding in the face of poorly expressed or extra-grammatical input, as is common in tutorial dialog. More robust probabilistic and data-driven algorithms for parsing, part-of-speech tagging, reference resolution and discourse processing have been developed. Also, classification-based approaches using statistical and connectionist methods, including latent semantic analysis, have been advanced recently. Thus, there is increasing interest in building dialog-based tutors, and considerable optimism in the possibilities of success.

Revised Approach and Architecture

Given the above analysis, a revised architecture (Figure 1) for ATEC has been developed that contains a dialog manager/processor. The ATEC (Interface) Server is a web-based personalized interface (as originally planned), and the Vignette-specific Domain Knowledge component, Student Model, and Performance Assessment component are part of the iGEN instructional agent (also as originally planned). The lower box, ATEC Development Tools, is a new component that we propose to develop in Phase II to aid in authoring and editing the knowledge representation of additional vignettes.

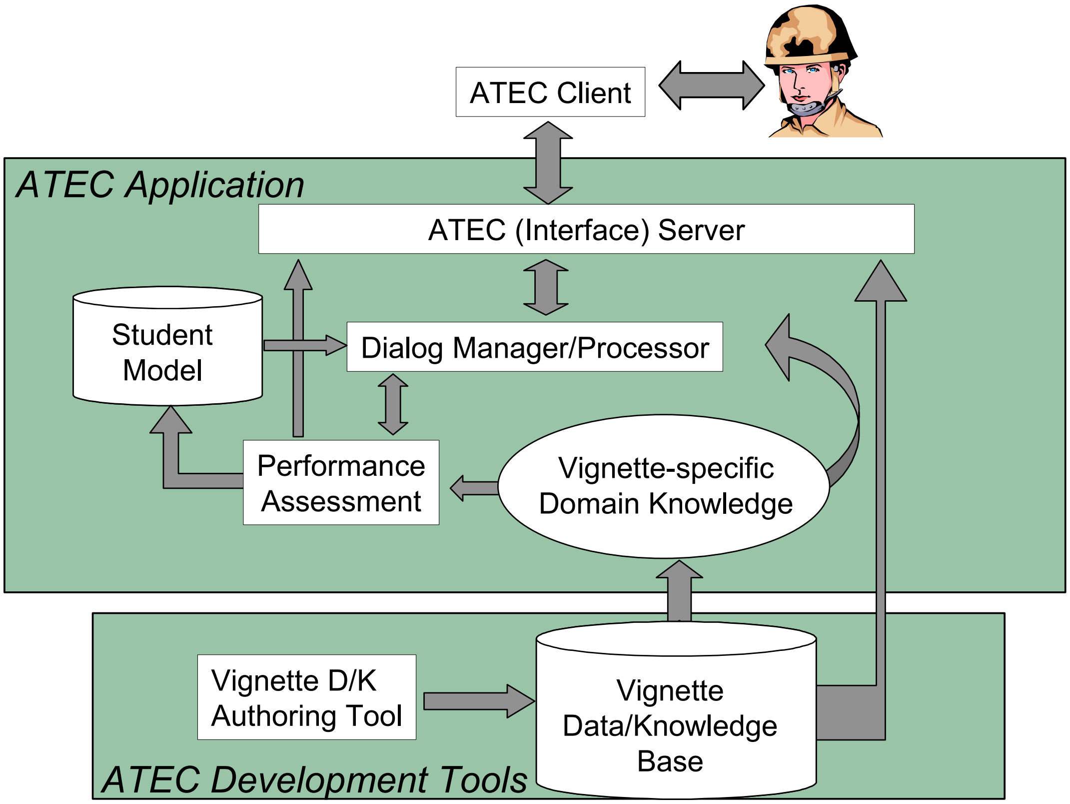

Figure 1. Revised ATEC Architecture

Operational Concept

The ATEC tutoring begins in the same way as the current TLAC program in the classroom setting, with the viewing of the vignette. After the vignette has been viewed, the tutorial dialog begins with the instructor asking a question and waiting for an answer. The ATEC system is entirely dialog-based with constant exchange of instructor questions and trainee responses. At any time during the tutorial interaction, the trainee can re-play the vignette, zoom in or out on the vignette maps, access supporting materials including field manuals, and a brief description of the eight themes to consider while working on the scenario. In addition, the trainee can view any of 30 supporting documents including the 2nd BDE decision support matrix, the BDE offense battle boards, and fragmentary orders or FRAGOs.

The dialog-based concept for the instructional tutor has to address trainee motivation. How can an automated system keep a commander in training engaged and bring him back to use the system? A motivational strategy should probably be built into the tutor that incorporates: (a) evaluation of trainee inputs, (b) elaboration on trainee inputs, and (c) examples of other possible answers. For example, one way for the instructor to communicate evaluative feedback is to provide visual performance feedback to the trainee through varying his facial expression according to the quality of the response to a question. Another way the tutor can keep the trainee engaged is to be intelligent in its selection of questions. The tutor is designed to ask questions about topics that are relevant to what is already being discussed, and to only ask as many questions as it takes to elicit the necessary information to demonstrate the trainee's knowledge level. In this way, the tutor avoids boring the trainee by adapting to his knowledge level and focusing on his weaknesses. These are two examples of motivational features already minimally incorporated into the tutor, although more may be incorporated as development progresses in later phases of the project.

The ATEC system has an instructional agent that controls and maintains the list of which questions should be asked and evaluates what knowledge the trainee has demonstrated. The agent communicates with the trainee and vice versa through an interface where the output dialog is displayed and the input dialog is entered. All of the available supporting documents are also features of this interface. When a message is received by the interface, it is passed into a component of the agent called the shell. It is in this shell that the parsing of the concept keywords takes place. The shell then passes the concept information into the instructional agent for processing. Within the agent, the parsed trainee responses are analyzed to evaluate the specific response and to update a representation of trainee performance, which is maintained as a student model.

An important part of evaluating the feasibility of developing an ITS for battlefield command reasoning was developing a proof-of-concept prototype. This report section describes our prototype, explaining the development process and how the components relate to the planned full system. First, the domain analysis is discussed, then the prototype architecture, followed by a description of each component of the prototype system (language processor, instructional agent, and user interface).

Domain analysis. A methodology for developing an instructional agent requires an in-depth analysis of the instructional reasoning in the domain in which the training is going to take place. The goal of this analysis was to determine the important aspects of the scenario so they could be used to develop questions that afford the trainee the ability to demonstrate decision-making knowledge. For the Phase I prototype, the knowledge of a CHI Systems, Inc. employee with previous army experience has been utilized to inform the domain analysis, rather than submit domain questions to the ARI military contacts. This approach followed ARI guidance to concentrate on process rather than content.

The scope of the domain analysis was limited by working with only one vignette and theme. "The Attack Begins" was the vignette chosen, because earlier research had delineated a set of indicators associated with this vignette. "Use All Assets Available" was the chosen theme.

The subject matter expert (SME) helped to parse the information in the facilitator notes. This person also developed appropriate question and answer pairs for the aforementioned information. The questions were constructed in a hierarchical tree-like structure beginning with general questions and progressing to specific questions. An example of one branch of questions is as follows:

0: Given the situation as described, what assets are available to help you in this situation?
1: What resources already assigned to the 6-502nd can you call upon for help?
1.1: What 6-502nd assets are available to help you?

1.1.1: How can the company on the left flank provide help?
1.1.1.1: Did you consider having the company on the left flank suppress the enemy's operations?
1.1.1.2: Did you consider having the company on the left flank continue their assigned mission to secure the bridge?
1.1.2: What support might the company on the left flank need to continue their mission?
1.1.2.1: Did you consider calling in artillery from 4-41 field artillery to help?
1.1.2.2: Did you consider calling in aviation from B/1-55 to help?
1.1.3: What can the survivors of the companies that have been hit do?
1.1.3.1: Did you consider having the companies under fire help themselves?

The tree diagram shown in Figure 2 shows where each question fits into a small piece of the tree structure.

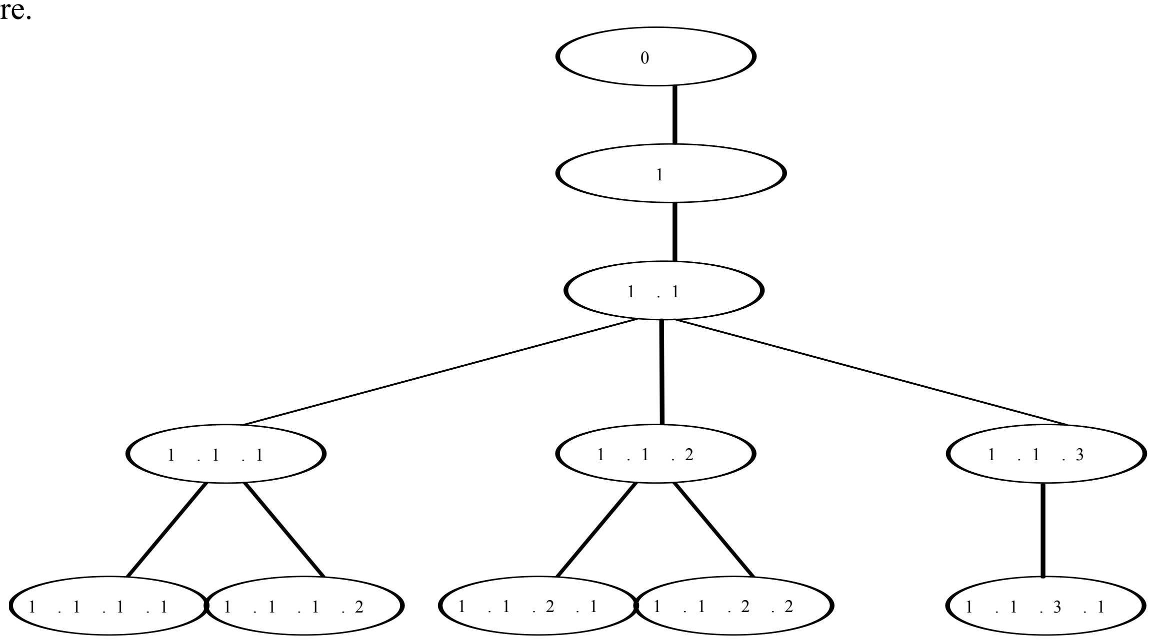

Figure 2. Example Tree Diagram

The full set of questions developed and corresponding tree diagram are provided in Appendix A. For each question, the SME associated concepts that should be represented in the answer to prove that trainees have fully demonstrated knowledge of the topic. For example, a trainee's answers for the question dealing with available assets for the 6-502nd should include the terms: (a) company, (b) left flank, and (c) survivors; otherwise the trainee has not demonstrated that he/she understood the tactical concept associated with this question. The complete set of concepts for each question is provided in Appendix B.

In addition, the SME indicated a set of keyword synonyms for each concept. The word "Pickup" could mean "Pickup area," "Secure pickup," "Secure area." The synonym list includes common misspellings so that they are recognized. The full synonym list is provided in Appendix C.

Prototype architecture. The prototype architecture is shown in Figure 3. It contains three major components: (a) a language processor, (b) an iGEN instructional agent, and (c) a web-based user interface. Each component is discussed below.

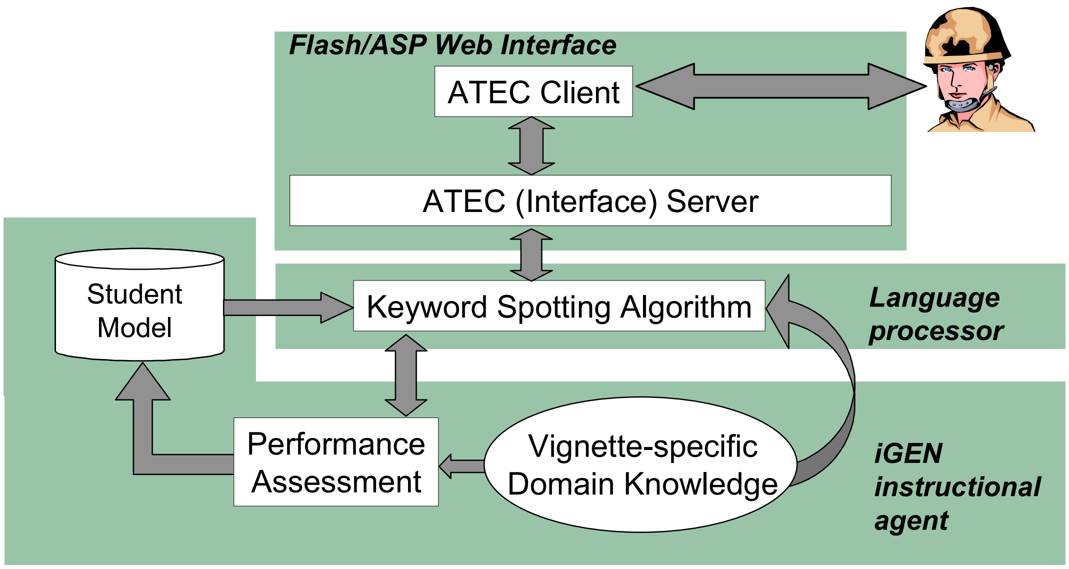

Figure 3. Prototype Architecture.

Language processor. As mentioned previously, the ATEC agent is a dialog-based tutor. In order to appropriately handle the free text input of the trainee, a language-processing tool is needed. For the purposes of the Phase I prototype, a text parsing keyword-matching tool in the shell of the agent has been built to handle the inputs of trainees. So, for the example above, trainees could type any response that they chose but the agent would only search to see if the indicated concept matching keywords were present. This simple keyword tool allows ATEC to parse trainees' inputs at a minimal level for demonstration purposes within the scope of the Phase I effort, but this example clearly demonstrates the need for a more sophisticated language analysis capability in Phase II. Using the question tree and keyword matching tool, a tutorial strategy has been developed for the prototype that is limited but extensible.

Instructional agent. The ATEC instructional agent maintains a tree-like representation of trainees' knowledge that is referred to as the student model. This student model matches the structure of the question tree and allows the instructional agent to monitor trainees' responses to each particular question by populating the branch of the tree that specifically correlates with the stated question. Further, the agent considers all responses given by a trainee and populates any nodes of the tree with concepts mentioned in any response. Through this two step process, as trainees respond to questions, the instructional agent analyzes the answers and the student model gets updated to include any new relevant information the trainee provided. By maintaining a student model that is constantly updated, the instructional agent is aware of the current level of knowledge demonstrated by trainees at any point.

After the vignette has been viewed, the instructional agent begins to ask the trainee a series of open-ended non-leading questions about the current situation. The questions become more specific as the trainee provides more information regarding his understanding of the situation. Once the trainee's model is populated with a predetermined level of knowledge on any given theme, the model moves to a new theme. If the trainee does not provide the necessary level of knowledge about any given theme after several progressively leading questions, a final question is posed. This final question contains the piece of information the instructional agent was expecting. After the last question on each theme is answered, the instructional agent moves to a new topic area and begins with the highest-level question that is appropriate to ask (in essence, traversing the question tree shown in Figure A-1 from top to bottom and left to right). The agent actually checks all of the keyword synonyms of the concept to tell if the concept has been represented in a trainee's answer or not.

In order to evaluate trainee performance, the instructional agent monitors the student model to determine when the trainee has answered a question adequately to move on to a new question. In choosing the next question, the agent evaluates the concept keyword list that matches the next lower node of the tree to see if has already been completed. If the next node is complete, the agent skips that question and goes to the next lower node until the branch is completed, then the agent moves to the next branch and starts evaluating the highest node question. This process continues until all nodes of the tree have been checked and are populated with matching concepts. The instructional agent also maintains a record of which questions were asked and which questions elicited the matching concepts to use in evaluation. In conducting the evaluation, the agent compares how far down the tree or how specific and leading the questions had to become before the trainee demonstrated satisfactory understanding of the relevant concepts. For further evaluation, the instructor agent looks at the completeness of the information provided.

User interface. The user interface consists of a section of the screen reserved for displaying information including the vignette, field manuals, and other forms including the commander's critical information requirements (also known as the CCIRs), FRAGOs, the eight themes description, etc. There are buttons allowing the vignette to be replayed and exited at any time. There are also map manipulation buttons that provide the ability to zoom in or out on the different maps of the vignette.

There are additional buttons for the following functions: (a) description of the eight themes (b) show the orders, (c) view map, and (d) download relevant documents (docs). The button for "orders" brings up a list of thirty documents including the 2nd BDE decision support matrix, the BDE offense battle boards, and FRAGOs. This list is displayed in the information display section of the screen, as is any selected document. The button for the view map simply sets the information display back to the map view. Finally, the button to 'download docs' brings up a list of field manuals that can be viewed on-line. There is a space for a talking head where the narrator of the vignette appears during the scenario presentation and the instructor appears during the tutoring interaction. The talking head has different facial expressions that are displayed based on feedback from the instructional agent regarding the quality of the response. Finally, there is a section of the screen reserved for dialog boxes for the tutor output and the trainee input. These are not shown during the vignette presentation. In the current prototype, all of the interaction occurs via typed text. The instructor's questions are presented as text in a box and the trainee must type his responses in a text box.

The interface was designed by KC-EP, a Provant company, using Macromedia Flash 5.0 and Adobe Photoshop 6.0. Photoshop was used for creating and editing images to be imported into Flash. Flash was selected because of its the personalization and flexibility.

ATEC Detailed Design

The third objective of the Phase I effort was design of a full system that could be built in a Phase II effort. This objective involved creating a design that builds from the Phase I analyses and prototype. It also involved determining the possibility of implementing a major component of the feasibility analysis.

The initial Phase I architecture did not incorporate a natural language-based interaction component. Indeed, as (Graesser, et al., 1999) notes, this is typical for intelligent tutoring approaches. However, the Phase I research did identify a clear requirement for an interface between ATEC and the trainee that was not only natural language based, but that also engaged the trainee in relatively complex dialog which facilitated the learning and introspection process on the part of the trainee. This type of interaction requires a sophisticated dialog management strategy. While all other components of the ATEC architecture were at least partially implemented in Phase I, the dialog management approach was only approached in a 'placeholder' manner, using a minimal keyword-spotting algorithm. In the Phase II approach, the development of a suitable natural language-based dialog processor and manager represents a major portion of the effort. The ATEC design, detailed below, calls for adaptation and integration of a specific and proven existing approach/technology called AutoTutor. As background and justification for this decision, the literature on dialog management research and technology is briefly reviewed below, followed by an (also brief) overview of AutoTutor.

Dialog management. Interactions between two agents (whether person or machine) can be organized into a hierarchy according to the temporal depth and complexity involved. At one end lie simple communicative acts (declarative utterances, simple commands, information displays, etc.) that express information. When two communicative acts are directly connected (e.g., question-answer, command-affirmation), then they constitute a transaction. These are the levels at which most human-computer interaction research and technology has focused. However, the transactions themselves can be organized into a larger purposive task, the subject of which may be the transactions themselves (e.g., a conversation) or some other domain-situated activity (e.g., some team-work task, or, in the case of interest here, tutoring). These connected sets of transactions constitute a dialog; when the dialog is discontinuous over time and involves multiple parties, it is often termed discourse.

The key issue is that each higher level involves additional levels of knowledge and cognitive processing. Traditional interactive computer systems (regardless of the application) typically simply catalog the transaction types, and provide a separate and independent mechanism for each. In order to engage in a sustained purposive dialog, however, the organization of the transactions over time must be explicitly managed. Over the years, two main computational paradigms for dialog management have been advanced. Both rely, to varying degrees, on the notion that the parties engaged in the dialog must have some sharing of knowledge or mental model about the dialog and the task in which it is situated. These two approaches can be termed as the intent inference approach and a situational approach.

The intent inference approach is conceptually based on the AI paradigm of plan recognition (Schmidt, Sridharan & Goodson, 1978). In plan recognition, one agent (e.g., a machine) observes the behaviors (communicative and otherwise) of another (e.g., the human 'user') and attempts to infer the intentions and ideally the plan held by that other agent, typically in terms of some task/domain-based canonical plan or planning space. Once this plan and intent are inferred, they become the basis for the first agent to form its presumed shared mental model of the underlying process. The first agent uses this model to guide the management of the dialog (e.g., Jones, Mitchell & Rubin, 1988; Vaubel & Gettys, 1990). The widely used 'knowledge tracing' approach in intelligent tutoring systems can, in fact, be seen as a special case of the intent inference approach. While an attractive paradigm, attempts to develop a robust interaction and dialog management process from intent inference have generally not succeeded outside of very narrow domains. There are two main reasons for this. The first is the fact that plan recognition seems, by itself, too weak to carry out the intent inference task. The actions people take (verbal or instrumental) are prone to multiple interpretations; even people frequently experience profound miscommunications as a result (i.e., when they realize that they have been talking about completely different subjects without being aware of it). To compensate for this, people typically engage in various actions, most commonly speech acts, to test, assess, and repair the shared understanding (e.g., "do you know what I'm saying?" "which artillery unit are you talking about?" etc.). Suchman (1990), among others, has shown how people assume this process in interactions with 'smart' machines, and how disastrous the results can be when the machine is unable to extend its dialog to repair a faulty model of the user. The second reason, paradoxically, is that full intent inference may be too ambitious a goal (even if it were achievable). That is, humans seem to engage in cooperative coherent dialogs without attempting to fully model the plans and intentions of the other (s) involved. When people collaborate in a work context, they appear to base their interactions and dialogs not on detailed models of each other's mental states, but rather on their presumptively shared understanding of the work/situational context. They use that shared context representation to interpret each other's actions and behave collaboratively (Robertson, Zachary & Black, 1990; Eisenberg, 1990), based on inferences about what would be reasonable/expected given the role/relationship of the other person or agent in that situation.

The situational approach builds on these two limitations of intent inference. It assumes that the agents engaged in a purposive dialog need only share an understanding of the domain context in which the dialog is unfolding and of the situation within the dialog itself. This shared understanding must be actively maintained through an interactive process of confirming shared understanding, checking specific elements, and actively repairing any ambiguities and/or misunderstandings that occur (and that are meaningful to the management of the dialog). The actual dialog process is managed by knowledge about how each party is expected to behave (e.g., the tutor acts like a tutor and the trainee like a trainee, or that a confused trainee can be brought back into line through a set of established strategies) and proceeds by applying that knowledge dynamically, given each individual's role and relationship to the situation.

The general dialog management problem is made more difficult in practice by the orthogonal issue that, when the agents involved are human, dialog and discourse are typically situated in the medium of natural language (Clark, 1996). This has posed a problem for intelligent computational systems, such as intelligent tutoring systems or performance-support/decision support systems (alternatively called expert systems, advisors, critics, intelligent interfaces, associates, etc.). Often, such systems (whether using an intent inference or a situational approach to dialog management) have focused on domains where the medium is either purely symbolic communication (e.g., mathematics, physics, computer programming) or graphical/iconic communication (power plant operations, telecommunications network management, etc.) or where the language used is highly restricted (e.g., command and control). The major successes of the knowledge-tracing variant of intent inference in tutoring (e.g., Anderson et al., 1995; Van Lehn, 1990; Lesgold, Lajoie, Bunzo, & Eggan, 1992) have all come in such cases. On the other hand, the CHI Systems research team has substantial success with the situational approach (using the COGNET/iGEN technology) in both tutoring and performance support domains (Zachary, Ryder, Santarelli, & Weiland, 2000).

Within the context of ATEC, the Phase I research has shown that natural language is a required medium of interaction, and that the interactions between the tutor (whether human or automated) and trainee in the TLAC curriculum are not transactional but rather need to create a facilitative dialog. Given the discussion above, any intent inference approach will not likely succeed. A situational approach is thus needed. Among the most advanced examples of a situation-based approach to natural language dialog management in a tutoring context is the AutoTutor system, developed by Graesser and colleagues at the University of Memphis (Graesser et al., 1999; Person et al., [in preparation]).

Overview of the AutoTutor. The AutoTutor is an interactive tutoring system that teaches a college-level course on the fundamentals of computer literacy. The main features of AutoTutor are its ability to engage the trainee in dialogs that simulate those of human tutors, and its opportunistic use of heterogeneous computational methods. The system has progressed through at least two major revisions (AutoTutor1, which simulates simple tutoring strategies and AutoTutor2, which adds more sophisticated ones). This Java-based system has five major components:

- <u>Curriculum script</u> – these are structured lesson-plan like knowledge elements that define the lesson concepts, canonical questions, cases and problems that comprise individual lessons, organized into higher level topics and macrotopics. The curriculum script includes ideal answers and corrective 'splices' for expected incorrect answers.

- <u>Language analyzers</u> – this is a set of interconnected tools for analyzing and generating individual natural language communicative acts, consisting of the following:

(a) word and punctuation segmenter, which breaks down the trainee's utterances into syntactic, lexical, and punctuation components; (b) syntactic class identifier, which takes the results of a segmentation analysis and identifies the possible syntactic classes to which each element could belong; and (c) speech act classifier, which uses a neural net to classify the utterance into one of several speech act categories (e.g., assertion, what/where/why/etc. question, directive, yes/no question, declarative response).

- Domain knowledge – the AutoTutor requires knowledge about the domain in which it is tutoring and interacting with the trainee. Because domain knowledge acquisition was not a focus in AutoTutor research, an implicit domain knowledge representation was developed using Latent Semantic Analysis (LSA) of a large corpus of material on elementary computer concepts and computer literacy (Landauer & Dumais, 1997). However, the use of LSA dimensional space is not an inherent part of AutoTutor, and the domain knowledge within the system could alternatively be represented as a set of formal propositions.

- Dialog move generator – this is a combination of a fuzzy rule-based state transition network that situates the dialog in a higher-level model of the tutor-trainee interaction (a situational role model, as discussed earlier). It uses the inputs from the domain knowledge and language analyzers to determine the next type or range of utterance-types expected from the trainee, and to determine what the next type of speech act should be undertaken by the AutoTutor.

- Animated agent – this is a rudimentary visualization of the tutor (i.e., a 'talking head') that also provides speech interaction and some capability to undertake pointing gestures and express facial emotions. The animated agent, developed from COTS software, is the medium through which AutoTutor interacts with the trainee.

The interaction between the AutoTutor and human trainee is captured as portions of the state transition network in the dialog move generator. The overall dialog model of the tutoring strategy is also represented in the same way. For example, AutoTutor1's strategy includes such kinds of 'moves' as: (a) providing positive (or negative or neutral) immediate feedback, (b) pumping for more information, (c) providing hints, (d) elaborating on a trainee's response, (e) correcting content after a trainee error, or (f) summarizing recent utterances/transactions.

The overall curriculum script is thus applied in the context of a tutoring strategy, while the shared understanding of the situation is developed, maintained, confirmed, and repaired on an ad-hoc basis.

Phase II: Architecture and Technical Approach

The Phase II design is based on the revised ATEC architecture presented in Figure 1 above. The architecture is partitioned into four groups of components. The components in each set perform a common function (and could thus potentially also be thought of as representing a separate subsystem), and are also implemented using a similar technology. The user interface components are implemented as Flash and Java. The dialog manager/processor is implemented as a derivative of the AutoTutor system. The domain knowledge and reasoning capability about each vignette, and the related performance assessment component and student model are implemented as intelligent processes in iGEN. The tools to author and store information about each vignette are stand-alone processes implemented in C++ (the authoring tools) and Access (the database). This functional/development organization is pictured in Figure 4. The remainder of this section details the approach taken to creating the various pieces of this architecture.

User Interface

An initial version of the user interface subsystem was developed and incorporated into the Phase I conceptual prototype. The interface features: (a) a display map panel where the vignette is displayed, (b) control buttons to play the vignette (and/or re-play, zoom-in, zoom-out, stop), (c) a 'talking head' box where (emotive) narrator and instructor images appear at appropriate times, (d) dialog boxes for tutor output and trainee input, and (e)

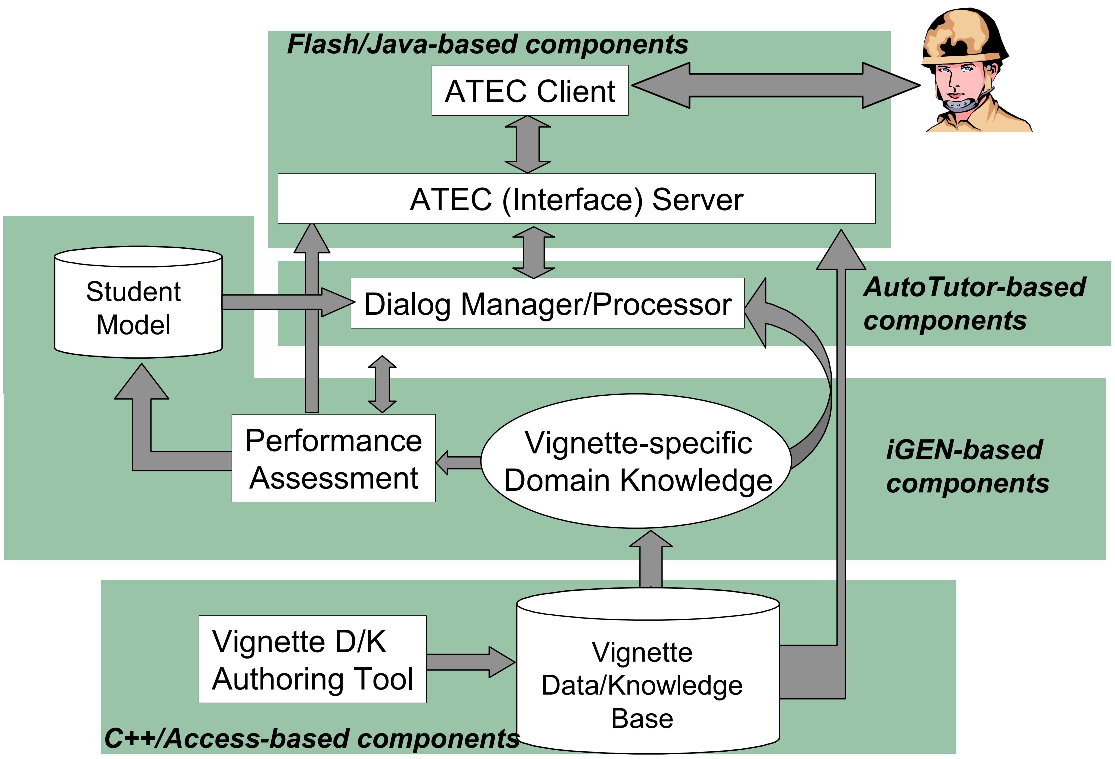

Figure 4. ATEC Functional/Developmental Organization

buttons that allow the trainee to link to supplementary materials (e.g., OP Order). The Phase II ATEC user interface subsystem will evolve forward from the Phase I design and implementation. Because of the web-based nature of ATEC, the complete interface subsystem consists of client-side and server-side components, implemented in Flash and Java.

The existing software needs to be modified and extended in three ways. First, the server-side implementation will be extended for full web-server functionality, allowing multiple simultaneous users with appropriately scalable server technology. Second, the actual user-interface design will be modified and extended, as required, by the development of the instructional management component, the trainee performance assessment component, and the dialog management component. Third, a number of possible features that could enhance the quality of the interface, the user-experience, and the robustness of the interaction will be investigated and incorporated if warranted. These include integration of a higher quality map display and map server, speech output to the user, speech input by the user (by means of voice recognition technology), and allowing pointing inputs (i.e., references to specific or general map locations or features).

Dialog Management

Development of a full dialog management subsystem will involve adapting components of the AutoTutor system to the ATEC system architecture and domain. Models of tutoring dialog that are used in AutoTutor were developed from empirical analyses of human tutors in the computer literacy curriculum that the system addresses (Graesser et al., 1999). The nature of the tutoring process in TLAC is generally similar, but also poses some key differences. These are:

- The use of a more facilitative form of tutoring, in which the emphasis is introspection and application of abstract principles to a specific vignette, rather than elicitation and learning of specific facts, relationships, and concepts; and

- The presence of spatial references in the dialog, such as terrain references ("that ridge over there can provide terrain masking") and unit references ("the artillery battery between the 6-502nd and the 5198th"). These may require additional dialog to establish, maintain, and, when necessary, repair understanding of the spatial referents (Schober, 1993, 1998).

These aspects of tutoring dialog are not dealt with in the current AutoTutor. The incorporation of AutoTutor technology into ATEC will thus require additional empirical research and dialog/discourse modeling, so that these behaviors can be incorporated into the ATEC dialog management development.

The goal of the modeling process is the design of a Dialog Advancer Network (DAN) component for the dialog move generator component. The DAN is a model of the general structure of a tutoring dialog within the curricular domain at hand. It is a state transition diagram in which each node represents a kind of transactional 'move', with fuzzy rules at each node for determining which state should be considered and selected as the next tutor speech act. Figure 5 shows an example of the DAN in AutoTutor1 (Taken from Person, Graesser, Harter, & the TRG, 2000).

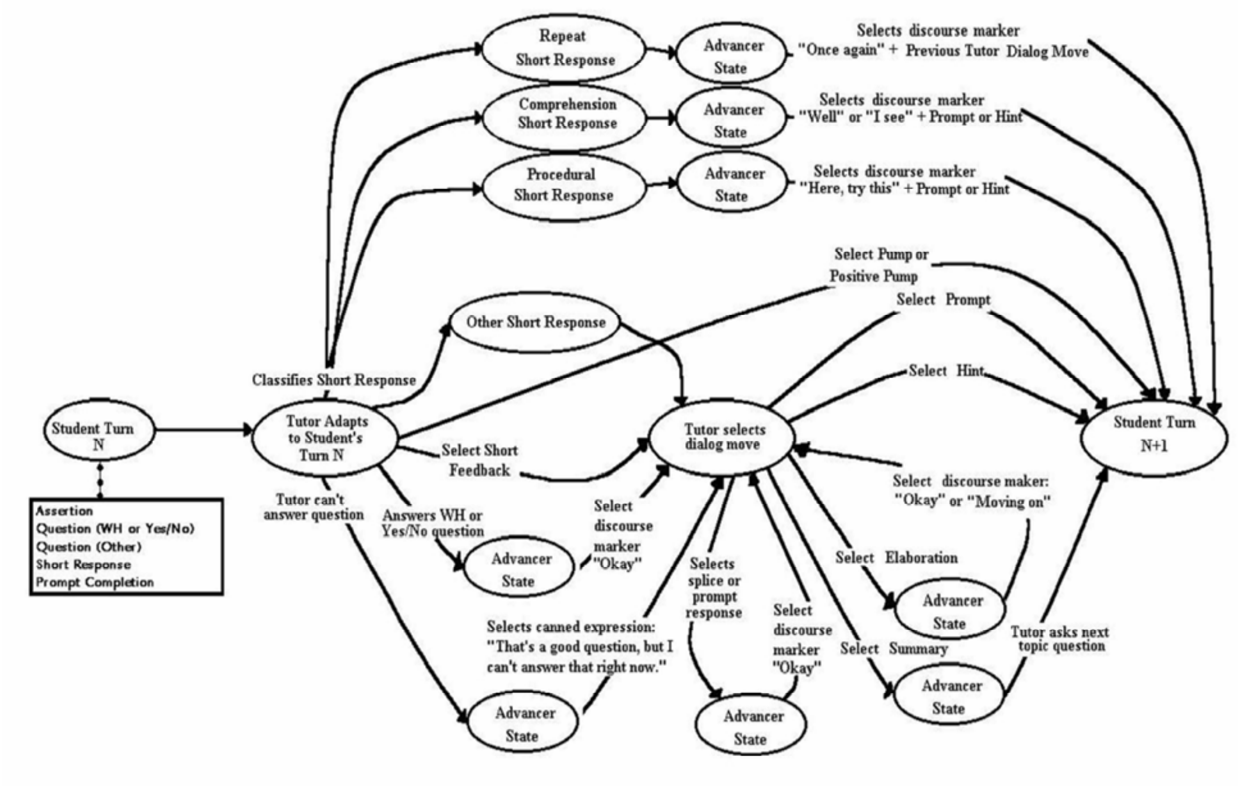

Figure 5. Example Dialog Advancer Network from AutoTutor

The data for these analyses can be drawn from several sources. The data relative to the TLAC facilitative introspective tutoring approach can be taken from recordings of TLAC classroom sessions. These can be recorded unobtrusively via video (although audio-alone would also be adequate), for a maximum of 24 hours of video. (The number is based on data provided by Graesser (personal communication, May 2001) from the experience at the TRG laboratory). If it proves practical to segment the spatial dialog element analysis from the tutoring dialog analyses, then data for the spatial reference analysis can be collected. These data would come from simulated TLAC interactions using local subject matter experts with Army command experience.

From these data, an iterative six-step analysis and modeling process will be applied:

1. Speech act and content analysis. The dialogs are broken into communicative acts and transactions, which form the units of analysis. These are then analyzed to identify classes of speech acts (e.g., providing immediate feedback, providing hints, pumping for more information, etc., for the tutor). It is expected that there will be some overlap between the speech acts used in the current AutoTutor and those found in the TLAC dialogs, but also some which are unique to the tutoring style of TLAC.

2. Develop transition/dialog model. Once the speech act categories are developed in step 1, the individual communicative acts in each dialog will be assigned to a speech act category. Next, the transactions are categorized as transitions between speech-act states. The results will be used to develop the state transition network that forms the skeleton of the DAN being constructed. The next set of rules determines the transition logic (when a given state may transition to more

than one 'next' state). The raw data are re-examined to reconstruct the situation/context/domain knowledge that was used to determine a successor state, given the vignette involved and history of the dialog to date. Specific transition rules will be developed from this analysis. In cases where the transition is indeterminate, the transition rule will be represented as a fuzzy rule. When the state transition network and all transition rules are defined, the current iteration of the DAN is complete.

3. Apply model to test data. Using either original data reserved from the analysis in Steps 1 and 2 or additional data (approximately 25% of the original data set in size), the DAN model will be used manually to recreate/predict the empirical transitions observed. This is done by segmenting the data as in Step 1, and then using the DAN to predict, after each communicative act, what the next type of act (or set of choices, in the fuzzy rule cases) should be.

4. Observer results. The results of the manual test application of the model are then used to evaluate the model, with emphasis on the cases where the model failed to predict correctly. If the model fails to predict above some threshold (for specific transitions or overall, the process proceeds to Step 5. Otherwise, it terminates, with the current DAN being accepted as an adequate dialog model.

5. Revise/extend model to correct errors. The additional data set used in Steps 3 and 4 is then subjected to the analysis of Step 1 and 2, to identify additional speech act categories and/or additional or more fine-grained transition rules. This will result in a refined DAN model.

6. Revert to Step 3.

Data from past experiences with this method at the TRG laboratory suggest that two or at most three iterations of this process will likely be sufficient, with a substantial possibility that an adequate fit may be achieved after even one iteration.

The ATEC dialog management component will be developed as a set of extensions and modifications to the AutoTutor system described above. The proposed extensions are summarized below in terms of the major components of AutoTutor as listed above.

- Curriculum script. The current curriculum script contains data and knowledge relevant to the computer literacy curriculum currently supported by AutoTutor. In Phase II, the TLAC curricular elements will be translated to the curriculum script format used by AutoTutor. It is expected that the themes level of the TLAC curriculum will be translated as the macrotopics, and that the elements within each theme will translate as the topics under the macrotopics. The vignettes will then become lessons. Additional information required by the AutoTutor curriculum script will have to be generated, such as links to appropriate domain knowledge elements, examples of complete or ideal responses/solutions for each theme in each vignette, corrective texts and vignette-specific knowledge elements for common or expected incorrect replies, initiating questions/probes for the various curricular elements, etc. To a large degree, this is simply a continuation of the process that was begun in Phase I, but focused on a specific format and specific set of requirements.

- Language analyzer. Most of the existing tools for language analysis in AutoTutor should be directly incorporated into ATEC. However, it will also require various extensions to ensure that these tools cover the Army command and control domain, which is quite different in lexicon and possibly in syntactic classes and speech act classes as well. Any additional speech act classes for the domain will emerge from the analysis described above, and the speech act classifier will be enhanced as needed to recognize these classes. Additional lexical and syntactic elements will be identified through analysis of a relatively small existing corpus of information, consisting of field manuals (e.g., FM101-5), the TLAC training curriculum and vignettes.

- Domain knowledge. The current AutoTutor uses an implicit representation of the computer technology domain that is created through LSA. While this approach is attractive if a suitably large corpus of relevant material exists, we do not believe this to be the case (the relatively small corpus discussed in the preceding bullet is too small, and too un-representative of the tutoring dialog contents). Thus, this computer-based LSA domain knowledge model will be replaced with an interface to the formal domain knowledge representation contained in the iGEN portions of the system. This interface should involve access to the domain knowledge blackboard structure, and probably the dynamic formulation of specific rules/propositions for use by the language analyzer and dialog move generator. In the event that this dynamic approach (which would be more generalizable) cannot work, a hard coding of domain knowledge propositions will be done within the dialog management system itself.

- Dialog move generator. The infrastructure for the use of a DAN (Figure 5 above) that is currently within AutoTutor can and will be retained in as complete a manner as possible. However, the actual DAN model for ATEC will be substantially new. The development of this model has been discussed above. The new DAN model will be integrated with the existing infrastructure already in AutoTutor. An example of this might be support for incorporation/processing of pointing data from the ATEC interface, to enable disambiguation of spatial references on the map underlying the vignette.

- Animated agent. The Microsoft Agent-based 'talking head' interface will be completely removed from AutoTutor, and will be replaced with the ATEC interface developed in Phase I. Although developed completely independently, there is a high degree of compatibility between the features of the current ATEC interface (see above) and those in the AutoTutor animated agent. This compatibility should make the replacement process straightforward, primarily involving modifying the various calls/call-backs associated with the current agent and replacing them with alternative calls to the ATEC interface.

Instructional Agent

Domain knowledge. Ultimately, the ability of ATEC (or any tutoring system) to provide both training and performance evaluation depends on its possession of appropriate knowledge of the training domain. The process of representing knowledge about the Army command and control process was begun in Phase I, and resulted in the extraction of an initial hierarchy of concepts that the system would seek in trainees' responses to ATEC's questions. These concepts represent both declarative and relational propositions that reflect the direct or inferred application of one specific TLAC theme to the particular facts and details of a specific vignette. For a full ATEC system, this analysis must be extended to consider all eight themes and all the vignettes. It must also support a much more robust natural language interaction capability.

The representation of domain knowledge for a full ATEC also has two additional requirements. First, it must be easily applicable to additional vignettes that may be developed in the future. The vignette knowledge editing tools discussed below address this requirement. Second, it should be defined at a dynamic and computable level to the maximum degree possible, (rather than just dependent on pre-defined 'markups' of the vignette). This requirement would allow the system to determine, for example, the applicability of a concept such as 'terrain masking' to be applied dynamically to a trainee's assertion. Consequently then, the system can determine whether the: (a) underlying concept of the trainee is applicable and relevant, and (b) the assertion is true. Unless the concept is defined in such a way that the veracity can be determined on the fly from the points given, either the system would have to ignore the accuracy of the assertion (which may go to the heart of whether the trainee knows how to apply the concept *correctly*), or would somehow require all possible terrain features to be pre-calculated and stored.

The second requirement, in particular, will constrain the approach to defining and representing domain knowledge. The basic approach will construct hierarchical models of applicable domain concepts, using the declarative knowledge representation in COGNET/iGEN, and using the semantic association or 'linking' capability to associate specific concepts (such as visibility) both to themes, and to more specific and more abstract concepts at other levels in the hierarchy. At the lowest level in the hierarchy are features of the vignettes, which may be either simply declared or created by domain reasoning processes. For example, we might define specific testable assertions as procedural knowledge (e.g., visible from) that can be instantiated and evaluated in the context of existing knowledge of the vignette, or data about it (e.g., a map). In all cases, the assertions and knowledge elements will be defined only as needed to support the assessment of trainee performance, and the evaluation of trainee assertions (and other speech acts) from a domain perspective.

The development of many of these dynamically testable assertions could be simplified by appropriating modules from CHI Systems' C3Core system (http://hawk.chiinc.com). This system provides components that can compute or instantiate many basic C2 concepts, such as terrain mobility limitations, visibility, mobility corridors based on unit type and size, fire envelopes, etc.

Performance assessment and student model. In the Phase I prototype, performance assessment was accomplished by determining the depth of questioning needed to get evidence that a concept was discussed. This assessment was maintained as a student model, but was not used to generate feedback or a final assessment. In a full system, the performance assessment can be extended in conjunction with an expanded domain knowledge representation and the dialog management capability. Rather than merely counting the number of concepts used in an answer, the actual assertions of the answer can be evaluated and specific evaluative comments provided by the virtual instructor. Ideally, this can provide context to the Dialog Advancer Network which determines the flow of transaction types within the dialog, which will help it decide among alternative possible responses or elaborations as part of the interactive dialog (such as types of responses other trainees may have given in the classroom setting). The student model will need to evolve in conjunction with the evolution of the performance assessment capability.

In addition to on-going assessment of each trainee response, a full system should have a summary performance evaluation screen for each vignette. The assessment will probably be based on the eight themes. It may be as simple as a numeric score for each theme, with scores based on the depth of questioning needed to get good coverage of concepts in the area. However, it may be possible to develop a richer overall assessment based on other aspects of the dialog and introspection process, which will be determined as the effort progresses. In addition, there needs to be coordination with the evaluation project being conducted by ARI-Leavenworth to determine if additional aspects of assessment can or should be incorporated.

Another aspect of performance assessment will be maintaining a trainee record across vignettes to allow evaluation of progress. The outputs from this may be trend graphs or tabular formats showing progress in depth of responses over time (presentation of such objects would require extensions to the ATEC user interface). It will be possible to maintain background data on the trainees (e.g., branch, rank, type of organization) and use this information to adapt ATEC's questions and responses based on the individual's background or past performance. Information in the trainee record can also provide additional context to the DAN, allowing it to vary its responses based on the trainee's prior interactions or progress in the course.

Authoring Tools

The TLAC materials should continue to be a dynamic entity, evolving both its instructional content and adding additional lessons (i.e., vignettes). This means that an operational ATEC system will need to allow the vignette data to be expanded and edited, and in a manner which curriculum designers, not AI programmers or cognitive scientists can do. To this end, development of a set of graphical tools for authoring and editing the information on TLAC vignettes will be needed by the ATEC system, and a capability for storing these data in a commercially available database management system.

The training content on vignettes that will be required by ATEC includes: (a) relevant domain knowledge (b) map/positioning data (c) descriptive texts, narratives, etc., for trainee background on the vignette; and (d) other notes and data (such as revision dates, author name, etc.). Each type of content will make use of a separate authoring/editing tool, based on its content. Of these, the most problematic is the tool for authoring and editing relevant domain knowledge.

A domain knowledge editor is envisioned that would be visually organized around the hierarchy and link structure. This visual organization will define, in essence, a 'blank' knowledge structure for the vignette, which the author must fill-in or associate with specific features of the map, force-lay down, unit command structure, etc. Where links are involved, the author may develop them through a simple point-and-click mechanism. For example, if the unit concept has a link to a controlled asset concept, the author could simply click on the symbol that is an instance of a unit in this vignette (e.g., B/1-55), then click on a different symbol for the attack helicopter (e.g., AH-64). The selection of the 'controlled asset' link, and the domain knowledge base would then know that there is a unit B/1-55 that controls AH-64. This piece of domain knowledge, once loaded into ATEC at the start of a session, would then permit the domain reasoning subsystem of ATEC to apply knowledge about controlled units in, for example, assessing whether the trainee was using all available assets in his reasoning process.

Such vignette authoring tools could be implemented in C++ with a Java interface which would allow them to be used as stand-alone tools, or over the web. The vignette data/knowledge base could be implemented in a commercial grade web-compatible database, such as Microsoft Access.

Summary and Conclusions

Developing an intelligent tutoring system for interactive training of thinking skills, such as battlefield command reasoning, is a difficult problem. It does not fit the mold of previously successful ITS because of the open-ended nature of the interactions and the dialog-based nature of the tutorial process involved. An interactive practice environment using instructional agent technology has been developed, an approach that the present authors have used successfully in previous research. It was determined that concept of an action-based interactive practice environment would not meet the requirements for an interactive training of thinking skills. However, incorporating a dialog management capability into the ATEC concept would meet this goal. This dialog management appears to be technically feasible, although pushing the state-of-the-art. A revised architecture and operational concept incorporating natural language processing and tutorial dialog was, therefore, developed.

A Phase I prototype system was developed that demonstrates the revised architecture. However, this prototype used a very simple keyword-spotting algorithm as a placeholder for a dialog management system. This placeholder needs to be replaced with a full dialog management system that includes: (a) linguistic analysis capabilities, (b) dialog turn generation (e.g., a clarification question needs an answer, an assertion needs an evaluative response), and (c) tutorial strategy (e.g., when and how to provide hints, prompts, change of topic, ordering of topics).

The student model will require several different levels of evaluative feedback. The Phase II effort should develop a model with a more robust dialog management capability and more extensive domain knowledge component. This extension will make the instructor responses to trainee input more evaluative and informative. Furthermore, the proposed student model structure should provide a direct means for a summary performance assessment based on the depth of questioning needed in each aspect of each of the eight themes, which is not the case for the Phase I prototype system.

A detailed design and implementation plan for the revised ATEC architecture has also been developed. A design and development plan has been developed for each of the four subsystems of the ATEC system: (a) the user interface, (b) a dialog management subsystem, (c) an instructional agent and (d) semi-automated vignette authoring tools. The proposed ATEC system appears feasible in a Phase II effort, although some aspects of the system are pushing the state-of-the-art and should be considered research efforts. In particular, development of a dialog management system that is able to deal with battlefield command reasoning is essentially a research effort. The use of components of a proven tutorial dialog system, AutoTutor, reduces the risk; however, two key aspects of domain discourse not addressed in AutoTutor must be addressed in ATEC: adapting the form of tutoring to application of abstract principles rather than specific facts and concepts, and adding dialog to deal with spatial referents.

The main research issue in a Phase II effort deals with the usability and training value of the ATEC system. This research should focus on using a partial or complete system prototype with potential trainees and evaluating it first for pedagogical effectiveness and conversational appropriateness, and later for training effectiveness. A Phase II development effort should include iterative cycles of evaluation and refinement of the dialog management portion of the system for pedagogical effectiveness and conversational appropriateness. A training effectiveness study should be conducted when the system is to be introduced into actual use.

Successful completion of the Phase II effort should help meet the Army requirement for more efficient and effective methods for instilling battle command thinking skills. Successful Phase III commercialization should extend similar benefits to a wide range of private and government sector training audiences. Moreover, the research entailed in Phase II development should increase our understanding of basic and applied research issues related to the training and assessment of conceptual reasoning skills.

Leave blank for page numbers

References

Anderson, J. S., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science, 228*, 456-468.

Anderson, J. R., Corbett, A. T., Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences, 4(2),* 167–207.

Clark, H. H. (1996). *Using language*. Cambridge University Press.

Eisenberg, E. M. (1990). Jamming: Transcendence through organizing. *Communication Research, 17(2)*, Sage Publications.

Fisk, A. D., & Rogers, W. A. (1992). The application of consistency principles for the assessment of skill development. In J.W. Regian & V. J. Shute (Eds.), *Cognitive approaches to automated instruction*. Hillsdale, NJ: Erlbaum.

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., & TRG (1999). AutoTutor: A simulation of a human tutor. *Journal of Cognitive Systems Research, 1*, 35-51.

Jones, P., Mitchell, C., & Rubin, K. (1988). *Intent inferencing with a model-based operator's associate* (Rep. No. 88-2). Atlanta, GA: Georgia Institute of Technology, School of Industrial and Systems Engineering.

Jurafsky, D., & Martin, J. E. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, NJ: Prentice Hall.

Klein, G. (1989). Recognition-primed decisions. In W. Rouse (Ed.), *Advances in man-machine systems research, 5*, 47-92. Greenwich, CT: JAI Press.

Landauer, T.,& Dumais, S. (1997). An answer to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 2111-240.

Lesgold, A., Lajoie, S., Bunzo, M., & Eggan, G. (1992). Sherlock: A coached practice environment for an electronics troubleshooting job. In J. Larkin & R. Chabay (Eds.), *Computer assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches*, Hillsdale, N.J.: Erlbaum.

Moore, J. D. (1995). *Participating in explanatory dialogues*. Cambridge, MA: MIT Press.

Person, N. K., Graesser, A. C., Harter, D., & the Tutoring Research Group (2000, July). The Dialog Advancer Network: A mechanism for improving AutoTutor's conversational skills. Presented at the 10th Annual Meeting of the Society for Text and Discourse, Lyon, France.

Person, N. K., Graesser, A. C., Kreuz, R. J., Pomeroy, V., & the TRG (in preparation). Simulating human tutor dialog moves in AutoTutor. *International Journal of Artificial Intelligence in Education.*

Polson, M., & Richardson, J. (Eds.). (1988). *Foundations of intelligent tutoring systems*. Hillsdale, NJ: Erlbaum.

Robertson, S., Zachary, W., & Black, J., (Eds.). (1990). *Cognition, computation and cooperation: An interdisciplinary study of multi-agent systems*. Norwood: Ablex.

Schmidt, C., Sridharan, N., & Goodson, J. (1978). The plan recognition problem: An intersection of artificial intelligence and psychology. *Artificial Intelligence, 11 (1 & 2)*, 45-83.

Schober, M. F. (1993). Spatial perspective-taking in conversation. *Cognition, 47(1),* 1-24.

Schober, M. F. (1998). How addressees affect spatial perspective choice in dialogue. In P. L. Olivier & K. P. Gapp (Eds.), *Representation and processing of spatial expressions* (pp. 231-245). Mahwah, NJ: Erlbaum.

Shute, V. J., & Psotka, J. (1995). Intelligent tutoring systems: Past, present, and future. In D. Janassen (Ed.), *Handbook of research on educational communications and technology*, Scholastic Publications.

Suchman, L. A. (1990). What is human – machine interaction? In S. Robertson, W. Zachary, & J. Black (Eds.), *Cognition, computation and cooperation* (pp. 25-55). Norwood: Ablex.

U.S. Army Research Institute (2001). Think Like A Commander [Computer CD-ROM and Materials]. (Available from the U.S. Army Research Institute for the Behavioral and Social Sciences, 851 McClellan Ave, Fort Leavenworth, KS 66072).

Van Lehn, K. (1996). Cognitive skill acquisition*. Annual Review of Psychology, 47,* 513-539.

Van Lehn, K. (1990). *Mind bugs: the origins of procedural misconceptions*. Cambridge, MA: MIT Press.

Vaubel, K. P., & Gettys, C. F. (1990). Inferring user expertise for adaptive interfaces, *Human-Computer Interaction, 5*, 95-117.

Wenger, E. (1987). *Artificial intelligence and tutoring systems*. Los Altos: Morgan Kaufmann.

Zachary, W. & Ryder, J. (1997). Decision support: Integrating training and aiding. In M. Helander, T. Landauer, & P. Prasad, (Eds.), *Handbook of human computer interaction* (2nd ed., pp. 1235-1258). Amsterdam: North Holland.

Zachary, W. W., Ryder, J. M., Ross, L., & Weiland, M. Z. (1992). Intelligent computer-human interaction in real-time multi-tasking process control and monitoring systems. In M. Helander & M. Nagamachi (Eds.). *Design for manufacturability.* New York: Taylor and Francis.

Zachary, W., Ryder, J., Santarelli, T. & Weiland, M. (2000). Applications for Executable Cognitive Models: A Case Study Approach. In *Proceedings of the Human Factors and Ergonomics Society 44th Annual Meeting*, Santa Monica, CA: Human Factors and Ergonomics Society.

Appendix A

Hierarchical Question List

0: Given the situation as described, what assets are available to help you in this situation?

1: What resources already assigned to the 6-502nd can you call upon for help?

- 1.1: What 6-502nd assets are available to help you?
  - 1.1.1: How can the company on the left flank provide help?
    - 1.1.1.1: Did you consider having the company on the left flank suppress the enemy OPs?
    - 1.1.1.2: Did you consider having the company on the left flank continue their assigned mission to secure the bridge?
  - 1.1.2: What support might the company on the left flank need to continue their mission?
    - 1.1.2.1: Did you consider calling in artillery from 4-41 FA to help?
    - 1.1.2.2: Did you consider calling in aviation from B/1-55 to help?
  - 1.1.3: What can the survivors of the companies that have been hit do?
    - 1.1.3.1 : Did you consider having the companies under fire help themselves?
- 1.2: What Brigade assets are available to help you?
  - 1.2.1: How can you use the 2 OH-58D helicopters from BDE?
    - 1.2.1.1: Did you consider using the BDE OH-58D helicopters locate and suppress the enemy OPs?
  - 1.2.2: How can you use brigade direct support artillery?
    - 1.2.2.1: Did you consider using brigade direct fire artillery to obscure enemy OPs with smoke/HE?
- 1.3: What Division assets are available to help you?

1.3.1: How can you use Division MRLS?

1.3.1.1: Did you consider having Division MRLS suppress enemy MLRS?

1.3.1.2: Did you consider having Division MRLS suppress enemy 152s?

1.3.2: Where are the 152s located?

1.3.2.1: Did you consider how the RFA in the Atchison area will affect Division MRLS support?

2. What resources could you ask to have re-tasked to support you?

2.1: How can the Scout weapons team help you?

2.1.1: Did you consider having the scout weapons team locate and suppress the enemy OPs?

2.2: What Corps assets are available to help you?

2.2.1: Corps has given BDE OPCON of AH-64 helicopters from B/1-55, did you consider using them?

2.3: How can you extract the wounded?

2.3.1: What do you need to use helicopter MEDEVAC?

2.3.2: What do you need to use ground MEDEVAC?

3. What are the options for taking Objective Meade?

3.1: What does the 2nd BDE decision support matrix list as the plan in the event that the 502nd cannot complete their mission?

3.1.1: Did you consider consulting the 2nd brigade decision support matrix to assist you in generating options for taking Objective Meade?

3.1.2: Did you consider having the 4-80 take Objectives Meade and Hill?

3.2: How would you reinforce the 502nd to take Objective Meade?

3.2.1: Did you consider reinforcing the 502nd with resources from 4-80, 4-81 or BDE reserves?

3.3: Can TF 4-80 take Objective Meade?

3.3.1: Did you consider having TF 4-80 take Objective Meade?

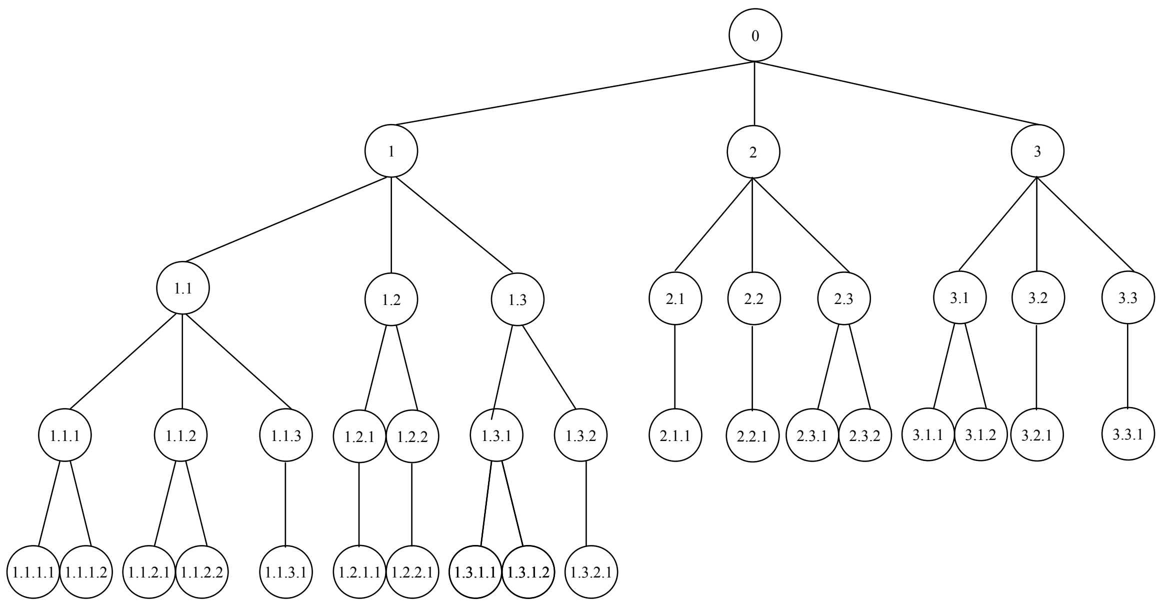

Figure A-1. Question Tree

Appendix B

Answer Concepts for Each Question

0: Given the situation as described, what assets are available to help you in this situation?
FORCE: Company LOCATION: left flank
FORCE: Company ASSET: survivors
FORCE: Brigade ASSET: helo
FORCE: Brigade ASSET: artillery
FORCE: Division ASSET: MRLS
ASSET: scouts
FORCE: Corps ASSET: AH-64
ASSET: MEDEVAC
OBJECT: DSM FORCE: 4-80 OBJECT: Meade
FORCE: 6-502 ACTION: continue

1: What resources already assigned to the 6-502nd can you call upon for help?
FORCE: Company LOCATION: left flank
FORCE: Company ASSET: survivors if T go to 1.1.3
FORCE: Brigade ASSET: helo
FORCE: Brigade ASSET: artillery
FORCE: Division ASSET: MRLS

1.1: What 6-502nd assets are available to help you?
FORCE: Company LOCATION: left flank
FORCE: Company ASSET: survivors

1.1a: What other 6-502nd assets are available to help you?
FORCE: Company ASSET: survivors

1.1b: What other 6-502nd assets are available to help you?
FORCE: Company LOCATION: left flank

1.1.1: How can the company on the left flank provide help?
ACTION: suppress FORCE: enemy
ACTION: suppress OBJECT: OPs
ACTION: continue OBJECT: mission

1.1.1.1: Did you consider having the company on the left flank suppress the enemy OPs?
QA: Yes No

1.1.1.2: Did you consider having the company on the left flank continue their assigned mission to secure the bridge?
QA: Yes No

1.1.2: What support might the company on the left flank need to continue their mission?

FORCE: brigade ASSET: artillery
FORCE: brigade ASSET: aviation

1.1.2.1: Did you consider calling in artillery from 4-41 FA to help you?
QA: Yes No

1.1.2.2: Did you consider calling in aviation from B/1-55 to help you?
QA: Yes No

1.1.3: What can the survivors of the companies that have been hit do?
ACTION: nothing
ACTION: help OBJECT: themselves
ACTION: help OBJECT: wounded

1.1.3.1 : Did you consider having the companies under fire help themselves?
QA: Yes No

1.2: What Brigade assets are available to help you?
FORCE: Brigade ASSET: helo
FORCE: Brigade ASSET: artillery

1.2a What other Brigade assets are available to help you?
FORCE: Brigade ASSET: artillery

1.2b What other Brigade assets are available to help you?
FORCE: Brigade ASSET: helo

1.2.1: How can you use the 2 OH-58D helicopters from BDE?
ACTION: locate OBJECT: OPs
ACTION: suppress OBJECT: OPs

1.2.1.1: Did you consider using the BDE OH-58D helicopters locate and suppress the enemy OPs?
QA: Yes No

1.2.2: How can you use brigade direct support artillery?
ASSET: arty TASK: obscure OBJECT: OPs OBJECT: smoke

1.2.2.1: Did you consider using brigade direct fire artillery obscure enemy OPs with smoke/HE?
QA: Yes No

1.3: What Division assets are available to help you?
FORCE: Division ASSET: MRLS

1.3.1: How can you use Division MRLS?
ASSET: MRLS ACTION: suppress FORCE: enemy

ASSET: MRLS ACTION: suppress ASSET: MLRS
ASSET: MRLS ACTION: suppress ASSET: 152

1.3.1.1: Did you consider having Division MRLS suppress enemy MLRS?
QA: Yes No

1.3.1.2: Did you consider having Division MRLS suppress enemy 152s?
QA: Yes No

1.3.2: Where are the 152s located?
ASSET: 152 LOCATION: RFA
ASSET: 152 LOCATION Atchison

1.3.2.1: Did you consider how the RFA in the Atchison area will affect Division MRLS support?
QA: Yes No

2. What resources could you ask to have re-tasked to support you?
ASSET: scouts
FORCE: Corps ASSET: AH-64
ASSET: MEDEVAC

2.1: How can the Scout weapons team help you?
ACTION: locate ACTION: suppress FORCE: enemy OBJECT: OPs

2.1.1: Did you consider having the scout weapons team locate and suppress the enemy OPs?
QA: Yes No

2.2: What Corps assets are available to help you?
ASSET: AH-64

2.2.1: Corps has given BDE OPCON of AH-64 helicopters from B/1-55, did you consider using them?
QA: Yes No

2.3: How can you extract the wounded?
ASSET: airmed ASSET: groundmed

2.3.1: What do you need to use helicopter MEDEVAC?
OBJECT: PZ

2.3.1.1: Did you consider whether there were secure PZs for the helicopter MEDEVAC?
QA: Yes No

2.3.2: What do you need to use ground MEDEVAC?
OBJECT: pickup OBJECT: mines

2.3.2.1: Did you consider whether there was a secure pickup area and if the roads were clear of mines to use ground MEDEVAC?
QA: Yes No

3. What are the options for taking Objective Meade?
OBJECT: DSM FORCE: 4-80 OBJECT: Meade OBJECT: Hill
FORCE: 6-502 ACTION: continue

3.1: What does the 2nd BDE decision support matrix list as the plan in the event that the 502nd cannot complete their mission?
OBJECT: DSM FORCE: 4-80 OBJECT: Meade OBJECT: Hill

3.1.1: Did you consider consulting the 2nd brigade decision support matrix to assist you in generating options for taking Objective Meade?
QA: Yes No

3.1.2: Did you consider having the 4-80 take Objectives Meade and Hill?
QA: Yes No

3.2: How would you reinforce the 502nd to take Objective Meade?
FORCE: 4-80 FORCE: 4-81 FORCE: BDEres

3.2.1: Did you consider reinforcing the 502nd with resources from 4-80, 4-81 or BDE reserves?
QA: Yes No

3.3: Can TF 4-80 take Objective Meade?
QA: YES

.3.1: Did you consider having TF 4-80 take Objective Meade?
QA: Yes No

Appendix C

Synonyms

Company
- Co
- Units
- 6-502
- 6502
- 6502^{nd}

Brigade
- BDE

Division
- DIV

Corps

Enemy
- Dakotan
- Dakotans

4-80
- 480

4-81
- 481

BDEres
- Brigade reserves
- BDE reserves

Survivors
- My people

Helo
- OH-58
- OH58
- OH-58s
- OH58s
- OH58D
- OH-58Ds
- OH58Ds
- OH-58D
- Scout helicopters

Artillery
- Arty

MRLS
- MRLS
- Multiple Launch rocket
- Multiple rocket launch
- MLRS

Aviation

152

scouts

AH-64
- AH64

MEDEVAC
- Evacuate casualties
- Evacuate wounded
- Evacuate injured

Airmed
- Helicopter medevac

Groundmed
- Truck medevac

Left flank
- West

RFA
- Restricted fire area
- Restricted fires area

Atchison
- Atcheson
- Achison
- Acheson

Themselves

Wounded
- Injured
- casualties

OPs
- Observation points
- Observation posts
- Observers
- Outposts
- Observation point
- Observation post
- Observer
- Outpost

Smoke
- HE
- High explosives

PZ
- Pickup zone
- Landing zone

Pickup
- Pickup area
- Secure pickup
- Secure area

Mines
- No mines
- Clear roads
- Clear routes

DSM
- Decision support matrix
- Decision matrix
- Support matrix

Meade

Hill

Suppress
- Fire upon

Continue
- Proceed
- Continue
- Proceed
- Stick to
- Follow

Nothing
- Not much
- Ineffective

Help
- Assist

Locate
- find
- detect

Yes
- True
- T
- Y

No
- False
- F
- N