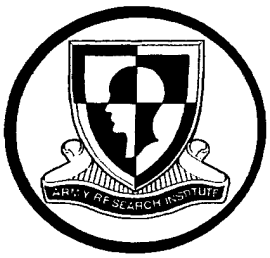
Technical Report 1143

Phase II Final Report on an Intelligent Tutoring System for Teaching Battlefield Command Reasoning Skills

Eric A. Domeshek
Stottler Henke Associates, Inc.

March 2004



United States Army Research Institute for the Behavioral and Social Sciences

Approved for public release; distribution is unlimited

20040419 056

U.S. Army Research Institute
for the Behavioral and Social Sciences

A Directorate of the U.S. Army Human Resources Command

ZITA M. SIMUTIS
Director

Technical Review by

Bruce W. Knerr, U.S. Army Research Institute
William R. Howse, U.S. Army Research Institute

NOTICES

DISTRIBUTION: Primary distribution of this Technical Report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, Attn: DAPE-ARI-PO, 5001 Eisenhower Ave., Alexandria, VA 22304-4841.

FINAL DISPOSITION: This Technical Report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this Technical Report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

1. REPORT DATE (dd-mm-yy)	2. REPORT TYPE	3. DATES COVERED (from. . . to)
March 2004	Final	October 2001 to November 2003

4. TITLE AND SUBTITLE

Phase II Final Report on an Intelligent Tutoring System for Teaching Battlefield Command Reasoning Skills

5a. CONTRACT OR GRANT NUMBER
DASW01-01-C-0039

5b. PROGRAM ELEMENT NUMBER
0602785A

6. AUTHOR(S)

Eric A. Domeshek (Stottler Henke Associates, Inc.)

5c. PROJECT NUMBER
A790

5d. TASK NUMBER
211

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Stottler Henke Associates, Inc.
951 Mariner's Island Blvd. Suite 360
San Mateo, CA 94404

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: DAPE-ARI-IK
5001 Eisenhower Avenue
Alexandria, VA 22304-4841

10. MONITOR ACRONYM
ARI

11. MONITOR REPORT NUMBER
Technical Report 1143

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Report developed under a Small Business Innovation Research Program, Phase II, 2000.2 contract for topic OSD00-CR02.

14. ABSTRACT *(Maximum 200 words)*:

This report documents Phase II efforts to develop a Socratic Intelligent Tutoring System (ITS) for Teaching Battlefield Command Reasoning Skills. The ultimate goal of this research program is to develop new ITS techniques and technology for teaching skills that cannot be taught as simple methods and procedures to be followed. Achieving expert levels of proficiency in professional-level reasoning skills—whether for battlefield commanders or for professionals in a wide range of other fields—requires extensive practice, coaching, and feedback. Students must be given a chance to drill on detailed and situation-specific knowledge, as well as high-level thinking habits and skills applicable across diverse situations. We studied exemplary coaching by observing tutoring sessions centered on Tactical Decision Games. We analyzed those sessions to identify behaviors that could be enacted and controlled by an automated Socratic tutor. We explored techniques and ultimately built an operational prototype exhibiting many of the key behaviors identified. We also built extensive tools to support authoring of the prototype's knowledge and behavior. This work has produced extensive data, analysis, and implementation, advancing our understanding of, and ability to model, professional-level Socratic tutoring.

15. SUBJECT TERMS
Intelligent Tutoring Systems, Socratic Dialog, Battlefield Command and Control, Simulation-based Training, Artificial Intelligence, SBIR Report

SECURITY CLASSIFICATION OF			19. LIMITATION OF ABSTRACT	20. NUMBER OF PAGES	21. RESPONSIBLE PERSON (Name and Telephone Number)
16. REPORT	17. ABSTRACT	18. THIS PAGE			
Unclassified	Unclassified	Unclassified	Unlimited	186	Dr. Carl Lickteig DSN 464-6928

Best Available Copy

THIS DOCUMENT CONTAINED
BLANK PAGES THAT HAVE
BEEN DELETED

Technical Report 1143

Phase II Final Report on an Intelligent Tutoring System for Teaching Battlefield Command Reasoning Skills

Eric A. Domeshek
Stottler Henke Associates, 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 22304-4841

March 2004

2O262785A790

Personnel Performance and Training Technology

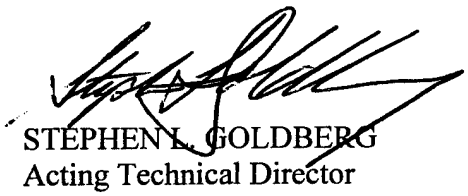
Approved for public release; distribution is unlimited

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 is through an intelligent tutoring system (ITS) for training battlefield command reasoning skills.

This Phase II Small Business Innovation Research Program effort targeted the design and limited demonstration of a Socratic 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. 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 Training of Future Brigade Combat Team Commanders and Staffs. Results of this effort were briefed to the Director of Bio-Systems, Office of the Director, Defense Research and Engineering and the Deputy Director, Directorate of Training, Doctrine, and Combat Development, U.S. Army Armor Center.



STEPHEN L. GOLDBERG
Acting Technical Director

ACKNOWLEDGEMENTS

This report reflects work performed under a Small Business Innovation Research Program 99.2 contract for topic OSD00-CR02. The project aimed at design of a Socratic Intelligent Tutoring System (ITS) for high-level battlefield command reasoning skills. What is reported here reflects the efforts of a large team.

I would like to thank our contract monitors at the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) Fort Knox Field Unit. Dr. James Lussier had the vision that led to this line of research. Dr. Carl Lickteig was immensely helpful in managing all of the details of the project. Dr. Ted Shlechter was also quite supportive of our work.

Our primary subject matter experts, Maj. John Schmitt, USMC (Ret.) and BGen Keith Holcomb, USMC (Ret.) were both central to the development of this project. We quite literally could not have done it without them. Their tactical expertise, combined with their enthusiasm for and experience with the Tactical Decision Games (TDG) format (including development of the electronic distributed eTDG format) was invaluable. Equally critical was their ability to recruit active-duty Marine Captains to serve as subjects of tutoring observations. It is a tribute to their reputations and recognized skills that they were easily able to provide as many volunteers as we could use. I would also like to thank the many Marine and Army officers who volunteered as subjects for the Phase II observations and other system evaluation interactions. In addition, Maj. Andrew Harris, USAR was also helpful in providing additional information and an Army armor perspective.

Our colleagues at Klein Associates, Inc., Karol Ross, Deborah Battaglia, Rob Hutton, Danyelle Harris, Stacey Green, and Sterling Wiggins provided significant support in designing and carrying out the research reported here. They contributed most heavily in the instructional observations and analysis, helping to develop the format and schedule, serving as observers of the actual sessions, preparing session transcripts, and providing analysis of the results. Their analysis of instructional strategies was a notable part of the project.

Special thanks to Dick Stottler of Stottler Henke Associates, Inc. for conceiving this project, setting it up, and providing direction. Thanks to Jim Ong and Susann LuperFoy who provided critical supervision and insight. And last, but certainly not least, I gratefully acknowledge the efforts of Eli Holman and David Struck who implemented the Phase II prototype system that is described in this report.

EXECUTIVE SUMMARY

Research Requirement:

Under extreme stress, time pressure, and uncertainty, commanders must assess complex, ambiguous tactical situations, develop and prioritize goals, and pursue these goals by issuing orders, reports, and requests that consider the situation, own assets and capabilities, and the assets, capabilities, and intentions of the enemy. These reasoning skills cannot be taught simply as methods and procedures to be followed. Reaching expert levels of proficiency requires extensive practice including coaching and feedback from instructors to identify strengths and weaknesses in performance, encourage reflection, stimulate the student's thought processes, and enhance the student's reasoning capabilities.

Coaching and feedback are typically provided by expert instructors during dialogs with students. An inherent limitation of coached instruction is that the expert coaches are an expensive limited resource that is infrequently available; when available their attention must often be shared among too many students for truly effective personalized interaction. Intelligent Tutoring Systems (ITSs) are computer-based training systems that mimic human instructors to provide automated, one-on-one instruction. Intelligent Tutoring Systems encode subject matter and instructional expertise to assess each student's performance, knowledge, and skills to provide individualized learning experiences, adapted to each student's skills, background, and learning preferences.

However, no ITS system developed to date supports a domain nearly as complex as battlefield command which requires sophisticated reasoning about a large number of goals, potential plans, variables, and constraints. Evaluation of each student's solution is extremely challenging because there is no single "correct" solution to each scenario. Thus, developing an ITS for teaching battlefield command reasoning skills will require "pushing the envelope" in ITS technology along dimensions such as: (1) representing complex problem solutions and rationale, (2) assessing student's reasoning skills against alternate expert-provided solutions and rationales, and (3) achieving student-tutor interactions that mimic expert instructor's techniques for enhancing high-level thinking habits.

Procedure:

During Phase II, we called out eight objectives continuing the analysis, design, and implementation tasks from Phase I, with additional emphasis on evaluation and transition. We continued to identify and encode tactical and other knowledge needed to support scenarios, student interaction, and evaluation of student efforts. We also explored techniques for accomplishing that interaction and evaluating the student's side of the dialog, including analysis of tutoring strategies observed in live tutoring sessions and new algorithms to embed such strategies in an automated system. Our implementation work was aimed at both authoring and runtime support—striving to produce a fully usable system. Evaluation and transition objectives

included demonstrating the effectiveness of tutoring techniques as embedded in our runtime system, accumulating guidelines and baseline data on authoring, and developing a preliminary analysis of issues bearing on long-term acceptance, effectiveness, and maintainability of an expanded, fully operational ComMentor.

Sixteen core activities were called out for the Phase II effort, falling into four clusters: (1) up-front data collection and analysis, including review of the Phase I results, Phase II observations of live tutoring sessions, and analysis of resulting data on expert tutor behavior; (2) ongoing modeling of instructional strategies, domain knowledge, curricular knowledge, and scenario knowledge; (3) iterative design and implementation of ComMentor's runtime and authoring tools; and (4) parallel observation, evaluation, analysis, and documentation of the evolving system.

Findings:

Our major Phase II accomplishments fall into four categories. First, we gathered and analyzed extensive data from live tutorial sessions in which expert tactical instructors worked with mid-level military officers (see "Results" sub-section "TDG Data and Preliminary Analysis"). Second, we developed initial taxonomies of instructional strategies, battlefield command skills, and domain concepts (see "Results" sub-section "Instructional Model"). Third, we built an end-to-end software system for authoring and running training scenarios with Socratic tutoring multimodal interaction (see "Results" sub-sections "Final Prototype Architecture and Design," "Final Prototype Runtime Capabilities," and "Final Prototype Authoring Capabilities"). And finally, we have explored the promise, possibilities, and limitations of a battlefield command reasoning ITS given the current state of technology (see "Conclusions" sub-section "Prospects for Battlefield Command Reasoning ITSs").

Utilization of Findings:

In this Phase II project, we deepened our understanding of how expert human tutors guide junior commanders through training scenarios in order to drill and improve their battlefield command reasoning skills, and how an automated tutor might mimic aspects of this behavior. This project leaves a rich legacy of data that should prove useful in future studies of professional-level training. The prototype ComMentor Socratic ITS, with its authoring tool suite, has established a baseline of capability and clarified challenges for future research and development. Significant pieces of the resulting code are already being reused by Stottler Henke in a variety of ITS applications and general knowledge-based system projects. The infrastructure developed is being applied to a range of professional-level training problems in the U.S. military and elsewhere. An ongoing U.S. Navy project will leverage much of ComMentor's infrastructure, and another recently launched project for the U.S. Air Force seems likely to adopt the same tools. A project to apply ComMentor's technology to problems in medical training is expected to start soon. Stottler Henke is also actively exploring the possibility of integrating the ComMentor tools with commercially available SimBionic agent authoring and runtime tools. Such research is needed to pursue refinements necessary to field a practical and effective version of ComMentor for training U.S. Army armor officers.

PHASE II FINAL REPORT ON AN INTELLIGENT TUTORING SYSTEM FOR TEACHING BATTLEFIELD COMMAND REASONING SKILLS

CONTENTS

Page

INTRODUCTION 1

The Problem of Training Battlefield Command Reasoning Skills 1
How ComMentor Trains Battlefield Command Reasoning Skills 2
Benefits of the ComMentor Approach 7
Innovations 7

BACKGROUND 8

Scenario-Based Instruction 8
Socratic Tutoring Systems 9
Multimodal Input Processing 10
Case-Based Situation Interpretation/Diagnosis 11
Application-Specific Authoring Tools 11

METHOD 12

Project Objectives 12
Project Tasks 13

RESULTS 16

TDG Data and Preliminary Analysis[2] 16
Instructional Model 24
Domain Model 35
Curriculum Model 36
Scenario Model 38
Final Prototype Architecture and Design 46
Final Prototype Runtime Capabilities 55
Final Prototype Authoring Capabilities 72

FUTURE RESEARCH AND REFINEMENT 88

Runtime Effectiveness and Evaluation 89
Cost/Benefit Studies of Authoring/Tutoring 90
Authoring Tool Refinements 91
Strengthening and Generalization of Domain and Tutoring Knowledge 92
Applications for Project Infrastructure 92

CONTENTS (Continued)

Page

Transition of ComMentor to Army Applications 93
Other Possible Markets for High-Level Socratic Tutors 93

CONCLUSIONS 94

Project Goals vs. Project Accomplishments 94
Student Input: The Problem of Natural Language 97
Prospects for Battlefield Command Reasoning ITSs 100

REFERENCES 103

APPENDIX A. Project Chronology A-1

B. Detailed List of Potential System Refinements B-1

C. Sample Session Transcript C-1

LIST OF TABLES

Table 1. Scenarios for Phase II Tutorial Observations 17
Table 2. Phase II Project Objectives and Accomplishments 95

LIST OF FIGURES

Figure 1. ComMentor Phase II Abstract System Architecture Design. 5
Figure 2. Map for the "Enemy Over the Bridge Scenario" 18
Figure 3. Narrative Introduction to the "Enemy Over the Bridge" Scenario 19
Figure 4. Top-Level Categories in ComMentor's Domain Ontology 36
Figure 5. Two Top-Level Breakdowns of ComMentor's Curriculum 37
Figure 6. Evaluation Tree for "Enemy Over the Bridge" (EOTB) Scenario. 41
Figure 7. Detailed Functional Architecture Diagram for Final Phase II ComMentor Prototype 47
Figure 8. Implementation Architectural Layers Diagram for the GRIST Package 49
Figure 9. Implementation Architectural Layers Diagram for the MILL Package 51
Figure 10. Implementation Architectural Layers Diagram for the GRAIN Package. 54
Figure 11. ComMentor Login Screens 58
Figure 12. Slide from ComMentor Scenario Briefing at Start of "Enemy Over the Bridge" 58
Figure 13. Main ComMentor Interaction Screen at Start of "Enemy Over the Bridge" 59
Figure 14. Main ComMentor Screen After Dragging Alpha Company Up the Road 61
Figure 15. Flowchart of Processing for a Dialog Node 62
Figure 16. EOTB Dialog Tree for Eval_RedPicture 64
Figure 17. EOTB Dialog Node Dial_RedPicture 64

Page

Figure 18. EOTB Dialog Node Dial_RedBattalion 65
Figure 19. EOTB Dialog Node Dial_RedOffensive 65
Figure 20. EOTB Dialog Node Dial_RedBridgehead 66
Figure 21. EOTB Dialog Node Dial_RedBattalion_MechCoInAA 68
Figure 22. EOTB Dialog Node Dial_RedBattalion_10T62sSeen 68
Figure 23. EOTB Dialog Node Dial_RedBattalion_FlowingFor2Hrs 69
Figure 24. ComMentor's Authoring Tool Launcher Window 76
Figure 25. ComMentor's Ontology Authoring Tool Window 78
Figure 26. ComMentor's Presentation Authoring Tool Window. 78
Figure 27. ComMentor's Situation Authoring Tool Window 80
Figure 28. ComMentor's Situation Authoring Tool's Force-Structure Editing Dialog. 80
Figure 29. ComMentor's Template Authoring Tool Window 81
Figure 30. ComMentor's Curriculum Authoring Tool Window 82
Figure 31. ComMentor's Tutoring Authoring Tool Window Focused on a Dialog Node. 83
Figure 32. ComMentor's Tutoring Authoring Tool Window Focused on an Evaluation Node ..84
Figure 33. ComMentor's Initial Loading Progress Bar C-1
Figure 34. Splash-Screen with Log-In and New-Account Options C-1
Figure 35. New Student Account Creation Screen C-1
Figure 36. Start of Briefing for "Enemy Over the Bridge" Scenario C-2
Figure 37. Second Briefing Slide for "Enemy Over the Bridge" C-3
Figure 38. Third Briefing Slide for "Enemy Over the Bridge" C-3
Figure 39. Fourth Briefing Slide for "Enemy Over the Bridge" C-4
Figure 40. Fifth Briefing Slide for "Enemy Over the Bridge" C-4
Figure 41. Sixth Briefing Slide for "Enemy Over the Bridge" C-5
Figure 42. Seventh Briefing Slide for "Enemy Over the Bridge" C-5
Figure 43. Eighth Briefing Slide for "Enemy Over the Bridge" C-6
Figure 44. Ninth Briefing Slide for "Enemy Over the Bridge" C-6
Figure 45. Tenth Briefing Slide for "Enemy Over the Bridge". C-7
Figure 46. Eleventh Briefing Slide for "Enemy Over the Bridge" C-7
Figure 47. Main Interaction Screen with Initial Order-Elicitation Prompt C-8
Figure 48. Initial Orders: Filling in Alpha Company's Tasking C-9
Figure 49. Initial Orders: Translation and Response to Alpha Company's Tasking C-10
Figure 50. Initial Orders: Layout After Orders Issued to A-D Companies C-11
Figure 51. The Red Situation: The Tutor's 1st Question C-12
Figure 52. The Red Situation: The Student's 1st Answer C-13
Figure 53. The Red Situation: Student's Answer to Follow-Up on 1st Question C-14
Figure 54. The Red Situation: The Tutor's 2nd Question C-15
Figure 55. The Red Situation: The Student's 2nd Answer C-16
Figure 56. The Red Situation: The Student's 3rd Answer. C-17
Figure 57. The Red Situation: The Student's Answer to the 3rd Question's 1st Follow-Up C-18
Figure 58. The Red Situation: The Student's Answer to the 3rd Question's 2nd Follow-Up C-19
Figure 59. The Red Situation: The Student's Answer to the 4th Question C-20

Page

Figure 60. The Red Situation: The Student's Answer to the 4th Question's 1st Follow-Up. C-21
Figure 61. The Red Situation: The Student's Answer to the 5th Question.....C-22
Figure 62. The Red Situation: The Student's Answer to the 6th Question.....C-23
Figure 63. The Red Situation: The Student's Answer to the 7th Question.....C-24
Figure 64. The Red Situation: Tutor Highlights Red Bridgehead Area. C-25
Figure 65. Attack on AA: The Student's Answer to Main Question.....C-26
Figure 66. Attack on AA: The Tutor's 1st Drill-Down Question.....C-27
Figure 67. Attack on AA: The Student Response to 1st Drill-Down Question Follow-Up. C-28
Figure 68. Attack on AA: The Student Responding to 2nd Drill-Down Question. C-29
Figure 69. Attack on AA: The Tutor's Query to Close out Drill-Down on Main Question.....C-30
Figure 70. Scouts Tasking: The Student Responding to 1st Question.....C-31
Figure 71. Scouts Tasking: The Student Responding to 2nd Question.....C-32
Figure 72. Scouts Tasking: The Student Responding to 2nd Question Follow-Up. C-33
Figure 73. Scouts Tasking: The Student's Response to 3rd Question. C-34
Figure 74. Scouts Tasking: The Student Moves the Scouts.....C-35
Figure 75. Scouts Moving to South: The Student's Answer to the 1st Question. C-36
Figure 76. Scouts Moving to South: The Student's Answer to the 2nd Question. C-37
Figure 77. Scouts Moving to South: The Student Responds to the 2nd Question Follow-Up. ..C-38
Figure 78. Ignoring the Bridge: The Student's Answer to the 1st Question. C-39
Figure 79. Ignoring the Bridge: The Student Responds to the 2nd Question. C-40
Figure 80. Ignoring the Bridge: The Student's Answer to the 3rd Question. C-41
Figure 81. Ignoring the Bridge: The Student's Response to 3rd Question Follow-Up. C-42
Figure 82. Ignoring the Bridge: The Student's Fourth Answer. C-43
Figure 83. Ignoring the Bridge No Longer: A New Tasking.....C-44
Figure 84. To The Bridge: Student's Answer to 1st Question.....C-45
Figure 85. To The Bridge: Student's Answer to 2nd Question.....C-46
Figure 86. To The Bridge: Student's Response to 3rd Question. C-47
Figure 87. Envelopment: Student's Response to 1st Overview Question. C-48
Figure 88. Envelopment: Student's Response to final Definitional Drill-Down Question. C-49
Figure 89. Envelopment: Student's Response to 1st Overview Question. C-50
Figure 90. Envelopment: Student's Proposed Pivot Point.....C-51
Figure 91. Reserve: What Constitutes Success.....C-52
Figure 92. Reserve: Reserve Forces End-State to Reinforce Success. C-53
Figure 93. Reserve: Choice of Reserve Forces to Reinforce Success. C-54
Figure 94. Reserve: Where an Enemy Might Go.....C-55
Figure 95. Reserve: Where a Reserve Might Watch for an Enemy Move to the North. C-56
Figure 96. Reserve: Where Else a Reserve Might Help Out. C-57
Figure 97. Reserve: The Student's Final Choice of a Location for the Reserve. C-58
Figure 98. Finale: The Reflective Questionnaire.....C-59

PHASE II FINAL REPORT ON AN INTELLIGENT TUTORING SYSTEM FOR TEACHING BATTLEFIELD COMMAND REASONING SKILLS

Introduction

This Introduction has two major purposes: first to lay out the problem being addressed in this project, and second to characterize the ComMentor solution. Given the initial discussion of our solution we then briefly review why we believe ComMentor is valuable and what is innovative about our approach.

The Problem of Training Battlefield Command Reasoning Skills

Under extreme stress, time pressure, and uncertainty, commanders must assess complex, ambiguous tactical situations, develop and prioritize goals, and pursue these goals by issuing orders, reports, and requests that consider the situation, own assets and capabilities, and the assets, capabilities, and intentions of the enemy. High performance decision-making requires the commander to apply detailed and situation-specific knowledge as well as high-level thinking habits and skills that are applicable across diverse tactical situations. These skills include modeling a thinking enemy, using all available assets, and considering how the commander's fight fits into the bigger picture from friendly and enemy perspectives.

These reasoning skills cannot be taught simply as methods and procedures to be followed. Although guidelines can help commanders achieve baseline levels of performance, achieving an expert level of proficiency requires:

- Extensive practice of command reasoning skills in a variety of situations to acquire and reinforce skills until they can be applied flexibly and automatically. This experience can include a combination of actual combat, live training exercises, computer-based and paper-based simulations, and mental exercises.

- Coaching and feedback from instructors to identify strengths and weaknesses in performance, encourage reflection, stimulate the student's thought processes, and enhance the student's reasoning capabilities.

Coaching and feedback can be provided by expert instructors during dialogs with students. An example of this type of instruction includes Think Like a Commander (TLAC) materials (U.S. Army Research Institute, 2001). These TLAC materials are currently used in officer courses at Fort Leavenworth and Fort Knox. Another example is Tactical Decision Games (TDGs) (e.g., Schmitt, 1994). In both examples, tactical scenarios are presented to students who must then reach conclusions and make decisions. If the student fails to take an applicable theme into account, the instructor asks the student questions, at first indirect, to encourage the student to reflect upon his or her thinking about a theme.

An inherent limitation of coached instruction is that the expert coaches are an expensive limited resource that is infrequently available; when available their attention must often be shared among too many students for truly effective personalized interaction. Classroom

instruction, such as Army officer training, adds the constraint that instruction is only accessible to students who can attend the course on site, requiring significant time and travel costs as well as scheduling difficulties. Automated, computer-based training can provide instruction "any time, any place." However, the types of expert coaching required to teach these command reasoning skills require far more sophisticated reasoning than is possible with current computer-based training (CBT) and web-based training (WBT) technologies that typically assess students based on simple multiple-choice and fill-in-the-blank questions.

Intelligent Tutoring Systems (ITSs) are computer-based training systems that mimic human instructors to provide automated, one-on-one instruction. The ITSs encode subject matter and instructional expertise to assess each student's performance, knowledge, and skills to provide individualized learning experiences, adapted to each student's skills, background, and learning preferences. Most ITSs provide a simulation or problem-solving user interface that enables students to solve problems or pursue goals by entering a sequence of actions or decisions selected from a wide range of possible choices. The ITS applies subject matter expertise to evaluate these actions and decisions to assess the student's knowledge and skills and then select and deliver appropriate instructional interventions.

However, no ITS system developed to date supports a domain nearly as complex as battlefield command which requires sophisticated reasoning about a large number of goals, potential plans, variables, and constraints. Evaluation of each student's solution is extremely challenging because there is no single "correct" solution to each scenario. As Gen. George S. Patton wrote, "There is no approved solution to any tactical situation." Thus, developing an ITS for teaching battlefield command reasoning skills will require "pushing the envelope" in ITS technology along a number of dimensions, such as:

- Representing solutions and their rationale to complex battlefield command problems,
- Assessing the student's reasoning skills by comparing each student's solution with multiple expert solutions and their rationale, and
- Implementing student-coach interactions that enhance high-level thinking habits, based on an assessment of each student's reasoning, as well as knowledge structures that represent each expert's reasoning.

How ComMentor Trains Battlefield Command Reasoning Skills

We proposed to develop a Command Mentoring Intelligent Tutoring System (ComMentor) in which user interaction would center on TDG scenarios. Each session would start with presentation of a particular TDG. Then the system would solicit an initial student solution to the TDG problem. The bulk of the session would be devoted to a Socratic dialog aimed at exploring the student's understanding of the situation and bringing the student to see where their thinking could be improved. Through this process, the system would also refine its model of the student's strengths and weaknesses. A combination of artificial intelligence and intelligent tutoring system technologies were identified that held promise as a basis for building such a system:

- Scenario-Based Instruction – Many ITS technologies based on expert systems approaches, such as model-tracing tutors, are inapplicable when the goal—as with ComMentor—is to build adaptive training for a task that is so difficult that it is beyond the state-of-the-art to build an automated system capable of performing competently on a challenging simulated version of the task. Faced with this difficulty in other military domains, Stottler Henke has had good success building ITSs that base their student assessments on a combination of limited general knowledge and more contextual scenario-specific knowledge. Typically, each scenario is defined with associated pattern-recognizers that evaluate the student's actions and simulated outcomes to estimate the student's mastery (or lack of mastery) of knowledge or skills. These mastery level estimates then provide the basis for selecting instructional interventions (e.g., lessons, feedback, hints, questions, and stories) that prompt the student to reflect on his or her thinking and search for and consider alternatives.

- Application-Specific Scenario Authoring Tools – Any knowledge-based software system is only as powerful and up-to-date as the set of concepts, rules, cases, and other objects in its knowledge base. In large, complex, and continually evolving domains, such as battlefield command, practical development and maintenance of a high-performance knowledge base requires the ability to capture most of the knowledge directly from subject matter experts (SMEs), without programming. Tools to support SME knowledge base construction tend to work better the more customized they are to the demands of a particular application. In ComMentor's Phase II authoring tool suite we have eliminated many occasions for generic knowledge-base (KB) editing, substituting more tailored editing tools and wizards. For example, special graphical tools support creation and manipulation of Curriculum points and their prerequisite relationships, the creation of force-structure models, and the annotation of maps with semantically meaningful regions; wizards help authors get new scenarios initialized and define scenario-specific questions.

- Rationale Capture – When evaluating a student's solution to a professional-level reasoning problem, it is necessary to understand the reasoning behind their decisions. Using scenario-based instruction, it is possible to capture the reasoning (or rationale) of experts for each scenario, ask the student questions that elicit his or her reasoning, and compare this reasoning with those of experts. The rationale underlying expert solutions can also be used to stimulate the student's reflection and reasoning by identifying issues considered by the expert, and then asking the student questions that lead the student to think about those issues as well.

- Multi-Modal, Mixed-Initiative Conversational Interaction – Almost any professional-level form of complex reasoning has evolved some specialized modes of representation to support that reasoning. In military command, maps, force-structure diagrams, and timelines are all used to support situation assessment and decision-making. The Phase I ComMentor prototype illustrated the interaction of graphical and textual input and output modalities. Our Phase II prototype extends that initial demonstration to more robust integration of forms, maps, and force-structure diagrams. We also introduce new capabilities that allow the student to ask questions as well as the tutor, and that allow the tutor to ask follow-up questions that are more responsive to what the student has (or has failed) to say.

From the student's perspective, using ComMentor should be like working through a set of TDGs under the guidance of an expert tactical instructor. Each ComMentor session is devoted to a single TDG, and is expected to take on the order of an hour to complete. This reflects the scenario-based instruction approach described above. Generally, a TDG is introduced through a briefing, presented as a slide show with narration. After the briefing, the student is asked to come up with a response to the presented tactical situation. Based on that response, the system initiates and leads an interactive discussion in which it questions the student about their proposed course of action (and potentially solicits revisions to their plan); with the intent of helping the student think through the situation as an expert tactician might.

To enable this style of interaction, ComMentor has a suite of application-specific authoring tools that support the creation of scenario tutoring scripts. These scripts are built around pattern recognizers called Evaluations that can be authored to look for typical patterns of appropriate or inappropriate action by the student (or for reasonable or unreasonable beliefs about the situation that serve as rationale for those actions). Each Evaluation contains not just patterns characterizing the conditions when the Evaluation applies, but also a set of Dialog trees that direct the automated tutor's possible lines of discussion when the Evaluation triggers. For instance, an Evaluation whose pattern looks for certain kinds of forces being sent along certain routes to carry out certain tasks might lead to a Dialog tree that aims to elicit and critique an expected range of student rationales for those taskings.

The Dialog trees are composed of nested sets of nodes, each aimed at eliciting a particular observation from the student. Each node has a (set of) questions such as "What sort of force do you think you're facing in your area east of the river?" or "Can you characterize what sized unit you are seeing here?" The student is encouraged to use a mixture of fill-in-the-blank forms and map or force-structure gestures to play their part in the multi-modal conversational interaction. If a student cannot give a reasonable answer to the questions at a particular Dialog node, then there may be a set of nested nodes that provide a way for the tutor to explore the problematic issue in more detail. In a sense, the Dialog nodes form trees that represent possible arguments for key observations about the TDG situation.

The main piece of a ComMentor session is rooted in this kind of question and answer interaction with the tutor primarily asking the questions. It is possible for the student to interrupt the main flow by asking questions of their own to a limited extent. The discussion ends when the tutor runs out of triggered Evaluations and accompanying Dialogs. At that point, the system offers the student a chance to reflect on their performance in the scenario. The tutor's assessment of the student's performance can also be used to guide selection of future TDG scenarios.

From an implementation point of view, ComMentor's design is based on three sets of data structures and five main processing modules (see Figure 1 for the abstracted architecture diagram reflecting the final Phase II implementation; see "Results" subsection "Final Prototype Architecture and Design" for a more detailed figure and accompanying discussion). The central data structure is the Scenario Script which serves first to initialize and later as a source of updates for the other two data structures—the Situation Model and the Session Agenda. The five processing modules work primarily as a cycle from Input Interpretation, Pattern Matching,

Agenda Management, to Output Generation (with a possible detour to Domain Inference in the midst of Pattern Matching):

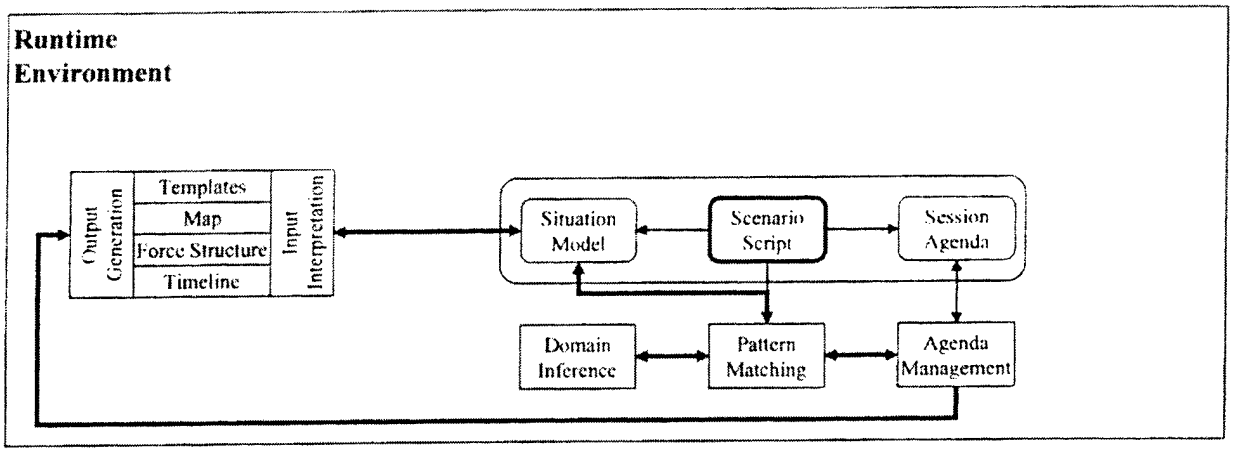


Figure 1. ComMentor Phase II Abstract System Architecture Design.

1. Input Interpretation – builds formal situation descriptions from form-based input augmented with map, force-structure, or timeline mouse clicks, icon placements, and dragging actions:

2. Pattern Matching – attempts to categorize the student's input as fitting with some expected set of patterns associated with Evaluations and Dialogs. Those Evaluation and Dialog patterns are modulated by the state of the tutorial interaction (e.g., the state of the Session Agenda) which in turn depends on the previous course of the interaction.

3. Domain Inference – applies rules from a domain knowledge-base to answer a range of relatively concrete questions, and derive alternate situation descriptions from what is given by the Input Interpretation module.

4. Agenda Management – based on pattern matches (driven by input and domain inferences) new nodes are added to the Agenda, old nodes are closed and moved to the history (marked either as succeeded or failed), and pending nodes may be rearranged. The pending nodes represent the tutor's evolving plan for further interaction with the particular student.

5. Output Generation – uses the current state of the Agenda to choose a next tutor move, playing the presentations that the student sees, and tweaking the user interface as needed to set up the next student input.

The Situation Model is built up incrementally from student input. It includes the orders the student has issued, as well as the student's answers to questions from throughout the tutoring session. For instance, the Situation Model might come to contain an order that sends a tank company down a particular road to attack an enemy force. It might also contain a record of the student's observation that there are friendly scouts on that road currently engaged with the enemy forces.

As suggested earlier, the Scenario Script contains expert authored trees of Evaluations and Dialogs, and each Evaluation and Dialog node may include patterns for entry and/or exit conditions. The patterns get activated and deactivated as Evaluation and Dialog nodes move through the Agenda. The Scenario might contain an Evaluation with a trigger pattern looking for an attack down that road on that enemy force. The Pattern Matching module is constantly checking for complete matches against active patterns (invoking the Domain Inference module as necessary to pursue such matches). Successful pattern matches lead to changes in the Agenda structure, which in turn lead to changes in the set of active patterns. In this case, the Evaluation's pattern would be satisfied, the Evaluation placed on the agenda, and eventually processed so that its first Dialog node too got onto the agenda. Eventually the Dialog node would move to the front of the agenda at which point its exit condition pattern would be activated.

Again, the point is that Evaluation nodes represent "good" and "bad" (fragmentary) solutions specific to a scenario. It is much easier for SMEs to provide context-bound evaluations than to come up with a complete set of general purpose analysis and assessment rules. When an Evaluation fires, it may activate other (nested) Evaluations as well as Dialog trees that encode tutor discussion strategies. In our example above, the activated Dialog might discuss the pros and cons of that particular plan of attack. There might also be a nested Evaluation looking for whether or not the student issued orders for the scouts to get out of the way of the oncoming assault (which in turn could activate another set of Dialogs covering the risks of fratricide in the situation).

Evaluations are also tagged with Curriculum points so that when a student's input matches the Evaluation's pattern, associated Curriculum points may be credited or debited. For instance, the nested Evaluation might link to a Curriculum point about minimizing the risk of fratricide. Thus Evaluations allow ComMentor to recognize strengths and weaknesses in the student's solution based on contextual factors, cue up further evaluative expectations and discussions that can refer to and exploit those factors, and also updates a more general persistent student model of skills that cross Scenarios. Use of multiple solutions increases the probability that the system can match a portion of the student's solution with a portion of an expert's. There may, for instance, be several different ways of evacuating the scouts from harm's way. The student's input (and thus what can be matched by Evaluation patterns) includes both overt actions (e.g., orders) and rationale (e.g., answers to questions that probe for student assessments of relevant situational features).

Dialog nodes directly control the ongoing Socratic interaction. Tutor output and student input may use a combination of graphical and textual means—what is referred to as multimodal I/O. Similar to Evaluations, SME authors can create context-bound discussions, so Dialog nodes can refer to the specifics of the concrete situation rather than deal in generalities. The discussion of fratricide in this case can refer to the particular units in question and their particular spatial and temporal relationships. Also similar to Evaluations, Dialog nodes can be linked to general Curriculum points to allow the incremental update of a student model that crosses Scenarios. Dialog trees allow construction of various discussion patterns such as successively more refined argumentation, and option enumeration/evaluation. These structures remain responsive to what the student actually says in response to tutor questions. The ultimate goal is

to continually refine the system's assessment of the student's reasoning skills and to continually be stimulating the student's thinking.

Benefits of the ComMentor Approach

Benefits of our approach include:

- Intelligent automated instruction of battlefield command reasoning – As with most automated instructional systems, our approach will enable students to learn "any time, any where," without the expense and scheduling rigidity of live human instruction. However, unlike CBT, WBT, and simpler ITSs developed to date, our approach will support sophisticated tutor/student interactions that stimulate the student's thinking skills for an extremely complex domain: battlefield command reasoning.

- Feasible scenario-based knowledge acquisition – Our scenario-based approach helps provide context that eases entry of subject matter expertise as compared to pure model-based knowledge entry. It also ties lessons to concrete situations that enhance comprehension and retention during the tutor/student interaction as compared to feedback couched at a more general level. Our authoring tools enable entry of essentially all system knowledge and content without programming, and can be extended to diminish the need to directly manipulate internal representations. By making it feasible to enter many Evaluations for a given scenario, the system will be able to draw upon the perspectives of many different experts to evaluate each student's solutions more accurately. By representing contextual factors the system makes it possible to probe for student rationale, and to make Evaluations sensitive to that rationale. Ultimately, by making it possible to create new scenarios quickly, it will be feasible to create a rich library of training scenarios that covers a wide range of situations and learning objectives.

- Adaptive, flexible instruction – Our system employs a mix of instructional strategies, exploits both scenario-specific and generic knowledge, ties tutoring session behavior to general curricular goals, and varies its lines of Socratic discussion, clarification, and feedback based on these factors plus student input throughout the session. The result is adaptive, flexible instruction that focuses on the student's needs, that is sensitive not only to what the student does but also to aspects of why they do it, and that leads them to construct an expert-sanctioned understanding of a situation and its implications. In short, the system provides competent Socratic tutoring for problems of a complexity that defy other earlier approaches.

Innovations

ComMentor represents an attempt to make the benefits of one-on-one Socratic tutoring on battlefield command reasoning skills available to a large number of U.S. Army officers by automating the tutoring process. It is the most sophisticated training application yet attempted for automated Socratic tutoring—an application representative of the large range of professional-level decision-making tasks that constitute perhaps the most important source of value in our increasingly service and knowledge oriented economy. We believe that the practical implementation of tutoring systems for this application, and for other similarly large, complex

domains, will be enabled by avoiding "deep," expert system representations of domain concepts in favor of scenario-specific (case-based) knowledge representations that are "deep enough."

In terms of technology then, it is the combination of general and specific knowledge—or in conventional AI terms, the coordinated exploitation of rule-based and case-based reasoning that differentiate ComMentor from prior systems. In our Phase II ComMentor implementation, this hinges specifically on the ability to link general domain Concepts and Curriculum points to Scenario-specific Evaluations and Dialogs covering an expected range of student behavior and rationale, while supporting adaptive deployment of contextually appropriate tutor behavior. The result, as suggested above, is a first step towards a new and valuable class of automated tutors for professional-level skills training.

Background

The Background to this work includes a wide range of disciplines and prior research. To focus the discussion here we consider five distinct areas: (1) "Scenario-Based Instruction," (2) "Socratic Tutoring Systems," (3) "Multimodal Input Processing," (4) "Case-Based Situation Interpretation/Diagnosis," and (5) "Application-Specific Authoring Tools."

Scenario-Based Instruction

The validity of the scenario-based approach has both intuitive appeal and empirical backing. As early as 1940, Gragg (1940) argued for scenario-based instruction. By presenting scenarios, which illustrate the important principles, the student can see how principles are applied in operational contexts and tasks. It also defeats the well-known problem of inert knowledge first described by Whitehead (1929) and frequently validated by other researchers. Inert knowledge is information or principles that a subject knows and can recall, but which he does not apply when the situation clearly calls for it. Scenario-based instruction (and related concepts such as example-based instruction, anchored instruction, case-based instruction, simulation-based instruction, and situated instruction) overcomes this problem by showing students the application of principles in an operational setting, and forcing them to apply them as well.

Learning different types of problem-solving tasks requires different types of tutoring system architectures. However, we believe that there are a manageable number of distinct types of abstract problem-solving tasks. We are encouraged and inspired by research in Generic Tasks (Chandrasekaran, 1987), Components of Expertise (Steels, 1990), and Goal-based Scenarios (Schank, Fano, Bell & Jona, 1994), who all argue that the number of types of problem-solving skills is quite finite. For example, proponents of Goal-based Scenarios at the Institute for the Learning Sciences at Northwestern University assert that:

- Students learn effectively when they are pursuing goals within the simulated scenarios, and

- One can categorize goals into a small number of abstract types (e.g., "Investigate and Decide," "Run a System or Organization," "Pursuade," etc.) which strongly affect the architecture of the appropriate tutoring system.

ComMentor's style of scenario-based instruction might be characterized as "Propose and Defend." The student is confronted with a problematic situation and asked to propose a solution or response. The automated tutor then questions them about their response, eliciting their rationale, bringing them to see non-obvious (to a non-expert) aspects of the situation, and especially helping them see the implications of their proposed course of action in light of a more informed view of the context. The result is a kind of Socratic instruction, as practiced in many disciplines' professional-level education programs.

Socratic Tutoring Systems

Socratic tutoring is a recognized style of instruction, often associated with expert tutors (e.g., Glass, 1999), and known to produce effective outcomes (e.g., Rose, Moore, VanLehn & Allbritton, 2000). Adopting the Socratic style is not a simplistic tutoring "trick"— it is not about syntactically transforming all tutor interventions into question form. Rather, it is best thought of as a bias towards having the student construct as much of their knowledge as possible, while also encouraging reflection and analysis. While there has been ongoing interest in automating Socratic—or more broadly dialog-based—tutoring over the last three decades, there are currently only a handful of leading laboratories with a major focus in this direction (e.g., Kurt VanLehn at Learning Research and Development Center (LRDC), Martha Evens at the Illinois Institute of Technology (IIT), Art Graesser at Memphis, Johanna Moore at Edinburgh, and the COLLAGEN group at Mitsubishi Electric Research Laboratories (MERL)[1].

Research in mixed-initiative and Socratic tutoring systems have been carried out for more than 30 years. In fact, much of the early work in intelligent tutoring systems focused on the support of dialogs between the software and the student. Early work includes the Scholar system that taught South American geography (Carbonell, 1970). It is significant in its early use of domain knowledge representations (about geography) that are separate from the instructional decision-making knowledge. However, this system taught facts about geography rather than reasoning skills and therefore did not pursue many goals typical of Socratic dialog. Collins (1976) developed a set of two dozen decision-making rules used by Socratic tutors and embodied these rules within the WHY system that taught meteorological reasoning about rainfall processes. Clancey (1987) developed Guidon, a tutoring system for diagnosing blood diseases that used as its expert model the Mycin rule-based expert system. A finding of the Guidon project was that expert system rule-bases frequently employed reasoning methods (in this case, rules) that were not intuitive when used to generate tutorial dialog utterances. Edelson (1992) describes Socratic tutoring in the domain of biology that relies on a case-base for its expert model.

Previous work provided useful findings and insights, especially in the identification of tutoring strategies. However, ComMentor required advancements in at least three areas: (1)

[1] All of these groups (except perhaps at MERL) are or have been funded by Susan Chipman at the Office of Naval Research. Most of them are now working together on joint projects. Dr. Chipman is also now funding Susann LuperFoy of Stottler Henke for work on dialog in tutorial contexts.

using complex plans and rationale structures supplied by experts as the basis for tutoring dialogs; (2) using expectations about multiple partial solutions for assessing student solutions; and (3) assessing and enhancing high-level thinking skills and habits as opposed to detailed knowledge and lower-level reasoning skills.

One observation to be drawn from past and current work is that most developers see unstructured language input as a useful complement to a Socratic tutoring approach. The general hypothesis is that much of the value in Socratic tutoring lies in having students work out for themselves what is going on in a given problem (thus leveraging the benefits of constructivist approaches and "self explanation") which would be somewhat short-circuited by pre-structuring lists of possible answers (or answer components) as in a menu-driven input system. For these and other reasons related to tutoring behaviors observed in our sample sessions we chose to pursue a natural language interface in our Phase I effort, and through the first half of our Phase II work. However, our evolving appreciation for the breadth of coverage required to support a natural language interface to ComMentor, and the need to focus our limited resources on demonstrating the possibility of Socratic tactical tutoring, led us, during the second half of our Phase II effort, away from unrestricted natural language understanding and towards the more pragmatic approach of limited phase parsing in the context of a forms-based input mechanism.

Multimodal Input Processing

Natural Language Processing (NLP) has been an active area of artificial intelligence research since the beginning of the discipline. Much work in the field draws on well-established linguistic traditions—adopting a multi-phased analysis mechanism that starts by emphasizing syntactic parsing. However, much of the most effective work in the field has taken alternate approaches—emphasizing semantic analysis tied to some particular context of language use. This often leads to faster, more useful processing, that is robust in the face of real-world input (ungrammatical, misspelled, or telegraphic input). In fact, the dominant trend in Socratic tutoring language components is to emphasize more effective semantic analysis techniques (e.g., Brown & Burton, 1975; Glass, 1999; Wiemer-Hastings, Wiemer-Hastings & Graesser, 1999). We note, however, that Glass (1999) has effectively critiqued the use of Latent Semantic Analysis (LSA) based approaches (as in AutoTutor) as unlikely to be very effective in domains that go far beyond general discussions of terminology and definitions, because LSA matching is primarily sensitive to overlaps in word meaning rather than to structured relationships among actions and situations so critical to scenario-based instruction.

Our input capture and processing approach aimed to make use of any implementable modes of interaction that could be exploited to extract structured descriptions. That included an approach to language interpretation based on nested patterns that ultimately proved too costly and insufficiently robust for the context. It also included integration of input generated by student manipulation of visualizations commonly used in the military domain: situation maps, force-structure wire-diagrams, and timelines. Integration of gestures manipulating these graphical user interface (GUI) components with the final forms-based input system proved less difficult than might have been the case had we continued with natural language processing since the forms themselves made heavy use of mouse gestures that could then be extended to include references to the visualizations.

Case-Based Situation Interpretation/Diagnosis

From the start, ComMentor aimed to go far beyond instruction on terms and definitions. The goal has been to tutor students in effective application of high-level reasoning skills in the complex domain of dynamic battlefield command. Gathering structured input descriptions was just a necessary first step, so that our student assessment could make relatively fine situational distinctions. This is where the technology of case-based situation interpretation and diagnosis was required. Our experience in building ITSs for similar domains showed that constellations of fragmentary cases—contextually relevant partial solutions and known problematic responses—can be an effective way to diagnose many student successes and failures in complex tactical scenarios. Furthermore, a case-based approach to building such recognizers tends to make it easier for domain experts to manage and maintain a tutoring system over time since the bulk of the knowledge in the system is collected in quite concrete contexts, rather than requiring the degree of abstraction typical in more rule-based systems.

Application-Specific Authoring Tools

The ITS technology has been under development for over three decades by researchers in education, psychology, and artificial intelligence at military and academic research labs. Research studies carried out so far show that students taught using ITSs generally performed better and learned faster, compared to classroom-trained students. However, a major impediment to the widespread use of intelligent tutoring systems is that they frequently require custom software development and are therefore expensive to create. Consequently, the development of authoring tools to enable lower-cost development of these tutoring systems is also an area of active research.

We hold, along with most of the ITS community, that a critical component of real-world success for ITS technology will be the ability to provide tools that let domain experts take more control of production and maintenance of training systems. We have already noted that any knowledge-based software system is only as powerful and up-to-date as the set of concepts, rules, cases, and other objects in its knowledge base. Knowledge in such systems tends to have a limited shelf-life as the world changes (e.g., new weapons and vehicles, new operational concepts, new doctrine) and goals for the system evolve (e.g., shift from focusing on conventional conflict to focusing on missions other than war). To remain effective in the long run, a system must be maintainable and extensible; to remain cost-effective, as much maintenance and authoring as possible should be supportable by available staff and those with first-hand expertise in the domain.

Raw manipulation of the underlying data structures that enable an ITS to function tend to require detailed expertise in artificial intelligence (AI) and ITS technologies. To enable domain experts to take on maintenance and authoring tasks, manipulations of system structures must be translated into terms that are meaningful in the domain. Furthermore operations must be clustered, sequenced, and modeled so that appropriate system modifications become easy, and invalid modifications become essentially impossible. Achieving these ends requires creation of application-specific authoring tools. Since we see this as an important consideration in every ITS we construct, this is an area where Stottler Henke has had extensive experience.

Much of the prior research in ITS authoring tools has been in support of procedural task training domains where there is a "correct" method for carrying out each procedure. Guralnik, (1996) describes an authoring tool which applies a content theory of task knowledge which enables the tutoring system to generate replies to important questions from the student, such as "What do I do next?" and "How do I do that?" (Munro & Pizzini, 1995) describes the RIDES system, which enables authors to create graphical training simulations (frequently, of devices) integrated with the intelligent tutor. In contrast, ComMentor required a new type of ITS authoring tool: one that captures the reasoning of experts in support of solutions, even when there is no single correct solution. In the end, as described in detail below, we constructed an integrated authoring environment that combined some relatively traditional authoring components (e.g., ontology authoring and curriculum authoring) with new tools for situation and tutorial dialog authoring.

Method

The discussion of our Method is broken down into two subsections reviewing first, the "Project Objectives," and second, the "Project tasks." Appendix A "Project Chronology" provides additional details on the work accomplished during this project.

Project Objectives

The Phase II proposal for this project called out eight project objectives:

1. Continue to identify and encode tactical analysis knowledge that should be represented by the system to support the evaluation of student solutions to tactical scenarios, in order to refine the system's student model. In Phase II, particular emphasis will be placed on (a) getting reasonably complete coverage of attributes and inferences in basic areas such as units, vehicles, and weapons, complemented by authoring tools to ease additions and modifications to the accumulated knowledge-base; (b) characterization and prioritization of the space of possible extension sub-domains of tactical analysis knowledge; and (c) further exploration of ways to unify ComMentor's tactical analysis knowledge with other efforts in the DoD community (e.g. adoption of proposed knowledge representation language standards such as the Knowledge Interchange Format (KIF) or the DARPA Agent Markup Language (DAML), and evolving ontologies such as those sponsored in DARPA research programs like High Performance Knowledge Bases (HPKB) or Command Post of the Future (CPOF).

2. Continue to explore and evaluate candidate interaction techniques that achieve the same types of instructional goals achieved by student-coach interactions carried out by expert human instructors. In Phase II we will devote particular emphasis to (a) exploring the extensibility of our current textual language processing approach versus other more structured input techniques; (b) exploring the applicability of evolving techniques for speech input and output; (c) pushing on the utility of, and coordination techniques for, multi-modal interactions that combine graphics and text (or speech) input and output.

3. Continue to identify, classify, and implement the types of tutoring strategies that can be automated in order to select and generate effective instructional interventions based on the student model. Many of the strategies identified in Phase I remain to be implemented or even seriously investigated. Others likely remain to be discovered. Klein Associates is particularly

eager, and well positioned to develop a model of Tutoring Instruction/Mentoring Strategies. Additional data collection and preliminary development of such a theory during our Phase II effort should help to provide a firmer empirical and theoretical foundation for ongoing design and implementation efforts.

4. Refine our methods for authoring ITS scenarios by entering good and bad sets of decisions and rationale, and then annotating these solutions with skills and knowledge that are demonstrated (positively or negatively) by each solution. In Phase II, emphasis will be placed on systematizing our approach to these tasks, developing custom authoring tools, and embedded guidance in application of those tools. Efforts to ease acquisition of this scenario-specific information complement the efforts in objective 1 to get a handle on the general (domain-specific) tactical analysis knowledge.

5. Develop a fully functional software prototype that not only illustrates the key ideas related to tactical analysis knowledge representation, student-coach interaction methods, and interactive tutoring strategies, but also serves as an effective learning aid for Army officers. We expect to observe, analyze and implement on the order of another 6 scenarios during Phase II. In concert with all the objectives listed above, this is a logical next step in efforts to scale-up the techniques identified and illustrated in Phase I.

6. Demonstrate the effectiveness of the tutoring system prototype developed in Phase II. Evaluation will be an ongoing effort throughout the Phase II, starting with formative evaluations of the Phase I prototype, continuing through early user trials and observations of the evolving Phase II system, and culminating in a controlled study of system effectiveness using tactical reasoning skill evaluation metrics and instruments, most likely adapted from those developed by ARI for use in the TLAC program.

7. Develop guidelines and baseline data on scenario authoring. As a result of developing a half-dozen scenarios, we will acquire a far better grasp of effective techniques, typical pitfalls, and likely costs involved in extending the system. In addition to authoring techniques concretized in authoring tools, and guidelines explicitly embedded in such tools' interfaces and/or help systems, there is likely to be other "wisdom" gained about how to build effective scenarios, including the costs and typical tradeoffs inherent in such efforts. The results of objectives 6 and 7 together will feed into addressing our final objective number 8.

8. Develop a preliminary analysis of the issues likely to bear on long-term acceptance, effectiveness, and maintainability of an expanded, fully operational ComMentor. In addition, consider the portability of the techniques embedded in ComMentor to other, more commercial domains and applications.

Project Tasks

The Phase II proposal called out sixteen tasks:

1. Perform Formative Evaluation of Phase I Prototype. Phase II work began with a formative evaluation of the Phase I prototype by our consulting subject matter experts. The

object was to identify aspects of the system that seemed to work well, as well as flaws and limitations. The comments collected fed into our ongoing design and implementation efforts.

2. Identify and Adapt Phase II Scenarios and Plan Tutorial Observation Sessions. Building on our successful approach to data collection on naturalistic battlefield reasoning tutoring, developed during Phase I, we worked with our consultants and subcontractors to select a set of six tactical scenarios, and arrange tutorial sessions with military personnel as students and our consultants as expert instructors. We scheduled four tutoring sessions per scenario, all with Gen. Holcomb and Maj. Schmitt as instructors.

3. Carry out Tutorial Observation Sessions and Analyze Results. The tutoring sessions were conducted using the distributed approach piloted so successfully in our Phase I work. Members of both the Stottler Henke and Klein Associates teams participated as observers. Klein led the analysis effort under direction from Stottler Henke.

4. Develop Instructional Strategies Model. One of the major outputs of the tutoring session analysis was an Instructional Strategies Model developed by Stottler Henke and Klein Associates. The goal was to establish a solid empirical and theoretical basis for the Instructional Strategies Knowledge Base.

5. Elaborate Instructional Strategies Knowledge Base. Elaboration of the Instructional Strategies Knowledge Base occurred in waves (as was true for the other three knowledge bases as described below). The purpose of the Model in the context of this project was to help us systematically identify and organize necessary tutoring control knowledge and mechanisms.

6. Elaborate Scenario Knowledge Base. The development of scenario-specific knowledge is a top priority in this project. The central contents of the scenario knowledge base will be fragmentary scenario solutions representing good and bad approaches to the problem. Each such fragment will include a set of recognition conditions, as well as a set of belief and rationale statements with indications of how those statements relate to the facts of the scenario, to good domain practice, and to possible tutoring moves.

7. Elaborate Curricular Knowledge Base. This knowledge base contains information about the principles, concepts, and skills to be taught by the ITS. That information includes interrelationships among curricular items, as well as possible ways to present information about, or probe for understanding of them. Ultimately, the individual student model is formed, in large part, as a reflection of this curricular knowledge base—recording what topics the student has been exposed to, in what form, and with what success.

8. Elaborate Domain Knowledge Base. It is critical to have a basic competence for representing and reasoning about the facts of the battlespace, but we limited the effort devoted to stretching that competence as much as possible. That is because the development of formal reasoning models for such a complex domain is known to be an open-ended (and quite difficult) problem; also, other research programs have focused, and are focusing, on developing such competences. From experience in developing other case-based ITSs, we believe the greatest

payoff—in rapid system construction and end-user extensibility—are achieved by focusing relatively more effort on scenario-specific knowledge.

9. Refine Phase II System Design. From a primary concern with data collection and knowledge codification, we shift here to system design and implementation. Based in part on the initial Phase I prototype formative evaluation, we reviewed the proposed Phase II system design looking for newly uncovered requirements, or opportunities to improve the system.

10. Implement ITS Core Extensions. We devoted a substantial block of implementation time to enhancing and refining the basic system code originally developed for the Phase I prototype, in keeping with the dictates of the newly refined Phase II system design. The object was to provide a firm foundation for a more functional, scalable and robust system.

11. Explore Alternate Interaction Mechanisms. We proposed to experiment with a range of alternate interaction mechanisms. Our work on natural language input processing proceeded through the first year, and convinced us that we ought to back off that approach (and the related goal of speech input processing) in favor of a forms-based input mechanism, heavily integrated with supporting domain-specific visualizations (e.g., maps, force-structure trees, and timelines).

12. Implement Authoring Tools. The development of authoring tools ultimately aimed at subject matter experts was a significant thrust of this project. We built tools for capturing not only scenario information, but also domain and curriculum knowledge.

13. Observe SME Use of Authoring Tools. We had our SMEs review the evolving authoring tool suite at several points during their evaluation, culminating in a small authoring trial with the nearly complete suite towards the end of the project.

14. Evaluate Phase II System Releases. We conducted limited trials of the ComMentor system as it developed, including trials with active-duty Army officers in June of 2002 and April of 2003. The first round centered on the text-based language input mechanism and led to design of the forms-based input mechanism. The second round centered on the first draft of that forms-based system and helped identify many useful simplifications as well as opportunities to integrate the forms with other aspects of the system display.

15. Analyze System's Long-Term Deployment Issues. In order to provide as complete a picture as possible of ComMentor's utility, we devoted time at the very end of the project to identifying, and, as far as is possible, quantifying, the likely costs and benefits of deploying the system and maintaining it over the long haul. The object was to get a clear vision of the road to Phase III adoption of and wide-spread use of this work.

16. Prepare Final Technical Report. In the last months of the project wrote up our results in this final technical report, documenting our efforts and accomplishments.

Results

This Results section constitutes the bulk of the report. It is broken up into 8 subsections: (1) "TDG Data and Preliminary Analysis," (2) "Instructional Model," (3) "Domain Model," (4) "Curriculum Model," (5) "Scenario Model," (6) Final Prototype Architecture and Design," (7) "Final Prototype Runtime Capabilities," and (8) "Final Prototype Authoring Capabilities." Together, these give a comprehensive view of the work accomplished during this project. Appendix C supplements this discussion with an extended annotated example of ComMentor's runtime in action.

TDG Data and Preliminary Analysis[2]

During the Phase I project and in the first third of the Phase II project, we conducted a total of 30 TDG sessions with active duty officers from the U.S. Marines and Army. This sub-section describes those sessions and their results in detail.

Objectives. Our tutorial observation sessions were intended to provide critical information on a variety of issues:

- Ranges of student ability.
- Expert mentoring techniques.
- Typical student behavior in specific scenarios.
- Typical mentor approaches and sequences in specific scenarios.
- Specific scenario contents and presentations.
- Relevant background domain knowledge.

Data on these issues was then used to inform the design and implementation of the ComMentor system. In particular, ComMentor was conceived as a scenario-based system, and so in our Phase II work we used these sessions as a way to develop detailed understandings of a set of six scenarios intended to form the initial training corpus for the delivered system.

Overview. Marine and Army Captains were invited to participate in individualized Tactical Decision Game training delivered in a mixed live/electronic format. These TDGs were designed and mentored by a pair of retired Marine officers recognized for their expertise in tactical training (see Schmitt, 1994; Schmitt, 1996). In all cases, one of the mentors was present in the room with the student while the other was connected by a voice phone line and a shared whiteboard application hosted in a web browser. The shared whiteboard was used to present task organization wire diagrams and scenario maps; it supported shared annotations, pointers, and screen capture. The on-site mentor provided technical support as well as normal mentoring

[2] This sub-section is taken essentially in its entirety (with some editing) from our I/ITSEC-2002 conference paper.

(informed, on account of collocation with the student, by the interpersonal cues that might be missed by the remote mentor).

Most of the TDG sessions lasted approximately 1.5 hours, except for the final sessions devoted to scenarios based on a futuristic Objective Force, which required over 2 hours because of time spent discussing the projected systems and capabilities envisioned in Future Combat Systems (FCS) concepts. Sessions started with an informal interview bearing on the student's background. Then the project, team, and purpose of our session were introduced. The mentors took the student through an introduction to the web-based whiteboard tools, and then finally, introduced the scenario for the session. At every point, students were allowed and encouraged to ask questions.

Sessions were tape-recorded, and occasional whiteboard screen-shots were taken during the interaction to capture the state of a shared map with its unit positions and other annotations. Sessions were also generally observed by at least two project team members in addition to the two mentors. In most cases, those observers were not present in the room, but were also connected to the teleconference and the shared whiteboard. One project team member was skilled in cognitive task analysis and modeling, while the other represented AI model design and implementation expertise. As noted earlier, the main purpose of these observation sessions was to provide data on which to base a cognitive model of battlefield command reasoning skills tutoring, preparatory to embedding that model in a functioning computer tutor.

Students. Students were active duty Marine or Army Captains. Marines were primarily tactics instructors recruited from the Quantico training center. The Army officers were drawn from a range of units and specialties at Fort Riley. A total of seventeen Marine Corps Captains participated in the thirty tutorial sessions. Four Marine Corps officers participated in two different sessions. One participated in three sessions over the course of the study. Those who participated in more than one session did not repeat the same scenario. The Marine Corps officers, all affiliated with combat arms, had a range of experiences, but all had been Platoon Leaders and Company Executive Officers. Their length of service varied from 6 to 14 years. Eight U.S. Army Captains participated in the tutorial sessions. Of those, two were currently Company Commanders. The Army officers were affiliated with combat arms and other branches of the service to include Intelligence, Signal, Armor, and the Quartermaster Corps.

Materials. The primary materials in this study were the TDG scenarios themselves. During Phase I, we held six sessions and worked with three different scenarios. During Phase II, we held a total of twenty-four sessions: four sessions for each of six different scenarios. Table 1 below summarizes the Phase II scenarios.

Table 1

Scenarios for Phase II Tutorial Observations.

Phase II Scenarios	Topic	Student Role
1st 2 Scenarios	Present-day armor	Battalion or Troop Cmdr
2nd 2 Scenarios	Interim Brigade	Battalion Cmdr
3rd 2 Scenarios	Objective Force	Company Cmdr

Each scenario typically consisted of a narrative that was read to the student with some supplementary commentary by one of the mentors. In addition, when the friendly or enemy task organizations were complex enough (and relevant and well known enough) a wire-diagram of those organizations was generally provided. Finally, every scenario had a (generally abstracted) map of some restricted area of relevant terrain. Mentors also provided answers (either factual or extemporary scenario extensions, as appropriate) to most questions asked by students.

As a sample, Figure 3 shows the narrative introduction to the first scenario used during the Phase II sessions. Figure 2 shows the accompanying map. In this case, the Blue task organization was considered simple enough that a force structure wire-diagram was not provided. Likewise, since the Red organization was unknown, no diagram was provided for that force either. Note that when the narrative says, "You are moving into the assembly area as shown," a set of movable icons were positioned on the map to indicate roughly where the student's forces were supposed to be located at that time. Both the student and mentors were free to later move those icons as the discussions about the student's response unfolded.

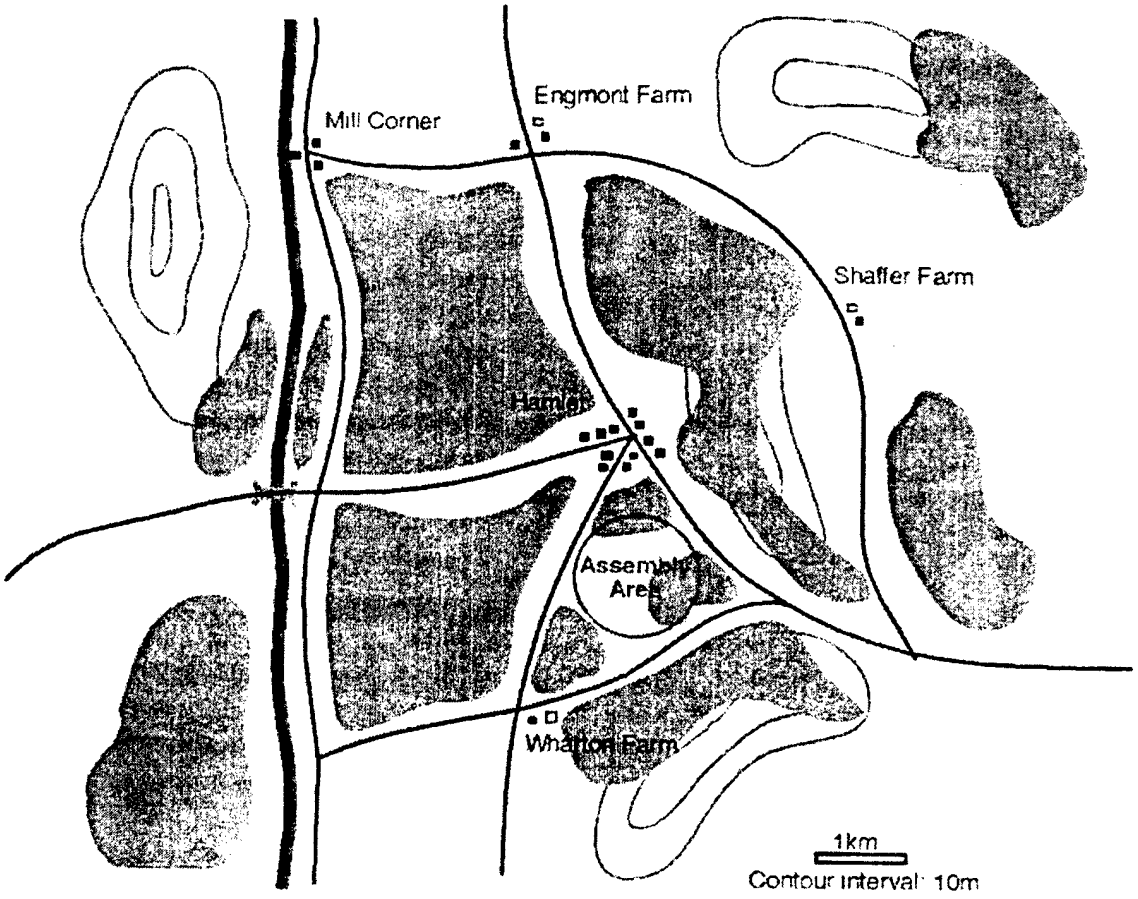


Figure 2. Map for the "Enemy Over the Bridge Scenario."

Procedure. As described, each session started with introductions of the student, background on the project, participants, purposes, and process of the session, and a tutorial on the web-based tools. When all that was done, the TDG proper began with the reading of the scenario narrative, and proceeded through a series of stages, as reported in the Results session

below. When the mentored scenario itself was over, each session ended with a brief post-scenario interview. In later runs this centered on the student's prior tactical training experiences, their appreciation of this experience, and their potential interest in additional sessions, either with live mentors or with a computerized mentor.

Enemy Over the Bridge

You command a balanced tank-mech battalion task force consisting of 2 tank companies (A and B), two mech infantry companies (C and D), a scout platoon and a mortar platoon, plus your medical, support and maintenance platoons organized into a company train.

You are fighting a capable, mechanized enemy equipped with T-62s and BMPs and supported by towed and self-propelled artillery.

Host-nation forces hold the bridge and the river line to your west. You have been told the river is unfordable. Reconnaissance elements are operating west of the river. In 48 hours, the division begins a major offensive west across the river to destroy enemy forces in zone, with the main effort in your brigade's zone. Your battalion will spearhead the brigade's attack.

You have been instructed to occupy the assembly area shown on the map east of Hamlet in preparation for the 0500 attack the morning after next. You are moving to the assembly area as shown. At 0100 your scout platoon, which is forward reconnoitering the route to the assembly area, makes the following report:

> Battalion, be advised have just made contact with a host-nation motorized reconnaissance patrol that was operating west of the river but about 2 hours ago was forced east across the river under fire. They came across the bridge and then via Wharton Farm. The reconnaissance patrol leader reports there is no sign of friendly forces holding the river line or the bridge and that enemy mech infantry and some tanks have been moving east across the bridge for almost 2 hours. He says he counted 10 T-62s in the last half hour; does not know how much mech. He says he has reported this twice to his higher headquarters. Over.

A few minutes later the scout platoon leader adds the following:

> Be advised we've got enemy mech infantry occupying our assembly area in strength. I say again they are enemy and not host-nation forces. I've got a solid visual on several BMPs. Don't know the size, but I estimate at least a company. They seem to be still moving into the area, over.

Suddenly, you start to see artillery impacting in the woods just north of Alpha Company at the head of the battalion column.

Moments later, you hear automatic weapons fire from the direction of the assembly area. "We're in contact!" the scout platoon commander shouts over the radio.

What do you do?

Figure 3. Narrative Introduction to the "Enemy Over the Bridge" Scenario.

Result: Ranges of Student Ability. Captains are a remarkably diverse lot. They can range in experience from 6 years to 14 years of active duty service. They can be deeply immersed in tactical art, or they can be extremely remote from it. As an example of one extreme, we worked with a very experienced Captain (an armor officer, with almost a decade as

an enlisted Soldier, and experience in Desert Storm, before even receiving his commission) who was extremely quick, both to act appropriately, and to grasp the flaws in his approach as soon as they were even hinted at. At the other extreme, one participant, lacking a combat arms background, also seemed to lack the language and mental models to grasp the variety of the elements in the tactical situation, and essentially froze when first presented with a TDG situation description.

Of course we primarily saw students distributed across the broad intermediate range of background and ability. Sampling the full range was quite valuable to our work, as it ensured that we got a more complete picture of how students might respond to the TDG scenarios. It also drove home the point that there would be limits on the final system's range of applicability; some students—especially those with too little experience—will not be well served by the tutorial behavior authored to address the needs of the more average Captain.

Result: Expert Mentoring Techniques. One of the major purposes of these observations was to see how expert mentors conduct the kind of training we aim to simulate in the ComMentor ITS. Accordingly, we report here the general mentored TDG interaction structure observed across our many sessions as a Result (rather than detailing it in the Procedure section above).

The scenario's introductory narrative generally ended with a challenge to the student to respond to the situation. In some scenarios, students were invited to start responding as soon as they wanted to (that is, they did not have to wait until the end of the narrative). In complex scenarios, students were asked to perform an orienting task, such as a hasty terrain analysis. That orienting activity might be requested in the middle of the narrative, in scenarios where a later crisis event potentially called for immediate action.

In all scenarios, a degree of time pressure was created by setting a limit on how long the student could think before acting. In some scenarios—particularly those with an unfolding crisis—the mentors reported consequences if the student delayed (e.g., changes in the situation, usually not to their advantage).

The students almost always produced some kind of response to the situation, though often the mentors had to help in drawing it out. Often there was a mini role-play with the student issuing fragmentary orders, as over a radio net, while a mentor played the parts of subordinates trying to understand and react to the orders (or superiors responding to reports and requests).

Once a student felt their response was complete, the mentors would typically proceed to probe for their rationale: what did they think was going on in the (usually ambiguous) situation? What were they trying to accomplish and why?

These rationale discussions generally transitioned into discussions bearing on the reasoning themes central to our training objectives: What did they think the enemy was trying to do and why? What did they think their higher commander would want them to do? What did they think would happen in some amount of time if they adopted their proposed course of action? Discussions, steered by the mentors, ranged from general themes, to specific facts of "battlefield calculus," to detailed critiques of particular courses of action, and proposals for alternatives.

Each scenario ended with several minutes of mentor-led reflection. The mentors consistently asked a suite of meta-level questions: How do you think you did? What do you think you did best? What do you think you did worst? What did you learn?

Result: Typical Student Behavior in Specific Scenarios. By running each scenario several times (four times each for the Phase II scenarios) we were in a position to see a range of possible student behavior on each problem. The first observation was that—as suggested by our expert mentors in advance—there were definite patterns and frequent commonalities in students' responses. For instance, in the "Enemy Over the Bridge" scenario, the most common initial reaction was to deal forcefully with the firefight in the assembly area, but to essentially ignore (or treat as a lower priority) the continuing flow of enemy troops over the unsecured bridge. With common student response patterns, it follows that there were common topics in mentored discussions.

The second observation was that different students required different amounts of discussion and prompting (often forcing the mentors to try several different approaches) in order to recognize key points about the situation. Getting the student to see the importance of the bridge might take as little as asking "Can you identify any pieces of key terrain in this situation?" or as much as a long discussion on the Blue and Red intentions in the situation, and explicit visualization exercises about what was likely to happen over time.

Result: Typical Mentor Approaches and Sequences in Specific Scenarios. As already noted, common student responses tended to elicit common mentored discussion topics. We inventoried the discussion topics for each scenario by analyzing and comparing across session transcripts.

For instance, again drawing on "Enemy Over the Bridge," we identified recurring discussion topics such as: (1) getting the student to realize that they must stop the flow of new enemy forces over the bridge; (2) getting the student to realize that attacking through the assembly area, on to Hamlet, and then to the bridge is equivalent to attacking an enemy surface rather than a gap; (3) getting the student to enumerate a plausible set of alternate courses of action for controlling the bridge; and (4) getting the student to critique each of those courses of action on its strengths and weaknesses.

Result: Specific Scenario Contents and Presentations. For each scenario observed in Phase II we gathered the basic scenario materials from our SMEs for adaptation and incorporation in the system. For instance, the materials shown in Figure 2 and Figure 3 have been converted into a format appropriate for use in the ComMentor prototype (see, for instance, Figure 11 and Figure 13 in the discussion of the ComMentor Runtime below). We also gathered the force structure diagrams and relevant icon sets used by our SMEs (again, as reflected in Figure 11 and Figure 13).

Result: Relevant Background Domain Knowledge. The expert mentoring we observed is well beyond the capability of any current AI approach in the general case. However, we do not necessarily need to match all aspects of our expert mentors' performance to have a useful system. Even for those aspects we do care about, an advantage of a scenario-centered approach is that we do not always need to build our system for the general case.

While ComMentor does not need to be able to reason as richly and creatively as our experts, it does need to be able to represent all of the kinds of statements of fact, intention, and conjecture that students typically utter during a TDG interaction. We can also reasonably aim to have it answer many basic questions of fact and simple computation that our experts classify as "battlefield calculus" (e.g., How many tanks in an enemy battalion? How fast can a tank company move over a paved road? How far and how fast, and with what probability of kill can a given tank shoot?).

The transcripts of our observed tutorial sessions provide an extremely rich source of information to address these domain knowledge representation issues. Combining analysis of transcripts and guidance from our expert mentors with other sources of AI representation for the military domain (e.g., results of the DARPA HPKB program) ultimately allowed us to assemble a rich ontology for relevant aspects of the military domain.

Impact: User Interface Structure. Several aspects of the TDG interactions we observed seem to be critical to capturing the flavor of these sessions and enabling the kind of deep engagement and constructivist learning these sessions engender. Here we enumerate several of the most important.

- Situation map as the central visualization.
- Force structure diagrams.
- Discussion-based interaction.

Based on our observations and advice from our mentors, the ComMentor prototype devotes the bulk of its user interface to a map of the relevant territory, which serves as the central visualization for the scenario. While the terrain displayed can be limited, and the detail (purposefully) simplified, we try to ensure that the whole map is fully visible at all times (e.g., there is no need to scroll madly to see the entire area of interest). The map display is manipulable, both by the student and by the (automated) mentor, so it can reflect the evolution of the situation over time, and so that it can be used to express geographically related meanings that are not as easily stated in other modes, such as language (e.g., positions and relative positions, paths of movement, lines of visibility, etc.).

The prototype's map is complemented by a tree display that depicts what is known of the Blue and Red task organizations. The display serves both as a mnemonic to help the student remember some of the key facts of the scenario, and also as a pallet from which the student can pick icons to place on the map. The force tree is coupled to a set of timelines that are intended to provide a visual display for Scenario events.

Finally, to complement these visualizations, the student's main interaction with the system is through forms that reduce the need for complex language processing. Natural Language Processing (NLP) technology is not completely mature or robust, largely because, in the general case, it addresses an unconstrained problem. Of course, not overtly constraining the student is precisely what we had hoped to accomplish by adopting NLP for ComMentor. The current form-based system is significantly less costly and more reliable. However, it may slow

the students down, inhibit them from making complex statements, or lead them too heavily. These effects may ultimately interfere with students focus on the scenario and their construction of knowledge—thus running counter to the goals of Socratic interaction. For the time being, it is the only tractable technology that allows us to capture accurate structured descriptions that reflect the student's intentions and rationale.

Impact: Discussion Structures. In order to make ComMentor practical, we had to find some way to restrict the interpretation problem. To this end, we noted the following four observations:

1. Observed student/mentor discussions were frequently built around the explicit reasoning themes ComMentor is being designed to teach;

2. Common student responses frequently elicited common discussion topics;

3. The mentors, for the most part, remained in control of the course of the session, (e.g., choosing topics and framing the discussion on those topics; but

4. Students required different depth of detail in discussions (or number of alternate discussion approaches) to arrive at important conclusions.

ComMentor, then, was designed as a discussion-oriented system where the tutor maintains the primary initiative in introducing and pursuing discussion topics based on the evidence it is able to collect about the student's understandings and intentions. Some initiative may be temporarily ceded to the student in cases where they interpolate questions (primarily about domain facts) into the flow of the conversation. But for the most part, the tutor asks the questions, and then tries to interpret the student's responses in relation to those questions. The formal constructs that underlie this approach are Evaluation and Dialog nodes.

For each scenario, discussion topics represented as trees of Dialog nodes are tied to Evaluation nodes whose patterns specify student behaviors or rationales that, according to the expert scenario author, are worthy of critique or complement. These scenario-specific Evaluations (e.g., the student says they are aiming to take control of the bridge, but they are doing so by attacking through the assembly area and Hamlet) are themselves tied to domain-general issues that appear in the system's Curriculum (e.g., the principle of "surfaces" and "gaps"), which can in turn often be tied to the system's main themes of tactical reasoning (e.g., make effective use of resources). Based on our observation work we identified four categories of Dialog nodes:

- Set-Pieces: These are mostly one-sided discussions—more like Presentations—in which the tutor has something to say, and just says it. There may be some limited pauses for student input (e.g., confirmation, or opportunities for questions), but they do not materially affect the course of the Presentation. Examples include initial scenario briefings, and "war-story" anecdotes intended to make a particular point.

- Static Fact Tree Dialogs: These are tree-structured interactive Dialogs, in which the goal is to get the student to see and acknowledge some fact about the situation that the tutor believes they may be missing. The tutor prompts (perhaps repeatedly in different ways) to get the student to state the key fact. If the student doesn't give the desired answer, then the system walks a level down the Dialog tree and tries to get the student to see/acknowledge a set of more specific facts that together constitute a good argument for the higher level target fact. Again, the system may have alternate arguments it can try in turn in its effort to get the student to reach the desired conclusion. Examples include arguments for why taking control of the bridge must be a top priority.

- Enumeration Dialogs: At the top level, these are list-structured Dialogs (however, each item in the top-level enumeration may have a static fact tree of arguments beneath it). The object is to get the student to generate (e.g., brainstorm) alternatives for some decision or situation interpretation point. Examples include enumerating courses of action that will achieve control of the bridge, or ways to maneuver the scouts out of the way of a battle at the assembly area.

- Pro/Con Dialogs: At the top level, these Dialogs are structured as two lists of arguments—those in favor of and those against a course of action or situation interpretation (again, each item at the top-level may have a static fact tree of argument beneath it). The object is to get the student to generate as many of the pro and con factors/arguments as possible. Examples include evaluating courses of action intended to achieve control of the bridge, or evaluating alternate orders that might be given to the scouts to get them out of the battle at the assembly area.

Instructional Model

The ComMentor project was intended to explore the possibilities of automating individualized instruction in battlefield command reasoning and for reasons reviewed earlier—and elaborated with more background in Klein Associates, Inc.'s companion report on their ComMentor project subcontract (Ross, Battaglia, Hutton & Crandall, 2003)—we focused on Socratic tutoring in the context of tactical decision games. Having observed a large series of live eTDG tutoring sessions, the resulting transcripts were used not only to develop ComMentor encodings of tutoring for the specific Scenarios covered, but were also analyzed for what they could tell us about appropriate tutoring strategies for tactical thinking in general. The Klein team supplemented the observations and transcript analysis with a variety of supporting activities including tutoring session debriefs with the SMEs, and card-sort activities to help categorize constructs culled from the resulting data. All of these activities are thoroughly documented in their report.

Here, we start with an extract from that report that provides useful context to frame the discussion of instructional strategies:

> *How can a mentor or tutor best facilitate cognitive development in tactical thinking? A series of research studies looked at what mentors and tutors actually do in natural environments as part of on-the-job-training (OJT) to communicate their expertise*

to less experienced performers (Crandall, Kyne, Militello & Klein, 1992; Zsambok, Crandall, & Militello, 1994; Zsambok, Kaempf, Crandall & Kyne, 1996). The researchers studied the practices of experts who were identified by peers and superiors as highly skilled in OJT and/or one-on-one tutoring. This line of research has identified a small set of practices that are critical to successful one-on-one learning (Crandall et al., 1992). The set includes:

- *The ability to pass on expertise (both knowledge and skills) that is not available from training materials and that is not easily acquired except through interaction with someone more expert than the trainee. The expert has to be able to talk about what he or she has seen and done, not just reiterate textbook knowledge.*

- *Successful tutors possess an inventory of methods to assess trainees' declarative knowledge, technical skills, and response to feedback through observation and query of the trainee. Their assessment skills are highly developed and applied flexibly over the course of training or tutoring sessions.*

- *Expert tutors possess a range of instructional practices including modeling, interpreting events, directing the student's attention, providing feedback, and setting reasonable goals.*

- *Expert tutors form expectancies as they plan and conduct training, and use their expectancies and assessments of the student to guide the content and pacing of training.*

- *Interpersonal skills such as providing guidance and support, lowering anxiety, and motivating the learner are important to success.* (Ross, et al., 2003, p. 15.)

In this project we were looking for a more detailed inventory of instructional strategies specifically relevant to Socratic tutoring of battlefield command reasoning skills. The ultimate goal is to inventory a comprehensive range of strategies, identify the conditions under which any particular strategy might be appropriate, and characterize the required context, input, and processes for an algorithmic implementation of each relevant strategy. In this project we did not reach that ultimate goal, but we did advance the work of building the inventory and beginning to characterize how such strategies might be deployed. Also, despite lacking a complete theory of when and how to deploy these strategies, we were able to integrate a sizable number of them into our automated tutoring system.

... A column labeled "Instructional Strategy" was inserted into the transcripts. Researchers examined each exchange between the tutor and the instructor for each of the 24 [Phase II] *sessions to determine whether a specific instructional strategy could be named. When possible, an instructional strategy was named, and the portion of the transcript indicating the strategy was noted. ...* (Ross, et al., 2003, p. 6)

In Wave 1, we used the 57 instructional strategies identified by OJT coaches as a starting point to identify what strategies were being used in the sessions. Our goal was to build an inventory of instructional practices used by the expert tutors in this project, and

to understand when and how they are employed to help students integrate their knowledge into mental models in tactical thinking and use those models to make judgments and predictions.

> *The instructional strategies list expanded to include 131 descriptions of instructor interactions with the student as researchers added strategies during analysis of the transcripts. We were surprised at how many strategies were actually being used. On the surface, our observation prior to the use of this list for analysis led us to believe that there were few strategies in use. As we dug into the interactions, more strategies emerged than we expected, and we added to the original list. Some strategies in the original list were too broad to capture the kind of interaction taking place in the session, and as the researchers suggested new strategies to compensate, some redundancy grew in descriptions across the expanded list of strategies. To reduce and refine the list of strategies, the researchers performed a card sort of all 131 strategies, which also served to group the strategies by similar function. Performing this sort enabled the researchers to name and define the functional categories of strategies while reducing redundancy in strategy descriptions. There were also strategies listed in the original list of 57 that were not used in coding and subsequently discarded. The 131 strategies were reduced to 73 strategies organized by 11 functional categories.* ... (Ross, et al., 2003, p. 32.)

Our review of Klein's analysis in the context of the evolving ComMentor implementation led us to condense the list slightly, recognizing 61 distinct strategies, and to develop an alternate organization that emphasized the status of the identified strategies with respect to ComMentor. Where Klein emphasized cognitive or interaction operations, we cluster the strategies here by their relationship to ComMentor's mechanisms. We ended up with four categories, each of which is discussed with its assigned strategies in a following sub-section:

- Wired-in.
- Authored-in.
- Inappropriate.
- Out of scope.

Wired-In. By "wired-in" we mean that these instructional strategies are at some level built into the concept of ComMentor and its normal flow of control. We categorize strategies as either heavily, moderately, or lightly wired-in. In all we classified eighteen strategies as falling under these headings. Starting with the heavily wired-in strategies we have the following:

- Observe – ComMentor starts out by setting up a problem and then asking the student to react. A large part of what the tutor does is to observe what the student does in response to the problem (what orders they issue) and to match those against its repertoire of Evaluations.

- Identify incorrectness/missing pieces in student performance – Evaluation patterns can be set up to look for the absence of expected behavior, and/or for the presence of specifically predictable common mistakes. Having identified the issue, the tutor can proceed in any number of ways, e.g., apply many of the strategies described here.

- Guide student in task performance – ComMentor structures each TDG into a series of Scenes, and within each Scene generally maintains the overall initiative as to what will be done or discussed next.

- Instructor identifies critical tasks for student – This can be viewed as a variant or extension of the previous strategy. ComMentor's Evaluations, the structure of its Dialog trees, and its mechanism for tracing through Dialogs all focus the student on what the system believes to be the most important aspects of the task at any given moment.

- Redirect student focus to concept, scenario content – Again, ComMentor maintains primary control of the discussions topics, including topic shifts and choices to deepen discussion on a particular issue. Through its questions, it is responsible for directing the student's attention to the concepts or scenario factors that it considers most important.

- Give hints prompting – ComMentor repeatedly asks questions intended to get the student to see and then say key facts about the situation. At any given Dialog node, it may have a series of alternate ways of asking essentially the same question, generally with each succeeding version hinting more strongly about how the expert thinks the student should be thinking about things.

- Break material down into smaller pieces – If ComMentor cannot elicit a target response from the student after a couple of tries, then it starts to work through a set of nested Dialog nodes that expand the higher-level node into a more explicit argument structure.

- Elicit synthesis of facts/assumptions – When ComMentor returns from a set of nested argument expansion nodes, it again asks the student to consider the higher-level issue.

- Summarize/integrate discussion/significance – When ComMentor closes out any node, it generally presents some summary discussion of the point of the node. If a set of nested nodes was explored, this also functions as a kind of synthesis of the argument and the contextual significance of the nested factors.

- State required response/assessment – In the worst case, when presenting a summary, ComMentor may need to state it in such a way as to imply the student must accept the tutor's view in order to move on with its argument about the situation. This was a move that the human tutors generally tried to avoid making because it tends to cause students to lose their commitment to thinking about the Scenario.

Next we consider the <u>moderately wired-in</u> strategies:

- Elicit graphical depictions – The situation map visualization dominates the ComMentor screen, and manipulation of that map is the natural way to initiate order entry, and to fill certain critical information into various forms. We are not yet doing as much as we might with student-drawn map annotations such as control lines and regions.

- Elicit more detail on previous response – Under one interpretation this strategy fits with the descent into nested Dialog nodes (as cited under "break material down into smaller pieces"). Under another interpretation, this is addressed by ComMentor's mechanism of refinement questions that can be used to fill in gaps or correct possible misstatements in what the student said.

- Pose closed-ended questions/rhetorical confirmation questions – The final question in each Dialog node is generally a closed-end yes/no question that often leads the student fairly directly to the desired response.

Finally we consider the lightly wired-in strategies:

- Encourage student to interact with training mechanism – There is little to say about this strategy from Klein's list, as ComMentor's entire purpose is to try to make TDGs more available and more valuable to junior officers so that they will spend more time getting tactical training.

- Show/indicate information on the map – The situation map and the Presentation system were designed to facilitate the use of custom graphical layers to help communicate the tutor's points.

- Elicit student questions – ComMentor provides a menu of possible student questions that is available at all times. It would be a small and useful tweak to have ComMentor encourage student questions immediately after the initial briefing.

- Respond to student questions with a fact – Among the questions ComMentor allows a student to ask are a number of basic factual questions about the performance of various technical systems (e.g., weapons ranges, vehicle speeds, and so on).

- Elicit student summary/critique of own performance – At the end of each ComMentor Scenario, the system closes with a version of the reflective sessions we observed in our live tutoring sessions. At present, it is a somewhat weak approximation, for while it can ask the right questions, it cannot really do anything about interpreting or responding to the student's answers.

Authored-In. There are nineteen other instructional strategies that while not directly built into ComMentor's structure or its normal operation, are nonetheless easily and naturally expressible given the structures the system provides:

- Query about domain/scenario facts – ComMentor Dialog nodes are set up so the author can have the system ask students questions that advance their understanding of the

Scenario. While top-level dialog nodes generally address some thematic issue, lower-level (supporting) nodes often address specific domain/scenario facts.

- Tell/describe/lecture/restate domain/scenario facts – ComMentor almost never uses this strategy as a first resort—its Dialog nodes generally start out with questions—but it often resorts to telling either to summarize a line of questions/argumentation, or to fill in the expert's favored answer when students do not arrive at it themselves.

- Highlight relevant situational factors – Since the author decides what Dialog nodes to introduce, and is trying to help the student improve their understanding/reasoning about the situation, chances are many of the nodes will bear on relevant situational factors (though not all, since nodes may also bear, for instance, on high-level themes or typical student misconceptions).

- Offer additional info/factors – Discussion of this strategy depends on which word you highlight. If you focus on "offer" then the key observation is that though Dialog nodes generally start by asking questions, they can serve the function of offering additional information either because the question itself suggests something the student has not yet mentioned, or because (perhaps after a set of nested attempts to help the student build the answer), it offers a summary statement that contains its answer. If you focus on "additional" then the key observation is that Dialog nodes can bring up topics that have not yet been introduced in earlier discussion (or under a reading of "additional" that contrasts with the previous strategy, can, if the author so chooses, deal with topics that are not even relevant, e.g., irrelevant factors that are implicated as part of typical student misconceptions about the scenario).

- Pose alternatives/options – Unlike previous strategies, this one focuses not so much on information or factors but on things the student might do in the situation. Where information introduced in the previous strategies is likely to combine to form support for higher-level conclusions, the options considered here typically need to be explored and evaluated in greater detail (needing more elaboration through nested Dialog nodes). The pattern designed for enumeration Dialog nodes in which a set of child Dialog nodes each seek to elicit a particular option is one way to implement a version of this strategy that keeps the student engaged. If the tutor wants to be more assertive about posing alternatives it can use nodes that do not adopt the question form.

- Suggest contingencies to get student reaction – This is a specific way of considering alternatives—in this case, likely the enemy's alternatives rather than the student's. The reaction that is desired is generally to introduce more possible courses of action, or to arrive at an elaborated student plan that is more robust to (plausible) contingencies.

- Identify anti-goals (what can go wrong with COA) – Where the previous strategy focuses on situations that may arise beyond the student's control (e.g., due to actions or inaction by other forces, enemy or friendly), this strategy focuses more on the possible intrinsic failings of a particular COA—e.g., internal friction, risks of fratricide, etc.

- Remind student of task within bigger picture – This strategy might simply imply ensuring that the tutor always resurfaces, once it finishes exploring supporting details, in order to refocus discussion on higher-level issues. However, in the context of the TLAC themes it also implies a more specific reasoning strategy that explicitly tries to connect the student's Scenario task or mission with the large-scale factors at play in the Scenario.

- Compare/contrast red/blue thinking/capabilities/mission – This strategy is another one that relates directly to TLAC themes. It calls for the author to structure a particular kind of Dialog tree with parallelism that reflects the red *vs.* blue situation, and potentially includes a third thread for explicitly drawing the contrast.

- Get student to de-center by role-playing red/higher – When a student seems fixated on a particular (limited) conception of a situation rooted in their viewpoint on the problem (one that puts them at the center of the interpretive framework), it is often useful to ask them to take a different perspective—that is, to de-center them. While in general, ComMentor does not support "role-playing" (in the sense of playing through a temporally extended simulation) it is possible to set up Dialog nodes that ask the student to role-play another relevant decision-maker in the current (or in some hypothetical) context.

- Reframe situation/question student's framing – There are things that a student can do that suggest they are fundamentally misunderstanding important aspects of the problem. Evaluations that look for such patterns (and potentially nested Evaluations that use explicitly elicited rationale to confirm the initial diagnosis) can be used to directly address the incorrect framing. The strategy of role-playing red or higher is an example of how the tutor might help the student to reframe the problem.

- Identify flawed assumptions – If an assumption can reasonably be inferred from things that a student is likely to say, or from things that can be elicited from them in response to a specific line of questions, then an Evaluation or Dialog node can be created to address that assumption.

- Correct student assumptions with fact – Once an assumption is either explicitly identified or inferred (as in "Identify flawed assumptions" above) the Evaluation or Dialog node can work through the logic of why the assumption is likely to be faulty.

- Elicit student predictions regarding cause/effect – This is a useful strategy to get students to apply their developing knowledge about the domain (helping to avoid inert knowledge), and to try to arrange things so that they buy into further discussion about contingencies in the Scenario (having themselves identified issues that are likely to arise). ComMentor Dialog nodes can query the student to describe future states of the world, and given limitations on student input and system interpretation, do well to frame the requested effects in the context of some particular causes.

- Use examples, analogs, historical cases – This is a proven strategy across the spectrum of professional-level training. In the current ComMentor implementation, descriptions of, or references to, past cases can be embedded in any Presentation (e.g., Dialog node

summaries). In the future, such discussions can be packaged as Presentations of their own and potentially attached to Curriculum points, thus becoming available to be drawn into the discourse on any Scenario that touches on that point.

- Tell assessment, assumptions, implications, intent – Dialog node summaries can, at the author's option, be explicit about how the author sees the situation and the student's performance relevant to that node.

- Give reasons why on comments/assessments – Expanding on the previous strategy, Dialog node summaries can contain justifications for the author's view of the situation or student performance.

- Tell synthesis of facts and assumptions leading to an assessment – Again, elaborating the previous two strategies, the summary for a node that itself dominates a tree of Dialog nodes constituting an argument for some point can summarize that argument as it lays out the author's view of the situation. The summary can also draw in facts from sibling nodes earlier in the discussion tree.

- Query for revised COA, rationale, etc. – The author can identify places in the discussion where it would be natural to expect that the student would have changed their minds about something they said earlier, and offer them the chance to revisit the issue and say something different this time. There is a pair of outstanding technical issues that make it impossible to implement this strategy in its full generality in ComMentor, but it is supportable in limited forms even in the system's current state.

Inappropriate. A dozen of the instructional strategies we observed were judged to be inappropriate, either for a Socratic tactical tutor (because they were too directive or didactic) or for a computer tutor (because they were too interpersonal, or required too much intelligence and creativity). First we consider strategies that do not seem to fit with the notion of a Socratic tactics tutor:

- Give tests – While it is certainly true that any ITS will implicitly be evaluating and scoring the student (often on a continuous basis), the point of most ITS, and certainly a Socratic ITS, is to structure a situation that emphasizes learning and practice in thinking, not to make the student self-conscious and fearful of the least mistake.

- Give panic button – There should never be a reason for the student to panic. They are not performing a real-time, or even a real-world, task with serious consequences. They are playing a tactical decision-game and practicing thinking about tactical situations in new ways under the guidance of an interactive tutor.

- Allow student to make mistakes/see consequences – In contrast to many tactical role play simulation/tutor combinations that Stottler Henke has worked on, ComMentor does not emphasize the real-time unfolding of events (and hence consequences). Rather it emphasizes discussion and analysis. The tutor may suggest possible consequences of a course of action (or get the student to work them out), but there is no simulation in which the consequences manifest themselves so that the student can "see" them.

- Get student to suggest how to do the task – ComMentor does not focus on the procedural aspects of battlefield command and so is not concerned with how the overt tasks associated with the TDG situations might be accomplished. It is concerned with battlefield command reasoning, but it is difficult to characterize how reasoning unfolds, and probably futile to ask a student to introspect and externalize the mental steps they are executing. Rather, the tutor implicitly suggests (essentially models) reasoning steps that ought to have been carried out through the series of questions that it asks.

- Explain why trainee needs to do it a certain **way** – ComMentor tries to follow our SME's explicitly stated principle that the tutor should never try to cram any conclusion down the student's throat if they do not want to accept it, and that includes not forcing them to structure their plan one way or another. However, ComMentor is much more limited than a human expert, and so there will more frequently be times when the system runs out of options for how to convince a student of something, and/or does not have enough knowledge to deal with the consequences of the student sticking to a plan that the SMEs thought they could surely be talked out of.

The strategies that were deemed inappropriate for a computerized tutor were put off either because they assumed a tutor capable of too much emotional understanding and interpersonal connection, or because they required too much intelligence and creativity. The four strategies deferred on the basis that the tutor could not establish and maintain an empathic interpersonal connection with the student might be revisited, particularly if (a) the tutor's assessments prove to be highly accurate most of the time, and/or (b) it proves possible to create a tutoring authoring style that puts the human SME who created the instruction sufficiently in the foreground that students come to feel a connection with that (physically absent) expert.

- Elicit student worry, stress, fear – This is probably the most emotionally interpersonal strategy of all those considered. A computer cannot credibly ask students to express their negative emotions because they are (correctly for now) not perceived as having any emotions of their own, and thus as lacking in ability to understand, empathize, or make any use of the information. In fact, it might be possible for a tutor to make use of such information if the student expresses concern about something that can be linked to a Curriculum point.

- Display confidence in student's ability to perform – Again, a computerized tutor might have a basis for making confidence judgments about student performance on issues that can be tied to a Curriculum point for which it has already accumulated some student modeling data, but we do not think that a computer tutor is yet sufficiently credible to deliver encouragement that would be accepted as meaningful by a student.

- Praise – As with displays of confidence, offerings of praise are likely to seem out of place coming from a computer. We do encourage authors to include some amount of valanced feedback (e.g., in each Dialog node's "success summary" and "failure summary") but we do not imagine that rising to the level of extended overt praise.

- Use humor – Humor can be used effectively in instructional settings, and ComMentor's authors are free to attempt to author humorous Presentations for the tutor. However, even person-to-person, humor does not always work, and computer-to-person it seems more likely to fall flat or be delivered in a slightly wrong context.

- Pose open-ended questions – Computers are not particularly good at interpreting human responses to truly open-ended questions. Of course open-endedness is a matter of degree (as are most of the empathic/emotional techniques just considered) and ComMentor can handle partially open questions—there are examples in the existing Scenarios of what and where questions, but probably not why or how questions and certainly not wide-open questions like "So what do you think?"

- Open-ended question following closed-ended question – Built into ComMentor's basic operation at each Dialog node is a progression from more open to more closed versions of essentially the same question, which is the opposite of this strategy. ComMentor's basic loop does lead to shifts from questions on more specific topics to questions that pop up to address a higher-level issue, but the question asked when popping up is generally a closed-ended (yes/no) question.

- Innovate/tailor session script or procedure – Computers are not especially good at innovating or moving beyond their scripts. ComMentor does however tailor its scripted responses in the sense that it chooses which parts of its script to pursue with any given student based on their responses.

Out of Scope. Another dozen strategies, while in principle relevant to ComMentor's goals were judged to be out of scope for the current effort:

- Instructor moves assets for student (interprets COA) – In order to make the user interface as fluid and usable as possible, we had to turn over control of the main force icons on the map to the student (as a means for partially communicating their taskings). This is contrary to the observed and stated practices of our SMEs who preferred to have students work verbally and take responsibility for most map updates upon themselves as a way of reflecting back possible misinterpretations of what the student said.

- Model or watch me – As already noted, ComMentor is not primarily about playing through a situation; it is about talking through a situation.[3] This limitation suggests that the next option is more relevant.

- Modeling thinking aloud – Having the tutor talk about an expert version of thinking through a situation fits better with ComMentor's focus, but is not part of the classical Socratic approach, and thus was ruled out of scope for this project.

[3] Stottler Henke is interested in combining tactical role-play and Socratic tutoring, and is actively promoting the idea of a Socratic tutor for pre-play planning and after-action review, with simulated role play (potentially with active coaching) in the middle.

- Think aloud while demonstrating – This is an augmented version of the previous strategy, and was ruled out of scope for similar reasons.

- Model/think aloud red/higher – Having the tutor model relevant decision-makers other than the student is an interesting idea. There are simple versions of this strategy that can easily be embedded within normal Dialogs. However, we did not have time to pursue the general strategy.

- Role-play red/higher/subordinates – This strategy asks the student to take the position of other relevant decision-makers. Again, there are simple versions of this strategy that can be embedded in normal Dialogs. More elaborate versions would escalate into full blown role-play, which as noted was judged out of scope.

- Correct on specific behavior in performance stream – The only examples of this strategy that ComMentor attempts are prompts for thoroughness during the student's issuing of initial orders (e.g., tasking all major combat units, and ensuring that taskings include intent statements). These first attempts have not been particularly satisfactory, so we are unlikely to proceed further in this direction for now.

- Allow student to start again – On some occasions, a student's initial response to the problem was badly enough flawed, and early discussion revealed enough factors that it became clear they were ready to reconsider their actions. In such cases the tutors often offered the student the chance to start over and submit a totally different plan. This would be a good feature to add to ComMentor, but it simply did not make it into the current implementation.

- Offer agreement in student thinking/performance – We think of this strategy as a variant of the earlier "praise" strategy that differs in its emotional impact (it is, perhaps, less manipulative, though also sometimes less satisfying). Again, authors are encouraged to include positive and negative feedback in the success and failure summaries at each Dialog Node, but that is the current limit in applying this strategy.

- Repeat instructor question to get definitive answer – The closely related strategy "Elicit more detail on previous responses" was categorized as wired-in (moderately). While ComMentor can ask follow-up questions and more leading rephrasing of questions, it never actually asks the same question over again, and it is not sensitive to hedges in the student's answer (which are not even expressible in the templated forms).

- Rephrase student response/question to confirm/clarify – While ComMentor has some mechanisms for pursuing resolution of incorrect, incomplete, or irrelevant answers, it does not seek to clarify questions (which can only be asked in fully specified form), nor does it pursue clarification by rephrasing what the student says. That would require more complex language processing than could be supported.

- Evaluate student's questions – It would be possible to allow student-initiated questions to be tied to Curriculum points, and to use the asking of a question as the basis for updating

the Student Model with respect to those Curriculum points. We have not explored this approach in the current implementation.

Domain Model

ComMentor's domain model is a large, but relatively simple ontology of military concepts. The ontology is large in that it contains more than 1200 concepts covering categories such as (a) military units, (b) vehicles, (c) weapons, (d) locations/battlefield-geography, (e) basic physical quantities and units, (f) military events and relationships, and (g) key military analytic categories. The ontology is simple in the sense that it includes almost no inference rules (axioms), and does not attempt to provide necessary and sufficient definitions for concepts (e.g., as in typical description logics). The ontology is built on a knowledge model similar to that provided by Protégé, on which it is loosely patterned. It supports a type graph (with both upward and downward branching) as well as metatypes, and allows definition of attributes/properties (slots) with filler constraints for types (and metatypes) that can apply either to those types or to their instances (and can be inherited down the type graph).

The content of the ontology derives primarily from three sources: (1) pre-existing ontologies, such as DARPA's High Performance Knowledge Bases (HPKB) ontology, and MITRE's Command and Control Simulation Interface Language (CCSIL) representation, (2) new analysis of sources such as the ComMentor tutorial session transcripts and U.S. Army FMs, including FM 3-90 in particular, and (3) input from the ComMentor project SMEs. Some of the content, and much of its form, was ultimately reshaped by the needs of the first ComMentor scenarios—the need to support expression of particular kinds of student inputs (e.g., to explain the tactically significant relationships between points on a map). We believe many pieces of the resulting domain model will be applicable to other military training applications. For instance, we are currently in the very early stages of a case-based instruction project for the U.S. Navy that emphasizes analogical reasoning, and we expect that ComMentor's domain model will serve as a useful base for that work.

The ontology exploits its upward branching typology to allow multiple forms of organization. Some classifications reflect properties of concepts that may be used to support inferences or system behavior; for instance a classification of control measures according to whether they represent points, lines, or areas can be used to help determine their layout on maps. Other classifications are tuned to provide choice lists for users to pick from in the user interface; for instance an organization of control measures according to their major functional areas (such as movement control versus fire support control).

Figure 4 presents a selective view of the top levels of ComMentor's military domain ontology. Even in this small sample, it is apparent how concepts can appear in more than one place (e.g., MilitaryThing appears both under TangibleStuff and under MilitaryStuff). Given the focus of this ontology, there is very little under IntangibleStuff or TangibleStuff that does not have military significance. For instance, most physical objects defined in the ontology are those that matter as MilitaryThings (and are so classified). The major exceptions are large-scale physical objects that can reasonably be interpreted as places, and thus potentially as battlefield features or targets. Many of the items under IntangibleStuff are so abstract as to be useful in the

military world, though not exclusively so (e.g., Dimensions of Length and Time with accompanying MeasurementUnits, attributes, and Comparison).

IntangibleStuff	TangibleStuff	MilitaryStuff
Happening	Agent	MilitaryThing
State	Person	BattlefieldGeometry
Action	Group	MilitaryHappening
MilitaryHappening	Location	MilitaryRelationship
Dimension	Place	MilitaryDimension
MeasurementUnit	BattlefieldGeometry	MilitaryValue
Value	MilitaryThing	MilitaryKnowledge
Comparison	Obstacle	MilitaryComunication
When	Target	
	MilitaryUnit	
	MilitaryInstallation	
	MilitaryGood	
	Supply	
	Vehicle	
	Weapon	
	Ammunition	
	Sensor	
	CommGear	

Figure 4. Top-Level Categories in ComMentor's Domain Ontology.

Focusing on the MilitaryStuff, in addition to MilitaryThings and BattlefieldGeometry descriptors, there are specialized Happenings (militarily significant States and Events, especially military tasks and the intents they are supposed to achieve), Relationships (including command and control relationships), Dimensions and Values (e.g., Echelon, Formation, DamageLevel, etc.), a breakdown of MilitaryKnowledge (e.g., Doctrine and Technique, Intelligence and CombatInformation), and MilitaryCommunication forms (e.g., the underpinnings of forms intended to capture standard communication formats such as orders and reports).

Curriculum Model

The general curriculum for ComMentor was set by a combination of ARI's expressed goal of developing battlefield command reasoning instruction, the example "Think Like a Commander" (TLAC) materials they made available, and the experience and practice of our major consulting SMEs. The specific curriculum was allowed to emerge, in an inductive fashion, from the particular scenarios and tutor/student interactions that we observed during the course of the project.

We ended up organizing the curriculum according to two different schemes. The first was rooted in the TLAC themes, slightly expanded to cover additional issues to which our SMEs devoted considerable time. The second started from an abstract task-based breakdown of an officer's reasoning processes, and moved at the next level to an analysis based on work products and attributes of those products. Figure 5 shows the top levels of both those breakdowns. In all, there are currently over 150 Curriculum points defined in the system.

TLACThemes	GeneralTasks
KnowTheFacts	CommunicationTasks
CommunicateEffectively	IssuingOrder
FocusOnMissionAndHighersIntent	IssuingReports
ConsiderTheBigPicture	IssuingRequests
ConsiderEffectsOfTerrain	AnalysisTasks
MakeEffectiveUseOfResources	AnalysisOfSituationAwareness
ModelAThinkingEnemy	AnalysisOfTerrain
PrioritizeAndTakeActionInTime	AnalysisOfForces
PlanFlexiblyAndForContingencies	AnalysisOfImpact
VisualizeAChangingSituation	AnalysisOfTerrainEvolution
	AnalysisOfForcesEvolution
	AnalysisOfAlternatives
	PlanningTasks
	PlanRelevance
	PlanEfficacy
	PlanEfficiency
	PlanFlexibility
	StandardPlanComponents
	TimeManagementTasks
	TimeAllocation
	TimeSequencing
	EventCoordination
	ContextualTimeFactors

Figure 5. Two Top-Level Breakdowns of ComMentor's Curriculum.

As a complement to our inductive work, we wanted to get some sense for formal tactical training in the U.S. Army. We reviewed the West Point course of study and listings for their Department of Military Instruction. The requirements and offerings at West Point suggest the kind of tactical training the best newly commissioned Lieutenants are likely to have had (a scan of randomly selected ROTC programs suggested that, as expected, their offerings are more limited). The total number of courses offered by West Point's Department of Military Instruction is not large. Not counting the required sequence of four courses (MS 102, 202, 302, & 402) offered during intersession, there are only 5 electives in the third year, and 4 in the fourth year (including an independent study course and a colloquium series). Two of the core courses appear to have significant tactical content. Two of the electives also appear to focus on the practice of tactics. The relevant courses are listed below with their synopses from the catalog.

- MS 102: Ground Maneuver Warfare I – Cadets will investigate fundamental warfighting principles. These will include Ground Maneuver Theory, Elements of Combat Power, Principles of War, and the Factors of METT-TC. Using narrative text methodology, the course will use accounts of warfare from text and film to illustrate these fundamentals. Then, using the current light infantry squad and platoon as a venue, cadets will apply the fundamentals in tactical decision vignettes and a computer simulation.

- MS 302: Ground Maneuver Warfare II – MS 302 educates 2nd Class Cadets in the essence of ground maneuver warfare; further evaluates the officership concepts of

leadership, teamwork, and competence; and inspires 2nd class cadets to a lifetime of professional military self-development. Using the phenomenon of combat power as a guide, the course will use a theory to practice model of learning. Cadets will learn the theories of past and current military thinkers; see the theories applied to historical small unit engagements; and apply concepts to tactical decision exercises.

- MS 310: Tactics – This course is designed to introduce the cadet to the tactical level of war with a thread of leadership, doctrine, and technology. The course is organized into three blocks. Block one provides a basic tactical foundation using U.S. Army and USMC tactical doctrine, including some discussion of the human and leadership factors of small unit engagements. Block two focuses on offensive and defensive operations concluding with three offensive engagements on constructive and virtual simulation. Block three is an application of knowledge learned in the first two blocks through practical exercise and briefings that concludes with a tactical battle or engagement analysis. This course makes heavy use of both virtual and constructive simulation including Delta Force 2 Land Warrior, M1 Tank Platoon 2, and the Janus Warfighting Simulation.

- MS 365: Structuring Operational Forces (Warfighting with Training Simulations) – This course examines the integration of Light (dismounted forces) and Heavy (mechanized forces) at the tactical and operational levels of war. Cadets will study the underlying factors that have influenced U.S. force structures from World War II to the present. They will study the Fundamentals of Army Operations and the current U.S. Army Divisions' structures and organizations. Cadets will utilize the Military Decision-Making Process to apply their knowledge of structure and doctrine to tactical situations. Cadets are challenged to test their plans and decision-making skills utilizing Army computer simulations. Finally, cadets will examine possible force structures and doctrine for the 21st century Army and analyze them with computer simulation. (USMA Redbook, 2001)

Over four years, it appears that a diligent student of tactics would have exposure to a handful of tactical decision vignettes and exercises, and work with constructive and virtual simulations to a limited extent. Starting at this entry level of Army officer training, ComMentor should be able to significantly improve the situation by making mentored tactical training scenarios more widely available to motivated students.

Scenario Model

Of the six scenarios studied in detail during the Phase II project, three have been developed to a point where they can be run using the ComMentor prototype, though to varying degrees of completeness. In addition, the scenario that served as the focus for our Phase I work, "Tanks on the Farm" has also been minimally developed in order to serve as an example for ComMentor's supporting documentation (in particular the introductory video that accompanies the system uses the "Tanks on the Farm" example). In this section we briefly characterize the scenarios we worked with, and then give an in-depth feel for the encoding of one that we explored most extensively: "Enemy Over the Bridge" (used as an example earlier in "TDG Data and Preliminary Analysis" and again later in "Final Prototype Runtime Capabilities" and "Final Prototype Authoring Capabilities" below).

- "Enemy Over the Bridge" (EOTB) – EOTB is a conventional armor scenario set terrain meant to resemble the rolling hills of middle Europe. The student commands a tank/mech battalion moving to an assembly area in preparation for leading an attack west across a river, and finds the enemy has already started moving east across the same bridge they were to use the next day. Some of the major themes in this scenario include: (1) building a coherent Red story, including where their forces might be, and what they might be trying to do (i.e., we are probably seeing a battalion-sized advance-guard for a larger enemy offensive into our territory), (2) recognizing the key problem and the general nature of the necessary response (i.e., the enemy is flowing into our territory over the bridge we need for our mission, so somehow we have to take back control of the bridge to stop the flow and salvage our own offensive), (3) prioritizing actions based on that understanding of what is most important (i.e., don't spend all your resource fighting the enemy you happen to see in front of you rather than dealing with the bridge), (4) considering and evaluating a range of COAs for getting to the bridge, (5) understanding what it means to take control of the bridge, (6) thinking in some detail about the risks in the situation and the appropriate sequencing of activities, and (7) considering in some detail the uses of a reserve force in a fluid and uncertain situation such as this.

- "Attack on Kalat" (AOK) – AOK is an objective-force scenario, set in a mountainous desert like Pakistan. The student commands an FCS company stocked with a broad range of new (notional) capabilities, tasked with destroying the much larger conventional (and guerilla) covering forces protecting a town on the way to a major enemy-held center. Some of the key themes in this scenario include: (1) basic terrain and situation analysis, (2) planning an assault that satisfies higher's guidance and intent (i.e., be sure to destroy key threat assets, and don't let them escape to fight another day), (3) recognizing the FCS systems' strengths and vulnerabilities, and how they relate to enemy threats (i.e., worry more about guerillas with RPGs than about enemy tanks), (4) exploiting the potential of your new FCS reconnaissance assets, (5) matching your FCS firepower assets against the various assets of the enemy, (6) reconceptualizing concepts like assault, envelopment and depth in the context of new FCS capabilities, and (7) working out timing and sequencing to achieve overwhelming massed surprise effect.

- "Clash at Timpan-ni" (CAT) – CAT is another conventional armor scenario, this time set in terrain reminiscent of Korea. The student leads a mixed tank/scout cavalry troop guarding the western flank of a brigade advancing towards an enemy stronghold; they are attacked by a sizable coordinated set of enemy formations. This scenario was used during our limited authoring exercises. As a result, the basic introductory and supporting materials were prepared and entered into the system, and a start was made on roughing in Evaluations and fleshing out the first set of Dialog nodes. Some of the key themes in this scenario include (1) conducting a hasty terrain analysis for a very complex region, (2) analyzing the Red picture, (3) recognizing the implications of a changed situation for your unit and for higher (i.e., this is a big enough enemy that a cavalry troop isn't going to stop it, and brigade is going to have to deal with the serious threat/opportunity to damage the enemy), (4) recognizing the appropriate high level response (i.e., delay the enemy attack until brigade can maneuver to effectively counter it), (5) choosing among

high-level options for achieving the necessary delay, and (6) pursuing some detailed analysis of possible positions and timing to maximize delay and minimize risk.

- "Tanks on the Farm" (TOF) – TOF is a conventional force, armor scenario set in quite simple terrain (flat open land with one major hill and some forests), in which a pure tank battalion is surprised by the presence of an enemy tank concentration well inside what was supposed to be friendly territory. This scenario was used extensively during our Phase I work, and has not been significantly developed beyond that point during Phase II, other than to port its introductory and supporting materials to the new prototype, and to define enough tutoring content to enable its use as a small canned demonstration in the ComMentor orientation video (i.e., just a couple of Evaluations with a handful of Dialog nodes in total). The major themes are (1) recognizing the application of higher's intent in a changed context, (2) knowing the capabilities of your assets, (3) recognizing the value of quick aggressive action given the likely evolution of the situation, and (4) thinking about the bigger picture and how events might unfold in the longer term.

Scenario tutoring structures in EOTB. EOTB was the Scenario that was the best developed during this project. We worked with it throughout the evolution of our thoughts on ComMentor's runtime capabilities, its Scenario encoding style, and its Authoring Tool approach. Many of our ideas about how to achieve various kinds of tutor behavior were worked out in the context of EOTB, and so it is interesting to review aspects of its structure to capture various ways in which ComMentor Scenarios can be structured to achieve certain ends. To provide context, Figure 6 shows the expanded tree of Evaluations for EOTB as a screen shot from the Tutoring Authoring tool.

A review of these 22 Evaluations and the Dialog nodes they package reveals an interesting set of usage patterns. The following survey of observations represents a first step towards abstracting and organizing the authoring process and resulting tutoring structures. We start with eight observations about the uses of Evaluation nodes:

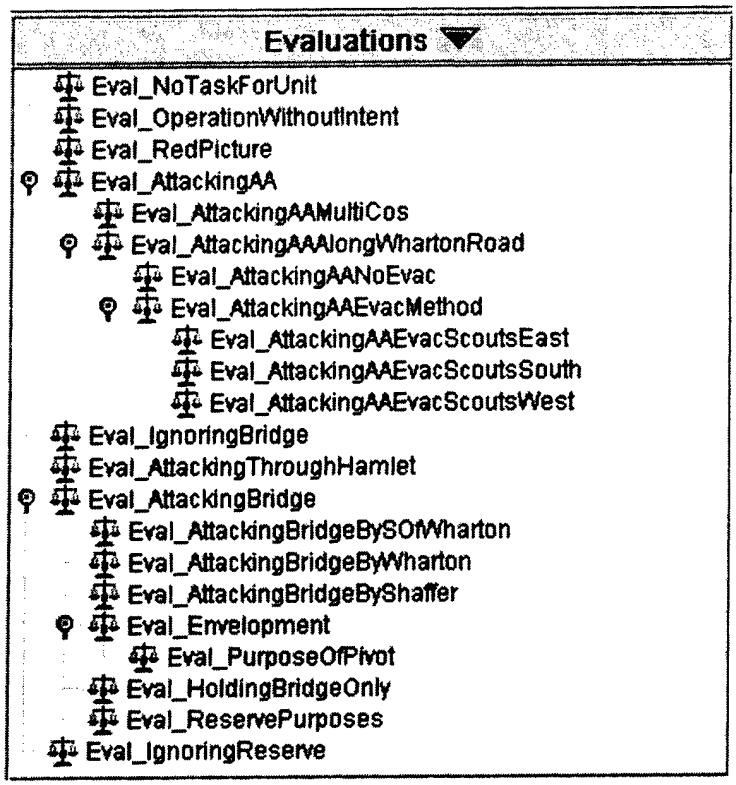


Figure 6. Evaluation Tree for "Enemy Over the Bridge" (EOTB) Scenario.

- Run-once vs. run-multiple Evaluations – ComMentor recognizes a distinction between Evaluations that should only be triggered once, and Evaluations that should be triggered as many times as there are ways to match its pattern. Most substantive discussions are only worth having once. The place where we found it useful to allow multiple executions of an Evaluation (and hence its Dialogs) was for relatively low level evaluations focused on the communications process itself: e.g., Eval_NoTaskForUnit and Eval_OperationWithoutIntent both critique the orders issued by the student, the former critiques the absence of an order for one of the major combat units (a violation of the TLAC theme of making efficient use of available resources), while the latter critiques the absence of an intent in an order.

- Evaluations triggered on absence of input – As suggested by the pair of example Evaluations just described, Evaluations can trigger not only on what the Student says, but on what they fail to say. Other, more complex examples include Eval_HoldingBridge, Eval_IgnoringReserve, the first of which triggers on the absence of taskings that put friendly forces on the far side of the river to hold the bridge (assuming some unit has been tasked with retaking the bridge), and the second triggering on the absence of a tasking that designates any unit as having the role of a reserve force.

- Run-always Evaluations – Some run-once Evaluations cover topics so central to the Scenario that they should be invoked no matter what the student says. There was not a single live tutoring session for EOTB that we observed where the tutors did not at some

point—and usually sooner rather than later—spend time talking about the Red situation. Accordingly, the trigger pattern for the Evaluation Eval_RedPicture is empty, which tells the runtime to queue up the accompanying Discussion no matter what.

- Nested Evaluations– Evaluations can nest beneath other Evaluations. The runtime interprets this construct to mean that the nested Evaluations should only be considered if the parent Evaluation has triggered. One example application of this capability is the cascade of Evaluations from Eval_AttackingAA, to Eval_AttackingAAAlongWharton, to Eval_AttackingAA_NoEvacuation. The first checks that the student has tasked some unit to attack the assembly area. The second checks that the student has tasked some unit to attack the assembly area by moving down the Wharton Farm Rd. The third checks that the student has not issued any orders to the Scout Platoon that tell it to go anywhere.

- Nested Evaluations triggered on accumulated bindings – Another smaller cascade illustrates a stronger form of coupling between a nested Evaluation and its parent. Eval_AttackingBridge checks that some unit has been assigned to attack the bridge, and binds variables to remember the tasking and the unit. The nested Eval_Envelopment checks that some unit has been assigned to attack the assembly area, and that the two taskings and two units are distinct.

- Evaluations triggered (and discriminated) on facts explicitly solicited by a Dialog node from parent/sibling Evaluation – A topic of discussion covered in an Evaluation node that diagnoses a problem with the student's plan may naturally culminate in the student recognizing the problem and wanting to issue new orders. Another set of Evaluations may be primed to react to the student's change of heart. As an example, the Eval_AttackingAANoEvac critiques the student for not moving the scouts out of the way of an oncoming assault, and its discussion ends with a final prompt to the student: "Why don't you give the Scouts some orders about what they should be doing--especially where they should go." The sibling Evaluation Eval_AttackingAAEvacMethod and its children are ready to respond to various orders the student may choose to give to the scouts. Due either to limitations on authoring time or to a desire to keep Scenario discussions more focused, this is not necessarily done every place it might be possible. For instance, Dial_ReservePurposes_AlternateSuccess_Position could potentially have been set up to collect student input that would then be used to discriminate among a set of Evaluations discussing alternate plans for the reserve forces, but it was not.

- Nested Evaluation triggered on facts established as normal part of some Dialog node from a parent Evaluation – A somewhat odd variant of the previous relationship between Evaluations is the case where the trigger pattern for one is something that the student is expected to say some time during the discussion of the other—where the fact is a normal part of the first discussion rather than an explicitly solicited transition. The trigger for the Eval_PurposeOfPivot is actually something that the student is expected to say some time during the discussion associated with its parent Eval_Envelopment. This is a peculiar usage, as there are other constructs that could accomplish essentially the same dynamic discussion flow making use only of features of Dialog nodes (see "Dialog node for soliciting options/possibilities/rationales" and "Dialog node for elicitation" with "Dialog

node for acknowledgement" in the list below). A possible advantage (probably not realized in this case) is that the nested Evaluation can potentially be triggered by input arriving in some other context including contexts after the end of discussion associated with Eval_Envelopment.

- Evaluations without Dialogs – Some Evaluations exist only to serve as parents to other Evaluations and do not themselves directly launch any discussion. Eval_AttackingAA is one such case. There is not a lot to say based simply on the fact that the student has decided to attack the assembly area (pretty much every student tends to do that). But seven other Evaluations nested under Eval_AttackingAA discriminate further and together control five different lines of discussion.

A Scenario contains many more Dialog nodes than Evaluation nodes. Not surprisingly then, there is more to say about different ways Dialog nodes can be used. The main use envisioned when Dialog nodes were first defined was to represent a claim about the Scenario (e.g., facts, situations, actions, with their meaning and justification in context). While that is still the main idea, there are many variations. Here we catalog fourteen observations about possible ways to use Dialog nodes based on usages in the EOTB Scenario:

- Dialog nodes with varying numbers of questions at varying specificity – The core structure of a Dialog node allows the tutor to ask a series of variant questions aimed at eliciting the same information (a basic strategy often observed in use by human tutors). The current design allows for an Introductory Question, a Leading Question, and a final Hinting Question (in practice, this last is often a yes/no question). Further, if the point of the Dialog node is in any way complex or based on inference, trees of nested Dialog nodes can be defined to lead the student through an argument for the main point (after the Leading question and before the Hinting question. Examples illustrate how these facilities may be used. Dial_ScoutsInWayOfAAAssault is a Dialog node that comes at the end of an argument sequence and only offers a single prompt before going to its summaries; this is probably because it seemed important to get to the conclusion that the sequence was building towards, and because there was concern that limitations of the system's input forms would make it difficult for the student to know how to say the right thing (so it was better not to belabor the point). In contrast, the node right before it in the sequence, Dial_ScoutsEngagedWEnemyInAA, provides two prompts before moving to its summaries. And at a critical point from an earlier discussion, Dial_RedOffensive uses all three of the available question slots to get the student to acknowledge that they are probably facing an enemy offensive.

- Simple fact Dialog nodes – The most straightforward kind of Dialog nodes are those that deal with simple factual issues such as the contents of a report from the scenario setup, or the size of a terrain feature (e.g., Dial_RedBattalion_MechCoInAA asks the student to recall and state what they were told was waiting for them in the assembly area, and Dial_AttackingAAMultiCos1SqKm asks the student to say roughly how big the assembly area is). These Dialog nodes essentially never have children. If the student cannot correctly state a simple fact, there is no point in breaking it down and arguing for the fact. The tutor simply goes ahead and states the correct answer. If the fact is simple enough,

the author may even choose to structure the Dialog node without any follow-up questions that offer the student extra chances to get the right answer.

- Dialog node straight to yes/no question – Some Dialog nodes deal with inferred facts, yet address them in a context where either they should be obvious or it is more important to get on to other things. In that case, a Dialog node may be structured with nothing but a yes/no question that leads directly to the final success/failure summaries. As an example, Dial_AttackingThroughHamlet_RedForces simply asks, "Are you expecting a concentration of red forces in the vicinity of Hamlet?" and once the student answers, offers the rationale for why they should be expecting such a concentration.

- Dialog node primarily for grouping – Many Dialog nodes exist only to group, control, introduce, or summarize sets of other (nested) Dialog nodes. A common pattern in which such utterance-poor Dialog nodes occur is in Pro/Con Dialogs (often themselves nested inside Option-Generation Dialogs). For instance Dial_RedBridgehead_Hamlet_Pro, is a Dialog node that covers the arguments in favor of the Red forces occupying Hamlet as part of establishing their bridgehead; it contributes only the setup-utterance "Being in Hamlet pretty much only has up-side for Red." and then dispatches to its nested Dialogs to actually discuss what the good points are. The same is also often true for the highest-level Dialog node assigned to any Evaluation. Again using a piece of the discussion controlled by Eval_RedPicture, the top-level Dialog node Dial_RedPicture has the tutor uttering the setup, "Let's talk a bit about the red forces and their situation." and then sequencing through a set of four immediately nested Dialog nodes.

- Dialog node for grouping also elicits post-argument summary/acknowledgement response – Some Dialog nodes that primarily group nested argument nodes (that do not start out with questions of their own) do not simply summarize the nested argument, but end up with a question intended to elicit a summary of acknowledgement of the argument. For instance, Dial_AttackingThroughHamlet ends with the question, "Would you agree that by going through Hamlet you're likely attacking through the thickest part of the enemy?" which summarizes the argument presented by its nested nodes. The question could not easily have been asked up-front in a non-hinting form.

- Dialog node for soliciting options/possibilities/rationales – A common structure for pieces of TDG mentoring sessions is for the tutor to ask questions that solicit sets of options, possibilities or rationales—options for the student's own actions/positions, possible enemy actions/positions, and reasons why various COAs might be good or bad. The current system is somewhat limited due to its input capabilities, and so uses this approach most effectively for issues that can be expressed by pointing to the map, i.e., to solicit possible positions and routes. Dialog nodes like RedBridgeheadLayout (exploring where Red might have forces) and Dial_HoldingBridge_Layout (exploring where Blue ought to think about getting forces) ask a question to solicit options and then rely on a set of nested Dialog nodes to respond to what the student said. Nodes like Dial_AttackingBridgeBySOfWharton and Dial_AttackBridgeByShaffer attempt to do something similar in soliciting rationale.

- Dialog node for elicitation (exit test only) – A structure that is sometimes used beneath Dialog nodes that solicit options/possibilities/rationales is a child-node that asks questions aiming to solicit a particular option from the set of likely responses. For instance, Dial_AttackBridgeByShaffer_HamletHill_Req is nested beneath the node Dial_AttackBridgeByShaffer; the parent asks where Red might have forces that could affect a particular Blue plan, and the child suggests one such location. Based on the standard processing of Dialog node satisfaction patterns, such a node is only used if the student does not suggest that place of their own accord.

- Dialog node for acknowledgement (entry test only) – This is a structure that is generally used in combination with the just discussed request nodes. Acknowledgment nodes like Dial_AttackBridgeByShaffer_HamletHill_Ack are paired with request nodes, but where the request node is only used if the target has not yet been said, the acknowledgement nodes are only used once the target has been said (by making the target an entry pattern). Acknowledgement nodes may actually be simple acknowledgements, or they may be grouping nodes that control elaborate discussions that assess the option once it has been put on the table by the student.

- Dialog tree sequentially eliciting a set of factors – Another, though less satisfactory, way of eliciting a set of factors from a student is to lead them through the issues by queries in a more sequential way. RedBridgehead_ForestsWBridge_Pro has three child nodes that attempt to elicit three particular reasons why a particular position might be considered desirable for Red. This is probably best thought of as an example of what not to do, as the request/acknowledgement structure is far more flexible and responsive to student input.

- Dialog tree sequentially discussing a set of factors – Yet another variant way to structure a discussion of a set of factors is to have the tutor simply introduce them in turn, leaving the question/answer aspects of the discussion for the deeper exploration of the issues (rather than for uncovering the issues). Two examples of this approach include Dial_Envelopment_Roles which says what the three major roles of an envelopment are and then asks what each of them is for, and Dial_ReservePurposes which again tells the student what the three major purposes for a reserve are, but then asks the student questions about how they apply in the particular situation.

- Dialog nodes set up as alternatives – Another Dialog node pattern can be used to make the system responsive to an expected range of contextual situations. Several nodes can be set up as alternatives for discussing essentially the same point, but doing it a bit differently depending on context. Dial_ReservePurposes_PreventFailure_EyesNorth and Dial_ReservePurposes_PreventFailure_EyesSouth both discuss the placement of a reserve force for purposes of watching an otherwise unwatched piece of terrain; the difference lies in which piece of terrain is known to be otherwise unwatched. These nodes have different entry conditions and different satisfaction conditions to ensure they are invoked under the right circumstances and satisfied in the right ways. The expectation is that only one of them will ever apply in any single student run.

- Dialog customization by use of variable bindings – The output of a Dialog node as seen by a student can vary according to the variable bindings established in the triggering Evaluation (or one of that Evaluation's parents). For example, Eval_NoTaskForUnit and Eval_OperationWithoutIntent both identify the unit in question and fold the relevant unit's name into the tutor's utterance in the associated Dialog node.

- Dialog nodes being reused – ComMentor allows the same Dialog nodes to be used in multiple places in a single Scenario. So far, this is rarely done, but it is a sensible thing to do if a given line of argumentation might be relevant in more than one context. The basic structure and rules of Dialog nodes can be used to ensure that the duplication in the Scenario structure will not lead to duplication in any one student's experience. Dialog nodes are generally defined with a pattern they are trying to satisfy and will only be processed if the target pattern has not yet been seen. Thus, whichever occurrence of a Dialog node comes up first will generally lead to a fact that satisfies the pattern being asserted, which will prevent the same Dialog node from running again. In EOTB, Dial_AttackingBridge reuses the five Dialog nodes originally authored in the context of Dial_IgnoringBridge since in either case it is worth discussing why the bridge is matters.

- Dialog node soliciting input that probably would come from earlier node – It may happen that a fact that is relevant in one discussion context is also relevant in another. Reusing Dialog nodes is one possible approach here. Another, of course, is to author a separate node that is looking for essentially the same (or a subsumed) fact. This happens in Dial_ReservePurposes_ReinforceSuccess_EndLocation where part of the discussion of the parent Evaluation of Eval_ReservePurpose would already have gotten at the question of where Blue forces should be placed to successfully hold the bridge. In the case where a satisfaction pattern is matched before the second (occurrence of a) Dialog node is considered, there ought to be a new kind of utterance associated with the Dialog node that could simply refer to and acknowledge the earlier discussion. This is not yet a feature of ComMentor's Dialog nodes.

Final Prototype Architecture and Design

In the Introduction to this report we included a brief overview of ComMentor's functional architecture. In this section, we cover the same territory in more detail, and also discuss aspects of the system's implementation architecture.

Functional Architecture. Figure 7 shows the complete functional architecture diagram (compare to Figure 1). Figure 7 includes significantly more detail about the Runtime Environment—notably through its inclusion of the main persistent data stores and associated processes. It also includes a new section for the Authoring Tool Suite showing the various tools used to stock most of the persistent data stores. In this diagram, rectangles represent processing modules, rounded rectangles represent the major in-memory data/knowledge structures, and cylinders represent persistent data/knowledge stores. Lines with arrows represent flows of control and data. The Runtime's major processing cycle is in heavier lines as in the original simplified diagram. All flows from the Authoring Tool Suite to the various persistent stores are drawn as dotted lines to distinguish them.

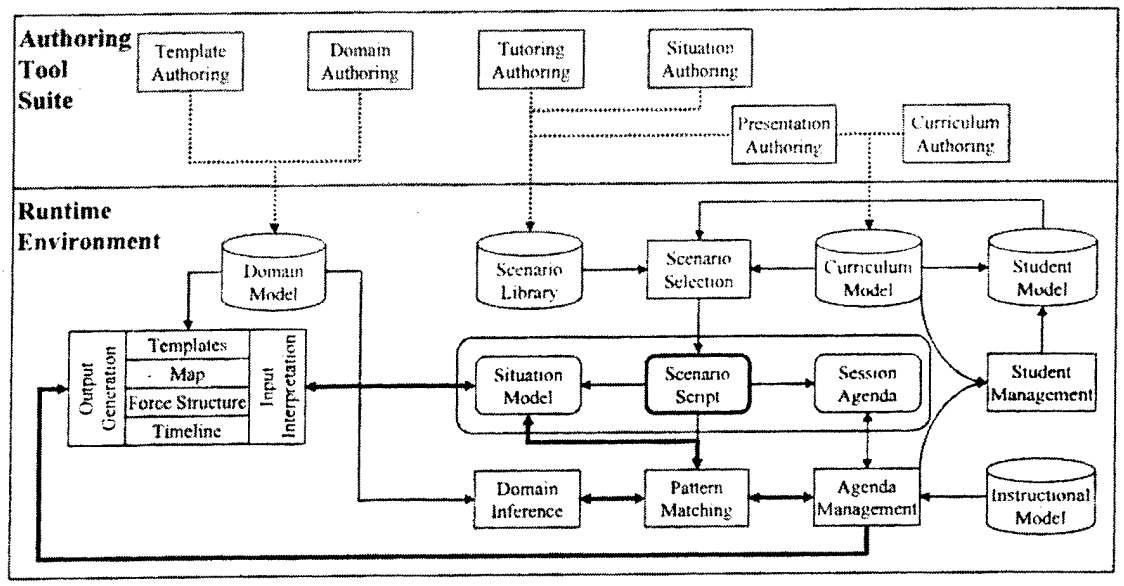


Figure 7. Detailed Functional Architecture Diagram for Final Phase II ComMentor Prototype.

The new diagram components in the Runtime Environment are primarily the five persistent stores: the Domain Model, the Scenario Library, the Instructional Model, the Curriculum Model, and the Student Model. The contents of most of these models have already been discussed at length in previous sections. The two new processing modules are those responsible for selecting which scenario to run during a session, and for managing the Student Model. Note that the Student Model is not authored per-se. Rather it is built as a set of student-specific annotations on the Curriculum Model, created automatically by the Student Management module that in turn is fed by the Agenda Management module. The Student Model accumulates a kind of win/loss record for Curriculum points linked to Evaluations and Dialogs as those nodes move onto and off the Session Agenda. That win/loss record (along with a memory of which Scenarios the student has already played) figures heavily in driving the Scenario Selection process.

The Authoring Tool Suite block in Figure 7 shows six different modules, each of which exists as a separate tool running in a separate window in the implemented system. Detailed discussion of each of these tools will be deferred to the final sub-section of this "Results" section. Here we simply note which authoring tools feed which persistent stores.

The Domain Model is primarily constructed using the Domain Authoring tool. The main screen in this tool is the ontology builder which provides a user interface that is intentionally similar to the tools provided by the Protégé system that we were forced to abandon at the end of project Year 1 for reasons of internal processing capabilities and flexibility. Patterns of natural language text that should map to a particular concept may be entered in this screen as well (primarily noun phrases). The tool also contains a rudimentary screen for defining inference rules that are used by the Domain Inference module. The Domain Model also accumulates some of its content from the more specialized Template Authoring tool, used to define the Templates that drive the forms-based user interface.

The Scenario Library is primarily constructed using the Situation Authoring tool and the Tutoring Authoring tool. The Situation Authoring tool provides the means to create Scenario-specific models, such as map annotation layers and force-structure diagrams. The Tutoring Authoring tool provides the means to create Scenario-specific Evaluation and Dialog trees (which refer to the entities created for the Scenario in the Situation Authoring tool). The ability to create simple text or text and GUI-configuration Presentations is generally built right into the screens of the Tutoring Authoring tool. However, the Presentation Authoring tool can also be used when necessary to create complex Presentations for display to the student at points during the scripted tutoring. Complex Presentations are generally sequences combining textual, graphic, and/or audio output (e.g., tutor text coordinated with manipulation of map layers to highlight portions of the map relevant to the current discussion).

The Curriculum Model is primarily constructed using the Curriculum Authoring tool. At its core, a Curriculum Model is a hierarchical arrangement of concept nodes intended to represent instructional objectives. These Curriculum points can be related to one another using a prerequisites language (though the prerequisite structure is not yet exploited by ComMentor's control structure). Curriculum points can also be annotated with Presentations that can be assigned roles with respect to the point such as introduction, remediation, example, anecdote, etc. (but again, these annotations are not yet exploited by ComMentor's control structure). As noted earlier, the main current use for Curriculum nodes is to annotate Evaluations and Dialogs, to help drive Student Model updates and Scenario selection.

Implementation Architecture. There are always multiple ways to slice any complex system. Here we present a partitioning that emphasizes the implementation layering, and aims to make clear which pieces of the system are reusable, and how those pieces are likely to grow over time. The ComMentor implementation can be thought of as being constructed in four main layers: (1) the General Representation Inference and Storage Toolkit (GRIST), (2) the Modeling Inspection and Linking Library (MILL), (3) the General Runtime and Authoring for INstruction (GRAIN), and finally ComMentor proper.

The GRIST is the knowledge representation infrastructure that we built starting in October 2002 as a replacement for the Protégé system we had used during the first year of the project. In the intervening year it has started to see application in a range of other projects, and those projects have started to expand the capabilities of the package.

Figure 8 breaks the GRIST package down into a finer-grained set of layers to depict its major capabilities and to emphasize ways in which it is designed for expansion. The five layers of GRIST include the Persistence Layer (which rides on some more primitive set of operating system and/or language provided capabilities), the In-Memory Layer, the Core API Layer, the Event Hub Layer, and the Processing Layer.

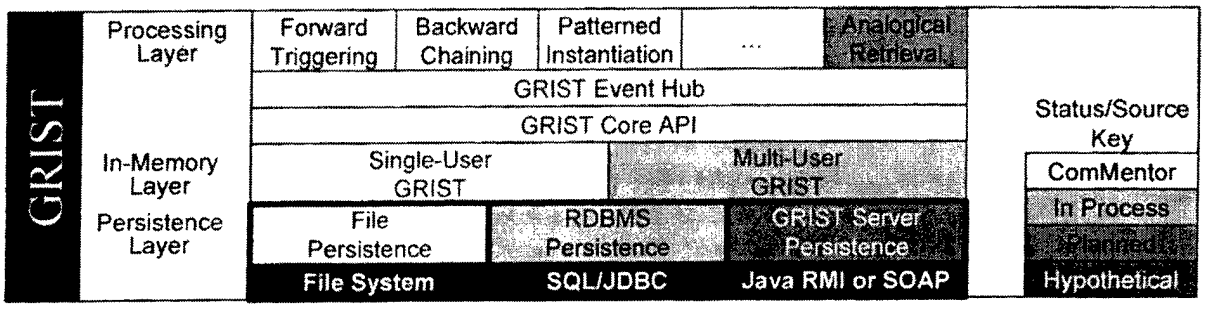


Figure 8. Implementation Architectural Layers Diagram for the GRIST Package.

There are currently three alternate Persistence Layer modules that have been implemented or planned. For ComMentor, we built and have been using the simple File Persistence module. This allows the contents of ComMentor's knowledge bases (KBs) to be stored in standard ASCII files in a standard file system (e.g., on Windows and Linux machines' hard drives). Another active project has been adding additional capabilities to the Persistence Layer. A Relational Database Management System (RDBMS) Persistence module is approaching completion. This module allows KBs to be stored in a standard relational database, and is intended to improve scalability, reliability, and ultimately to support concurrent access to KB contents. A multi-tiered system of intercommunicating GRIST servers—the GRIST Server Persistence module—is envisioned as future work, and may be developed under the same project that is currently building the RDBMS Persistence module. These three alternate persistence modules would provide the basis for broadly scalable applications of the basic GRIST knowledge model.

The In-Memory Layer implements the basic features of the GRIST knowledge model, which is broadly speaking, similar to the Protégé knowledge model. The primary view of knowledge in this system is as a set of frames (concepts) that by virtue of being instances of defined types, inherit a set of slots (attributes and relationships) that are constrained to take certain kinds of fillers (including most traditional data-types, as well as other GRIST frames). Types are arranged in an inheritance graph which allows them to have multiple supertypes. The system allows each frame to be an instance of multiple types (including type-frames, which can have multiple metatypes). A KB is a set of frame/slot definitions and instances. To encourage modularity and ontology reuse, KBs can include other KBs.

There are some significant differences between GRIST and the standard Protégé knowledge model. First, the underlying implementation of frames and slots is based on a logical assertion view of KB contents. This ultimately makes it easier to borrow proven techniques to implement inference algorithms more directly over GRIST structures, instead of requiring data export to some other module. Second, GRIST is able to manage multiple KBs simultaneously, so that the inclusion relationship is not a static file-based linkage, but is an in-memory relationship that supports dynamic updates to any KB and instant propagation of changes to other loaded including KBs. This is a major benefit in the context of ComMentor's extensive authoring tool suite that distributes authored contents across multiple KBs to encourage appropriate reuse. Finally, the original Single-User GRIST module built for ComMentor was somewhat more time-efficient in its basic operations than Protégé—at least for operations that were of special concern to ComMentor. In this layer too, we note that another project has

already funded implementation of a second in-memory module: Multi-User GRIST (which is a natural companion to the RDBMS Persistence Module, and eventually to the GRIST Server Persistence module).

The GRIST Core API layer is intended to provide a stable common view of GRIST data, independent of the particular in-memory implementation being used. The GRIST Event Hub layer provides a way for other modules to register to receive notifications of key events that occur in the GRIST package. This messaging system is used inside GRIST itself, but is exploited even more intensely when building on top of GRIST, as with the MILL framework discussed below.

Finally, GRIST's Processing Layer contains an expandable set of modules that exploit the basic GRIST knowledge structures in different useful ways. For ComMentor we built three processing modules: Forward Triggering, Backward Chaining, and Patterned Instantiation. These three modules share in common a language compilation technology that converts between the frame syntax of the overt GRIST knowledge model, and an assertional form that exploits the underlying structure of GRIST internal storage and simplifies the processing algorithms. Other processing modules are possible, and one—Analogical Retrieval—is in the planning stages for yet another funded project.

The Forward Triggering module supports matching GRIST knowledge structures against patterns specified in an expressive language that allows nested frame structures with variable binding, equality and inequality constraints, and logical combination through "and," "or," and "not" operators. Again, under funding from another project, we expect that language will soon be expanded to include quantifiers such as "all," "at-least," and "at-most." This module matches active patterns incrementally against changing KB contents. While not an implementation of the well-known Rete algorithm for matching many patterns against many (slowly changing) objects (Forgy, 1982), it shares some of that algorithm's efficiency in such application contexts.

The Backward Chaining module supports a very simplified Prolog-like inference mechanism. The current implementation is limited to operating over what is best characterized as a function-free variant of first-order logic, and lacks most of Prolog's built-in predicates and its typical efficient virtual-machine compiled implementation. This module is likely to be an area of active refinement across future projects that adopt the GRIST package as part of their development infrastructure.

The Patterned Instantiation module provides a way to specify new GRIST structures that should be constructed at run-time. Templates that define forms for the user interface generally have "make" patterns associated, so that when the user fills in the form the system creates new GRIST assertions that incorporate their input choices. When make patterns are associated with forward-triggered "match" patterns the result is a kind of forward-chaining inference system; the assertion of one set of structures (e.g., from user input) can trigger a rule that instantiates additional structure, that can in principle trigger yet another rule, and so on. The fact that make patterns share essentially the same pattern language with match patterns is what will allow relatively easy introduction of a layer for authoring Evaluation and Dialog (match) patterns using the standard forms-based user interface (and the associated make patterns).

Figure 9 is similar to Figure 8, but shows the layering structure of the MILL package. The MILL is the display and editing GUI framework that evolved, starting roughly in July 2002, to ease construction and to support integration among the many pieces of ComMentor's extensive authoring tools suite. Since its initial conception it has started to see application in a range of other projects, and plans are in process to refine the capabilities of the package within at least one current project. Figure 9 breaks MILL down into four main layers including the Display Components Layer (which rides on more primitive capabilities provided by Java and GRIST), the Layout Layer, the GUI Capabilities Layer, and the Packaging Layer.

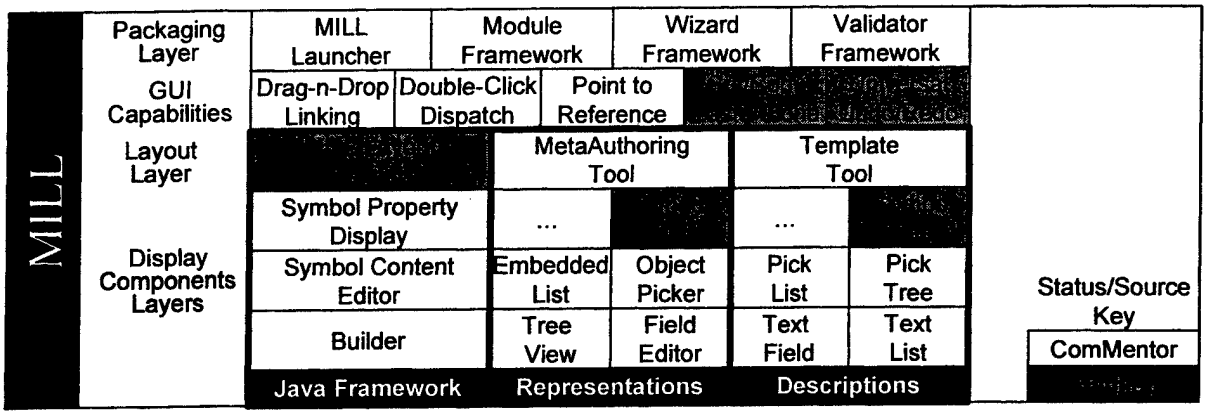


Figure 9. Implementation Architectural Layers Diagram for the MILL Package.

The Display Components Layer contains abstract and concrete objects that can function as pieces of a user interface. The three highlighted columns at this layer (and the layer above) are not alternate implementations, but rather clusters of components that work together to cover different types of GUI needs at different grain sizes.

The first column emphasizes objects that represent entire windows or major sub-panes of those windows intended to provide customizable display and manipulation of GRIST objects (frames). As shown, these are objects that build upon the general Java Framework for GUIs. A Builder is an entire window or window layout that typically manages a constellation of GRIST frames. Examples include all the different authoring tools (see "Final Prototype Authoring Capabilities" below for details of the authoring tool screens). A Symbol Content Editor is a major sub-pane of a Builder devoted to editing a particular kind of GRIST frame. In the Tutoring Authoring tool, depending on what is selected in the window's top band, a different Symbol Content Editor will appear to support display and editing of a Scenario, Scene, Evaluation, or Dialog. A Symbol Property Display is a widget that supports display (and optionally editing) of a particular property of a GRIST frame. Normally this is a single slot of a frame, but more complex properties can be mapped to this kind of object, as discussed below.

The second column contains objects that specialize in Symbol Property Display and can be configured into Symbol Content Editors. These specialized widgets exist to display (and edit) particular kinds of slots or more complex properties. A Field Editor allows the user to type in information for a slot that expects text. An Object Picker allows the user to choose an existing frame of the type required to fill a slot. An Embedded List allows the user to see and select among a set of frames that fill a slot. The Tree View is an example of a widget that can support

visualization of more complex properties; it is typically used to view and edit a recursive relationship such as a type or packaging hierarchy. Other such widgets, simple and complex, exist and more can be defined to fit into the framework.

The third column is in some ways similar to the second in that it contains an extensible set of relatively low-level widgets. The difference is that this set is used by the form-based input system to allow users to manipulate descriptions of GRIST frame slot fillers rather than to directly manipulate GRIST objects. That is, these widgets are used in templated forms that provide details to flesh out a make pattern which may then be turned into actual GRIST frames (or may be used for other purposes, such as generalization to a match pattern). This distinction accounts for why the third column is shown built on top of "Descriptions" (that is patterns) while the second column is built on top of "Representations" (that is actual GRIST frames).

The Layout Layer contains three different kinds of layout control mechanisms, one for each of the Display Components columns. The major GUI components are laid out using the standard Java notion of a Layout Manager. Plans are in place to develop a custom MILL Layout Manager that will do a nicer job of aligning things the way we want them to appear. The slot filler components are laid out using the MetaAuthoring tool. The pattern filler components are laid out using the Template tool. In the future there may be an effort to synthesize a common widget-set and layout mechanism across what are now the representation and description columns.

The GUI Capabilities Layer is where common user interface functionality is defined—features that should work (where meaningful) for every widget in every layout in every screen. Currently there are three major features provided by this layer: Drag-n-Drop Linking, Double-Click Dispatch, and Point to Reference. Other such capabilities are planned including a more universal implementation of cut, copy, paste, and clone operations, and a universal undo and redo capability. These are the sorts of capabilities that most users have come to expect from modern GUI applications.

Drag-n-Drop linking was one of the most difficult general features to implement for MILL. We have now generalized the capability so that it works both within individual widgets (e.g., the Tree View provides extensive and customizable support for rearranging hierarchies by dragging and dropping nodes within the tree), across widgets within the same layout (e.g., in the Presentation Authoring tool, Presentations can be reorganized within a sort of filing system by dragging and dropping from one pane to another), and also across windows (e.g., Presentations can be linked to Dialog nodes or Curriculum nodes by dragging them from the Presentation Authoring tool and dropping them in the Tutoring Authoring tool or the Curriculum Authoring tool).

Double-Click Dispatch is a useful feature that allows the user to focus on essentially any GRIST frame that they come across in any context by double-clicking on the reference to the frame. The result is that an appropriate viewer/editor opens up and displays the selected frame so that all of its details can be seen and updated as needed. For instance, when browsing the Curriculum in the Curriculum Authoring tool, the user can quickly view the details of a Presentation attached to a Curriculum point by double-clicking on the name of the Presentation in the Curriculum point's layout.

Finally, Point to Reference is a capability used in ComMentor's runtime to support multimodal input by interpreting mouse clicks on various situation visualizations as references to GRIST frames that should be plugged into a chosen blank in the current form. For instance, clicking on an icon in the force-structure tree can be interpreted as a request to fill the designated military unit frame into the "Enemy" blank of the Tasking form. Likewise, clicking on the map can be interpreted as a request to fill the Tasking form's "Where" blank with a particular location.

Finally, MILL's Packaging Layer is a set of frameworks to ease assembly of a complete application. The MILL Launcher is the default top-level window for a MILL-based application. It appears at application start-up and presents a list of available Builders, each representing a module for viewing and editing a particular sort of knowledge. The ComMentor authoring tool suite starts up with the MILL Launcher offering the six major tools shown at the top of Figure 7 (as well as two others—the MetaAuthoring Tool and the Ask Tool—that are not of such general use). Each tool is packaged as a MILL Module according to the conventions of the Module Framework. Basically this provides support for packaging conventions that allow delivery of a Module as a single Java .jar file and easy configuration of a suite (and Launcher choices) from a set of such Modules.

The Packaging Layer also provides some initial support for two classes of handy pop-up utilities available from the Launcher or from individual Builders: Wizards and Validators. Wizards are similar in spirit to the user-friendly dialog-sequences of the same name found in many modern applications. ComMentor has two Wizards defined for it: the New Scenario Wizard, and the Scenario Questions Wizard. Validators are routines that scan a GRIST KB to check for certain types of problems or inconsistencies, and produce their results as a list of frames with identifiable problems. The Double-Click to Dispatch feature then makes it easy to go edit the offending frames. Conceptually, a better approach would be to build the system so that it is not possible for such problems or inconsistencies to arise in a KB, but that is a very difficult goal to achieve, so for the foreseeable future, we expect ComMentor and other MILL applications to need a suite of Validators to quickly find and correct identifiable classes of authoring errors.

Figure 10 follows on Figure 8 and Figure 9, showing the layering of the GRAIN package. GRAIN is a first pass at a general ITS framework. Not surprisingly, since it sits at the top of the stack of reusable packages, GRAIN is the least well-developed of the packages discussed in this Implementation Architecture sub-section. Many of its modules are either not fully developed, require heavy customization for any particular application, or are purely notional. Nonetheless, the framework gives us a good way to think about the architecture of a fairly broad range of possible ITSs and to achieve some level of reuse. As shown in Figure 10 GRAIN is conceived of as having five layers: the Distribution Layer, the Universal Layer, the Algorithms Layer, the Visualizations Layer, and the Custom Layer.

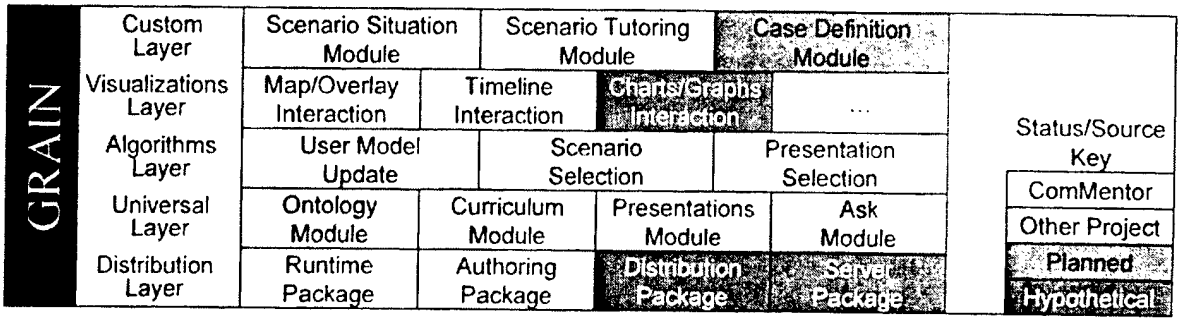


Figure 10. Implementation Architectural Layers Diagram for the GRAIN Package.

The Distribution Layer focuses on carving possible ITSs up into major components and supporting their distribution and delivery. ComMentor, for instance, has a Runtime Package and an Authoring Package. These are deliverable separately (though the Authoring Package effectively includes a version of the Runtime Package in order to facilitate testing of authored materials). The current ComMentor prototype has no Server Package, though it would probably not be a major challenge to configure the system for Web-based delivery. Likewise, ComMentor does not have a Distribution Package to manage delivery and/or updates of executables, ontologies, scenarios, or student models. These last two modules will remain hypothetical until some project comes along with a need and funding to build them.

The Universal Layer includes a set of MILL Modules that we believe are potentially of use in just about any ITS that might be built. The four modules shown here all have reasonably well developed implementations, and were all used in ComMentor (are part of the authoring tool suite), however two of them—the Ontology Module, and the Ask Module—were developed under funding from other projects. Over time, these modules may be refined and/or additional modules may be developed and migrate into the Universal Layer as their general utility is recognized.

The Algorithms Layer includes a set of modules that represent key recurring processes in many ITSs. ComMentor has particular implementations of each of these modules tuned to its particular style of instruction and its particular knowledge structures. In general, it is probably best to think of this layer as being populated by a pluggable set of alternate implementations for a set of common algorithmic needs, rather than to expect widely shared blocks of code. Put in object-oriented development terms, we need to first aim for stable interfaces for each key algorithm class, and later worry about sharing implementations. Even just within the scope of potential further ComMentor development and evaluation, supporting easy plugability of alternate implementations for these algorithms will potentially help us explore which heuristics produce the best results.

The Visualizations Layer is again a kind of GUI widget layer, but this time at a larger grain size and higher semantic level. The idea here is to accumulate a toolkit of frequently useful visualization types that can be used in many ITSs. For ComMentor, we built the Map/Overlay Interaction module and an initial capability for the Timeline Interaction module. Within the company other visualization modules, such as the Charts/Graphs Interaction module, are being developed, and over time they should be ported to work with GRIST knowledge structures in the MILL framework, and eventually become available in the GRAIN package.

Finally, the Custom Layer is similar to the Algorithms Layer in that the modules shown there represent commonly recurring constructs in ITSs, but likely require significant customization for any given ITS application. In this case, the modules represent actual MILL Modules—entire Builders that live in their own windows and support complex (customized) editing of particular GRIST frame types. In ComMentor, the Scenario Situation Module and Scenario Tutoring Module are elaborate custom Builders that appear in the Launcher as two of the major authoring tools. While many other ITSs might need Situation and Tutoring authoring, only those focused on tactical tutoring are likely to need exactly the particular set of Situation Authoring facilities developed for ComMentor, and only a Socratic tutor following ComMentor's approach is likely to need exactly the Tutoring Authoring facilities developed for ComMentor. The same project that is planning to add an Analogical Retrieval module to GRIST Processing Layer is planning to build a new Case Definition Module to author the cases that will be the subjects of analogical retrieval. While that project is also likely to need Situation and Tutoring authoring, it is unlikely to need exactly what ComMentor has created. Fortunately, the underlying capabilities of MILL and GRIST should make it relatively easy to configure new versions of Scenario Situation Module and Scenario Tutoring Module appropriate to the new application context.

Final Prototype Runtime Capabilities

Innovations summary. The final Phase II ComMentor prototype runtime system possesses, to varying degrees, a dozen general properties or capabilities that we identified as important and useful in an ITS for battlefield command reasoning skills.[4] Together, these give a good sense for what is innovative and interesting about ComMentor.

- Tactical Decision Game (TDG) Socratic tutoring – As we said in the "Introduction" sub-section on "Innovations," automating Socratic tutoring for a problem as difficult as tactical decision-making constitutes a major extension beyond the existing state of the art. While ComMentor is by no means the definitive word on tactical training, it has succeeded in demonstrating a basic level of capability at problem presentation, solution critiquing, rationale elicitation, and responsive discussions that together approximate important aspects of observed live TDG sessions.

- Pre-analysis of scenarios for Evaluations/Dialogs – Again, as noted in the "Introduction," the way to build a Socratic tutor for an open-ended problem area is to build it largely around "deep enough" representations tuned to specific scenarios. In ComMentor, commonality in domain ontology, curriculum, scenario structure, instructional strategies, and user interface are complemented with scenario-specific Evaluations and Dialogs identified and organized by SMEs.

- Multi-media/multi-modal interaction – Socratic discussion of professional-level problems benefits greatly from ability to use standard domain-appropriate representations to visualize and manipulate aspects of the problem situation. In ComMentor, use of situation maps, force-structure wire-diagrams, and time-lines is integrated with the

[4] The 12th major feature, not in this list of runtime features, is the existence of a comprehensive and extensible authoring tools suite.

template-driven forms-based input interface that constitutes our substitute for a full spectrum natural language understanding component.

- Provide a large, natural input space – Socratic tutoring works in large part because it encourages the student to engage deeply in the problem, while simultaneously offering guidance and modeling on how they think about and pursue solutions. Ideally, when it is the student's turn to offer up a plan or situation analysis, they should not feel overly constrained—that is, they should not be forced to say something other than what they were thinking because of limitations in the user-interface or the system's ability to comprehend input. ComMentor's input space is quite large, however, to speed interaction and avoid user frustration with the form-based input system, we have decided to provide far more prompting than initially planned. Only user trials and evaluation will reveal whether we have found a workable tradeoff. Improvements in language processing over time may provide more freedom to back off from our current amount of user guidance.

- Scenario selection based on user model – A student's ComMentor experience is organized around particular scenarios both to make knowledge representation and acquisition feasible, and also because case-based training is an appropriate, effective, and engaging approach to professional-level decision training (as witness common practice in a wide range of professional schools). If a tutor session is to be organized around a scenario, then the choice of which scenario to focus on in any given session becomes an important instructional decision. Though ComMentor still has an exceedingly limited repertoire of Scenarios at its disposal, it already has in place a straightforward algorithm for choosing scenarios based on the past experiences and performance of the individual student.

- Continuous non-local student input assessment/triggering/recognition – Over the course of the extended interaction that constitutes a ComMentor session, the system makes most of its important diagnostic and control decisions based on the accumulated record of what the student has said, not based solely on the immediate answer to the most recent question. This applies to Evaluation activation, and Dialog node pursuit. This is accomplished through exploitation of a pattern matching algorithm that continuously tracks updates to the system's knowledge bases.

- Active student input disambiguation – ComMentor provides mechanisms that allow the system to respond to partial or incorrect student inputs with specific requests for additional information, clarification, or correction. The same pattern matching mechanism that underlies Evaluations and Dialog processing is used to trigger disambiguation questions. The same Presentation and input processing mechanisms that support main-line interaction are available to handle disambiguation.

- Tutor initiative driven by a pervasive agenda mechanism – The non-local mechanisms by which Evaluations and Dialogs are triggered and satisfied is combined with an agenda management algorithm that allows the tutor to develop and maintain an extended plan for its interaction with the student. Triggered nodes are placed on the agenda and fully

processed or satisfied nodes (and their descendents) are removed from the agenda to maintain a coherent and user-focused overall session structure.

- Dialog prioritization based on author scripting – Given a Dialog activation process that hinges on the triggering of Evaluations based on input arriving at any time, there must be some mechanism for determining sequencing of pending Dialogs in the evolving agenda. Contrary to our original expectation that it would make sense to appeal to the student model to help determine this ordering, we have found that overall session coherence benefits from having the order determined at authoring time (to the extent that runtime constraints allow respect for that ordering).

- Tree-based dialog pursuit mechanism – A Dialog can be developed to any depth thought appropriate by a Scenario author. Generally, any high-level proposition that is the focus for discussion can be backed up with a set of more detailed propositions that together constitute evidence or an argument for the higher-level node. If students do not spontaneously understand or accept a high-level point, the system can traverse the nested structure in an effort to clarify the situation or convince the student of the point.

- Limited student initiative for question digressions – While the backbone of a ComMentor session is driven by the Evaluations and Dialogs that are processed through the agenda mechanism, the system also has the ability to deal with student-initiated questions that constitute digressions from the perspective of ComMentor's session agenda. The system is generally prepared to answer basic factual questions about the current scenario or about common entities in the military world (such as how far a given weapon can shoot).

Runtime details. A session with ComMentor starts with user login and Scenario selection, and proceeds to actual play of a particular Scenario. Scenarios are structured into a series of Scenes of different types. In the current system, every Scenario has three Scenes: (1) a scenario briefing and order-elicitation Scene, (2) a main discussion Scene, and (3) a final reflective session Scene.[5] The scenario briefing is a complex sequential Presentation that combines slides (generally .gif files) and audio (.wav files). Figure 11 shows the set of login screens. Figure 12 shows a screen shot from during the scenario briefing for the "Enemy Over the Bridge" Scenario.

[5] Feedback from one of our SMEs and experience with the "Clash at Timpan-ni" Scenario argue for some restructuring and expansion here. Order elicitation should be reconfigured so it can be used (and reused) during the main discussion. A new kind of Scene focused on terrain analysis should be introduced, and optionally placed before the main discussion. With these changes, the need for a separate introductory Scene would likely be eliminated.

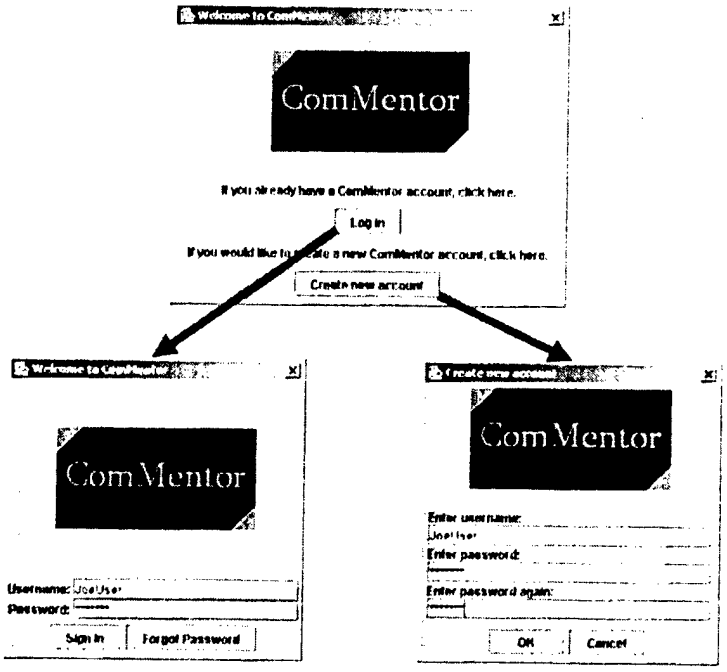


Figure 11. ComMentor Login Screens.

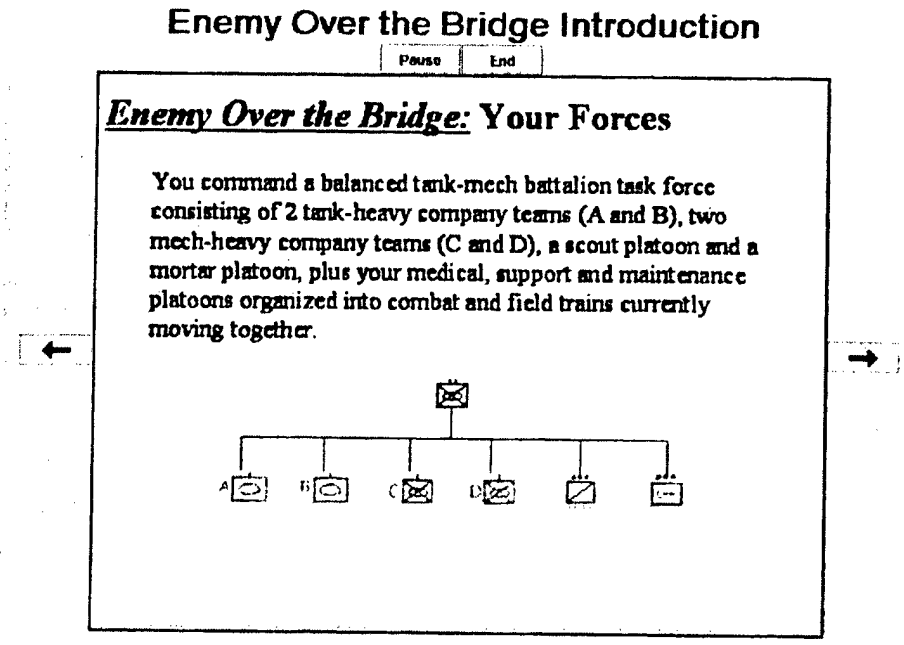


Figure 12. Slide from ComMentor Scenario Briefing at Start of "Enemy Over the Bridge."

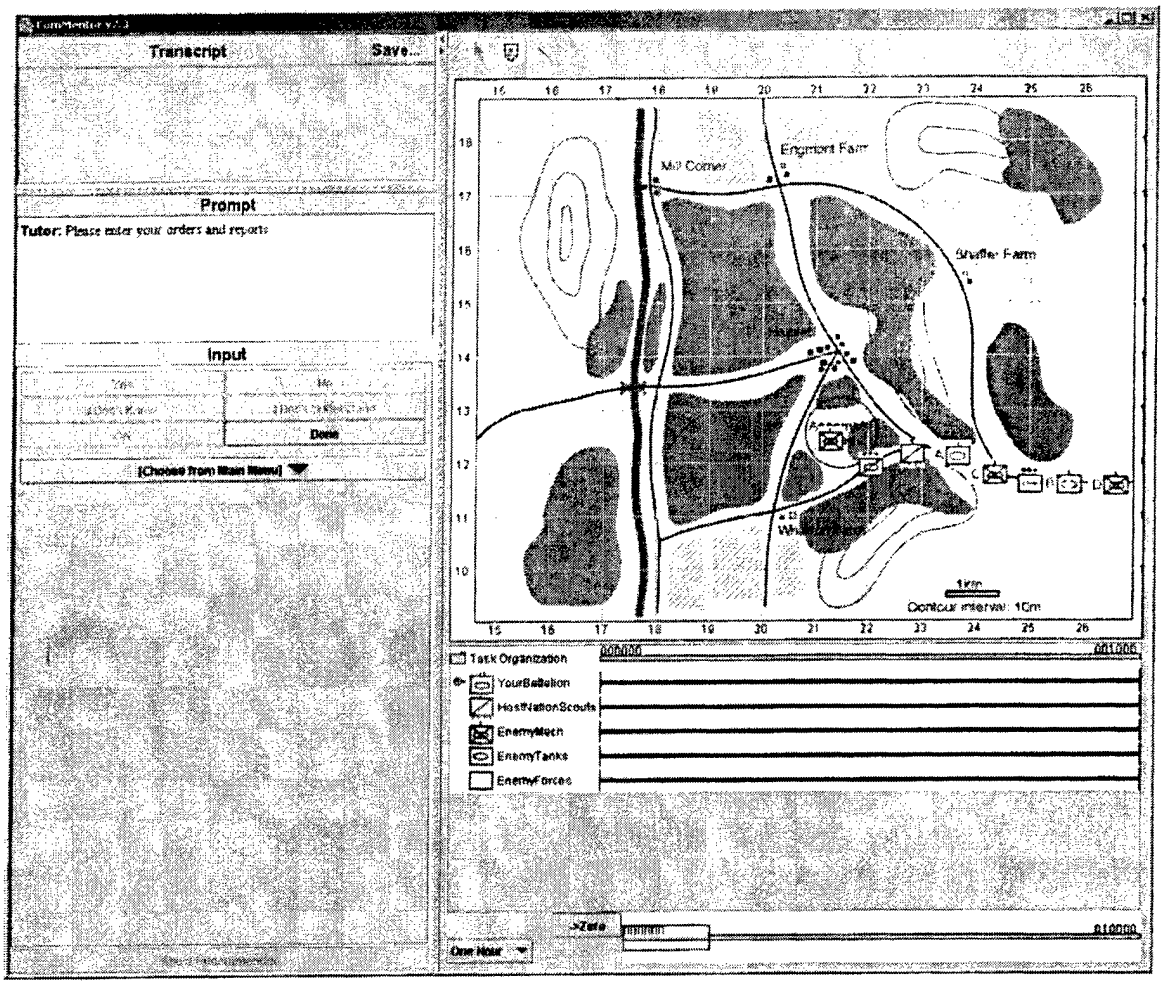


Figure 13. Main ComMentor Interaction Screen at Start of "Enemy Over the Bridge."

At the end of the briefing, the student is transitioned to the main ComMentor interaction screen shown in Figure 13. That screen has three major parts: (1) the map, (2) the force-structure wire-diagram tree and timeline, and (3) the forms-based input/output area.

The interactive situation map, located in the upper right, takes up the most space on the screen. Initially it shows the situation as described at the close of the briefing. This map can be manipulated by both the student and the automated tutor. Icons representing forces can be dragged to new positions. Checkpoints can be laid down and used to describe taskings. Locations and forces can also be referred to by pointing with the mouse.

Beneath the map is the combined force-structure wire-diagram tree and timeline view. A standard tree-display can be expanded to show what is known about blue and red force-structures and their relationships. Each element represents a military unit, showing its name and an appropriate icon. If not already present on the situation-map, those icons can be dragged and dropped on the map to suggest where the corresponding unit should be located. To the right of the force-structure tree is a set of timelines that show major scenario events and taskings assigned to particular units. Like the map, the timelines can also be used to fill in details of proposed taskings by pointing and clicking with the mouse. Controls at the bottom of the screen

allow you to scroll forwards and backwards in time, and to set an appropriate scale for the display.

The final part of the main ComMentor display—taking up the entire left column—is the form-based input/output area. The top pane (the Transcript pane) accumulates a textual transcript of all major events that happen during the tutoring session. Not surprisingly, that pane starts out blank. The next pane down (the Prompt pane) shows only the most recent output from the tutor. Initially the prompt pane contains the tutor's request that you enter "orders and reports" in response to the described situation. The bottom pane (the Input pane) provides a changing set of buttons, menus, and forms that offer the main way for you to participate in dialog with the tutor. This area usually contains an array of buttons labeled with standard options such as 'Yes' 'No' or 'I Don't Know', followed by a cascading set of drop-down menus (here labeled "Choose from Main Menu"). These menus offer you ways to say more complicated things to the tutor—to make statements or to ask questions:

- Students can make statements about facts in the military world such as the capabilities of weapons-systems, vehicles, and military units.

- Likewise, students can ask questions about the same sorts of facts.

- Students can also make statements about the facts of the current scenario—talking about terrain properties, or about military units, their activities, plans, and knowledge.

- Finally, students can also ask questions about the scenario, which will often take them back to parts of the scenario's introductory briefing.

The initial tutor request asks the student to enter orders and reports. The system does not yet have a good form defined for entering reports, so we focus here on issuing orders. There are several ways the student can assign a task to a unit. The most common approach is to use the mouse to drag the icon representing the unit to the location on the map where the student wants it to go. In response, the tutor opens the tasking form in the Input pane, and primes several fields of the form including who is receiving the order, where they are supposed to go, and the route by which they are supposed to get there. The other six fields of the tasking form are blank: the role the unit is supposed to play in the larger operation, the start and/or end times for the action, the specific task, the enemy force that is the target, and the intent behind the tasking. Figure 14 shows the result of dragging alpha company up the road and into the assembly area, as part of specifying an attack on the enemy mech located there.

Once a set of orders has been entered, the student clicks 'Done' and the system begins to respond to what has been said. The system does this by matching patterns associated with Evaluations against the Situation Model as updated by the student's input. There are at least two kinds of patterns associated with Evaluations: trigger patterns, and refinement patterns. Trigger patterns determine when the tutor will judge the Evaluation to be relevant, and thus queue up the corresponding Dialogs for discussion. Refinement patterns are generally subsets or variants of the trigger pattern that prompt the tutor to ask clarification questions.

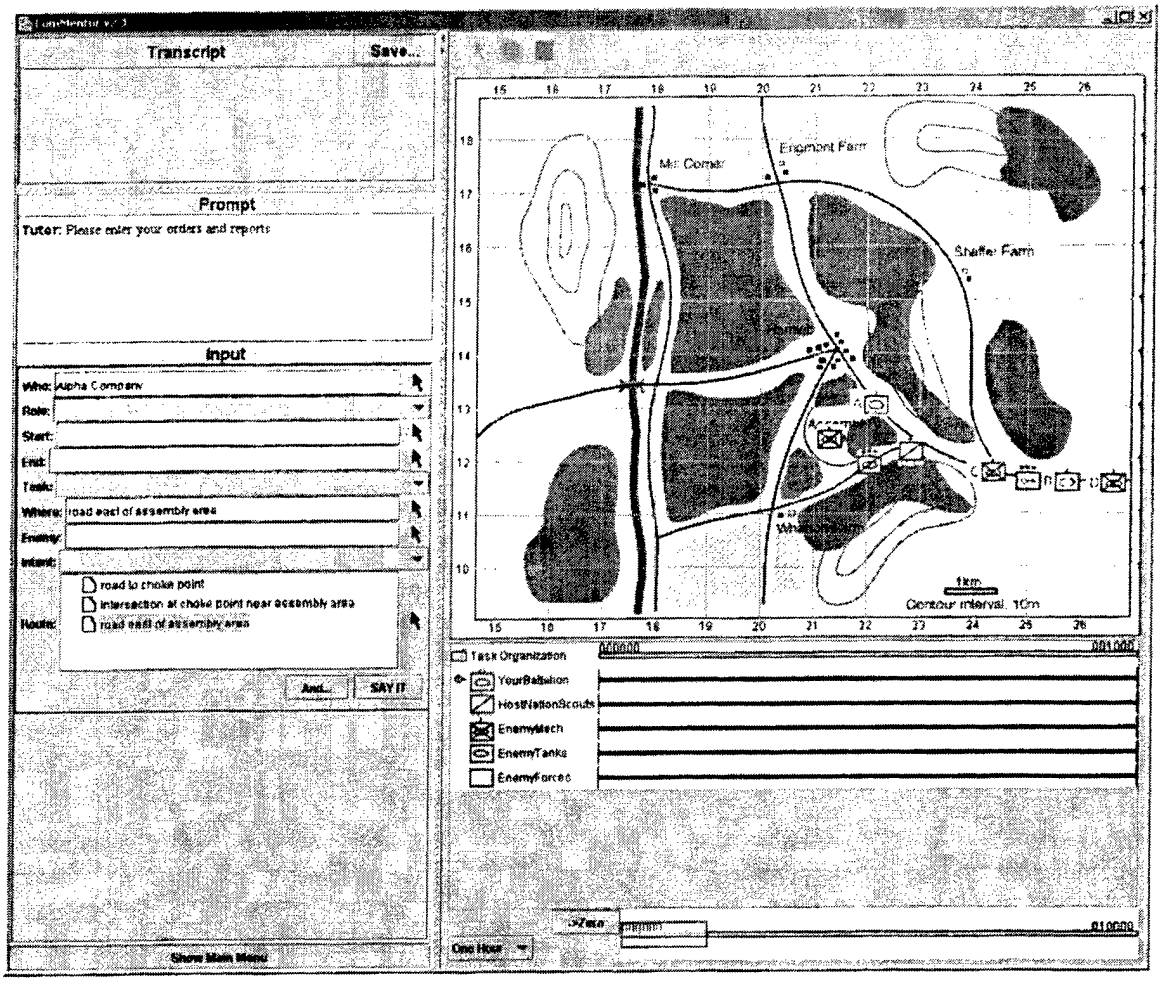


Figure 14. Main ComMentor Screen After Dragging Alpha Company Up the Road.

When an evaluation's trigger pattern is satisfied, its corresponding Dialog node is added to the tutor's agenda.[6] The node is added in a position that corresponds to the extent possible to the Evaluation's position in the Scene's tree of Evaluations. The idea is that the author should be able to control (or express preferences) about the order of Evaluation discussions by the order in which they put the Evaluation into the tree. If, however, an Evaluation that appears later in the tree was triggered earlier and has already been discussed (or is in process) then clearly the newly triggered Evaluation cannot have its discussions handled earlier, but rather must wait until the tutor has a chance to get around to the new topic. When the tutor needs to start a new Dialog (e.g., when the Scene starts, or when a previous Dialog has finished) it pulls the first Dialog node from the front of the Agenda.

The purpose of most Dialog nodes is to see if a student understands some point about the Scenario, or to get them to see (and say) that point if they don't spontaneously get it. For instance, a Dialog node might represent an assessment of the Red forces facing the student in a Scenario, where the evidence presented in the lead-in ought to suggest a mechanized battalion. In that case, the student succeeds on that node if they say they are facing a mech battalion. If the

[6] If multiple Dialogs are specified for an evaluation, they are joined together into an implicit or-Dialog node.

student has correctly characterized the opposing force before they even get to the Dialog node, then the node succeeds without doing anything. Otherwise, processing of the node is initiated.

When a Dialog is initiated the system goes through a standard cycle that involves playing through a set of up to six dialog Presentations, and possibly invoking a set of recursive Dialogs. All six Presentations are optional, as are the recursive Dialogs. The six Presentations are labeled: (1) Setup, (2) Introductory Question, (3) Hinting Question, (4) Leading Question, (5) Success Summary, and (6) Failure Summary. If they are present, each of the three question Presentations normally provides an opportunity for the student to say something to the system in response (or to ask a Digression Question of their own).

The standard sequence is illustrated by the flowchart in Figure 15 below. In this flowchart, processing starts at the triangle and ends at the octagons (stop-signs): the octagon with a '+' sign indicates the student is credited with having succeeded at the node, while the one with a '-' sign indicates the student is judged as having failed at the node. Diamonds represent tests, each of which has two outward bound arrows, one for when the test succeeds (labeled 'Y') and one for when the test fails (labeled 'N'). Finally, rectangles represent processing steps. The rectangles labeled 1, 5, and 6 represent the simple playing of a Presentation for the student. The rectangles labeled 2, 3, and 4 represent the playing of a Presentation and the requesting of input from the student. The rectangle labeled R represents the invocation of a recursive set of Dialog nodes. The dotted lines and rectangles represent optional paths for when the student either asks a Digression Question of their own, or gives an answer that triggers one of the tutor's follow-up Disambiguation Questions.

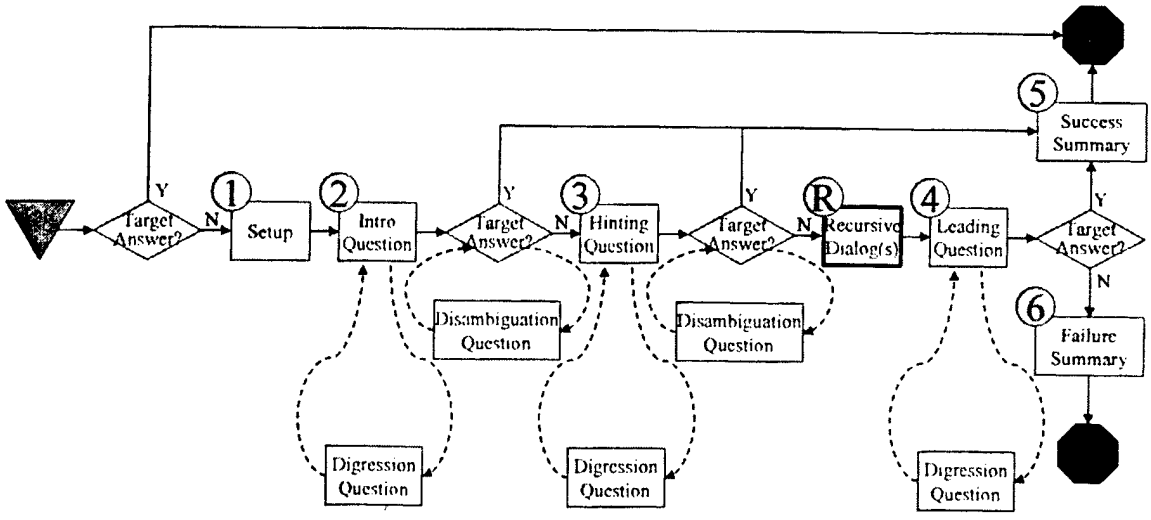


Figure 15. Flowchart of Processing for a Dialog Node.

The rule for processing of recursive Dialogs depends on the particular Dialog node. A node may specify its children should be handled using one of the options AND, OR, AT-LEAST, AT-MOST, or BETWEEN. The default rule is AND; in this case, all of the Dialog node's children will be executed. For an OR node, the children will be executed until either the student succeeds at one of them, or the tutor runs out of nodes. AT-LEAST is similar to OR, except some number more than 1 may be specified, and the system will keep attempting child nodes

until the student succeeds on the designated number (or the tutor runs out of nodes to try). AT-MOST nodes and BETWEEN nodes (which combine AT-LEAST and AT-MOST conditions) will only start to make sense once the system begins to factor real-world clock time into its tutoring decisions.

One other complication affects the processing of Dialog nodes. In addition (or as an alternative) to having a pattern for their target answer, Dialog nodes may specify an entry test pattern. This functions much like an Evaluation's trigger pattern in that the Dialog will not be attempted until the entry condition is satisfied. This can lead to Dialog nodes being executed in an order other than the one specified by the author. The primary use of this feature to date has been to construct alternative-evaluation Dialogs where pairs of children are authored to discuss different possibilities; one node in each pair has as its target answer one of the options, and the discussion at that node tries to get the student to recognize the option exists, while the other node of the pair has that same option as its entry condition so that only once the student states the option does the system start in on an evaluation of the option.

The Student Model is updated based on which of the Evaluations that become active actually get triggered, and based on which of the Dialog nodes that become active are exited successfully or unsuccessfully. Curriculum points can be linked to Evaluations in a positive or negative sense. If an Evaluation triggers, its positive Curriculum points get credit and its negative Curriculum points get blamed. If an Evaluation could have triggered, but didn't, then credit/blame is reversed—its positive Curriculum points get blamed and its negative Curriculum points get credit. Curriculum points can also be linked to Dialog nodes, but only in one sense. If the Dialog node exits successfully, then the linked Curriculum points get credited. If the Dialog node exits with failure, then the linked Curriculum points get blamed.

Runtime sample. Appendix C contains a detailed annotated transcript of a sample session with ComMentor. Here we examine some small snippets of that transcript in detail, looking behind the scenes at the data structures and processing that produce the observed behavior, as well as some possible alternative behaviors that might have been exhibited in response to other student inputs. We start by focusing on pieces of the transcript near the start of the session that deal with the Red situation. This allows us to explore the Dialog node processing mechanisms. Later we consider some examples that depend on the Evaluation processing mechanisms.

Tutor	Let's start by talking about the red forces and their situation.
Tutor	What sort of force do you think you're facing in your area east of the river?
Student	Enemy Mech is located at area east of the river.
Tutor	Can you characterize what sized unit you are seeing here?
Student	bat is located at area east of the river.
Tutor	Yes. Given that you've seen at least a company of enemy mech in what was supposed to be your assembly area, and given reports of at least 10 T-62s, plus the fact they've been moving over the bridge for at least 2 hours, it seems reasonable to assume that there's something on the order of a red battalion already east of the river.

This excerpt from the Appendix C transcript represents the tutor's initiation of the discussion managed by the Evaluation Eval_RedPicture. Eval_RedPicture is unique in EOTB in

that it is the only Evaluation that has no trigger Pattern to satisfy. That means it will always be discussed in every student session. Further, given that it appears early in the list of top-level Evaluations, it will always be discussed early in a session.

Figure 16 shows much (though not all) of the Dialog tree for Eval_RedPicture. This Evaluation has a single root Dialog node Dial_RedPicture, which has four sub-nodes: Dial_RedBattalion, Dial_RedOffensive, Dial_RedBridgehead, and Dial_RedBridgeheadLayout. Figure 17 is the first of a set of diagrams that pick out the main contents of various EOTB Dialog nodes—in this case the contents of Dial_RedPicture (that is, the root of the tree shown in Figure 16). These diagrams show the entry and target Patterns associated with a node, as well as the six Presentations: Setup, Introductory Question, Hinting Question, Leading Question, Success Summary and Failure Summary.

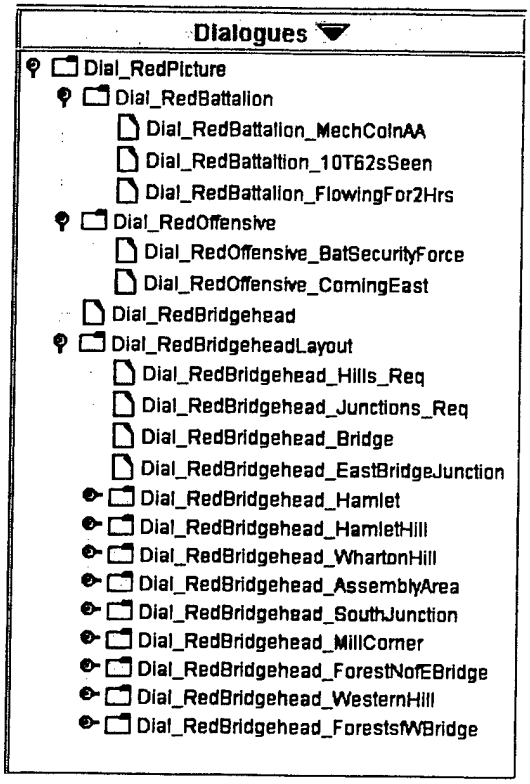


Figure 16. EOTB Dialog Tree for Eval_RedPicture.

Dial_RedPicture	
Entry:	
Target:	
Setup:	Let's start by talking about the red forces and their situation.
Intro:	
Hinting:	
Leading:	
Success:	
Failure:	

Figure 17. EOTB Dialog Node Dial_RedPicture.

Dial_RedBattalion (see Figure 18) is a Dialog node whose target Pattern aims to get the student to acknowledge that they are likely facing a battalion-sized enemy element east of the river. Dial_RedOffensive's target Pattern (see Figure 19) is the assessment that this battalion-sized element is probably the advance guard of an enemy offensive. Dial_RedBridgehead's target Pattern (see Figure 20) is the realization that Red is probably trying to form a bridgehead east of the river in order to secure the crossing. Dial_RedBridgeheadLayout does not itself have a target Pattern, but it does contain a large number of sub-nodes ready to elicit and then discuss a variety of possible positions Red might try to use for the bridgehead.

Dial_RedBattalion	
Entry:	
Target:	(AtLocation (subject (& EnemyUnit Battalion)) (object =EastOfRiver))
Setup:	
Intro:	What sort of force do you think you're facing in your area east of the river?
Hinting:	
Leading:	With reports of at least a company of mech, and most of a company of tanks, would you say you might be facing a battalion?
Success:	Yes. Given that you've seen at least a company of enemy mech in what was supposed to be your assembly area, and given reports of at least 10 T-62s, plus the fact they've been moving over the bridge for at least 2 hours, it seems reasonable to assume that there's something on the order of a red battalion already east of the river.
Failure:	Well, given that you've seen at least a company of enemy mech in what was supposed to be your assembly area, and given reports of at least 10 T-62s, plus the fact they've been moving over the bridge for at least 2 hours, it seems reasonable to assume that there's something on the order of a red battalion already east of the river.

Figure 18. EOTB Dialog Node Dial_RedBattalion.

Dial_RedOffensive	
Entry:	
Target:	((! Attack Guard Occupy) (who EnemyUnit) (where =EastOfRiver))
Setup:	
Intro:	What do you think the enemy is up to?
Hinting:	What task might a battalion-sized force be carrying out in this push across the river?
Leading:	So would you accept that a battalion-sized force could be the advance guard for a larger offensive operation moving into our territory east of the river?
Success:	Yes, since you've got most of a battalion on your side of the river, it's not unreasonable to guess that the enemy has beaten us to the punch, launching an offensive to the east, just as we were planning to launch one to the west..
Failure:	Well I would say that given you've got most of a battalion on your side of the river, it's not unreasonable to guess that the enemy has beaten us to the punch, launching an offensive to the east, just as we were planning to launch one to the west..

Figure 19. EOTB Dialog Node Dial_RedOffensive.

Dial_RedBridgehead	
Entry:	
Target:	(MilitaryOperation (who EnemyUnit) (why Secure) (where Bridgehead))
Setup:	
Intro:	If a red battalion has just crossed over the river in advance of a major red offensive, what do you think that battalion is now trying to do?
Hinting:	What would you do if you had just pushed a battalion into enemy territory across a bridge?
Leading:	Would it be reasonable for the red forces now east of the bridge to be trying to secure their bridgehead?
Success:	Absolutely. If we had just managed to push a battalion across a river into enemy territory, it's safe to say that we would be thinking about how to secure the bridgehead for follow-on forces..
Failure:	Actually I think if we had just managed to push a battalion across a river into enemy territory, it's safe to say that we would be thinking about how to secure the bridgehead for follow-on forces..

Figure 20. EOTB Dialog Node Dial_RedBridgehead.

The tutor utterance, "Let's start by talking about the red forces and their situation." is actually the Setup Presentation associated with the Dial_RedPicture node. From the flowchart in Figure 15 we see that so long as a Dialog node's target pattern is not satisfied before we start processing, then the first thing that is done is to produce the Setup Presentation. Since this node has no target pattern, it cannot be satisfied, and thus this Setup is used. Dial_RedPicture has no Introductory, Hinting, or Leading questions, but it does have a set of recursive Dialogs. Thus the next item in the transcript is an utterance from the first child node beneath Dial_RedPicture, which is Dial_RedBattalion.

The tutor utterance, "What sort of force do you think you're facing in your area east of the river?" is actually the Introductory question from Dial_RedBattalion. This is because this node has no Setup (actually, it has no Hinting question as well).

The student input, "Enemy Mech is located at area east of the river" is a narrow answer to the tutor's question. The student simply pointed to the only red icon on the map (which the tutor understands to be an enemy mech company). The target Pattern for this node specifies an enemy mech battalion, so this initial answer is not fully satisfactory. The node also has a set of targeted response, or disambiguation questions. One of these matches the student's answer enough to recognize that they have not provided the expected echelon or unit size.

The tutor utterance, "Can you characterize what sized unit you are seeing here?" results from the triggering of the Dial_RedBattalion disambiguation question. Other disambiguation questions trigger if the student says there is a battalion east of the river (but not that it is an enemy battalion), or if they say there is an enemy battalion (but not that it is east of the river).

The student input "bat is located at area east of the river" is interpreted as fleshing out the original answer, and so is interpreted as an "enemy battalion is located at area east of the river" which satisfies the node's Pattern. This input would make sense if the student used the tutor's

extra prompt as an opportunity to think a bit deeper about the situation and the question that was really being asked.

The result is that the tutor moves directly to the node's Success Summary: "Yes. Given that you've seen at least a company of enemy mech in what was supposed to be your assembly area, and given reports of at least 10 T-62s, plus the fact they've been moving over the bridge for at least 2 hours, it seems reasonable to assume that there's something on the order of a red battalion already east of the river." That constitutes the end of processing for the Dial_RedBattalion node, and the tutor moves on to the next node, Dial_RedOffensive.

We can consider some alternate student inputs that would lead to alternate tutor behavior. For instance, if the student had stuck with their initial shallow interpretation of the first question and responded to the disambiguation question with the answer "company is located at area east of the river," then Dial_RedBattalion's Pattern would not be satisfied and processing would continue. Since Dial_RedBattalion has no Hinting question, processing would move on to its recursive Dialog nodes: Dial_RedBattalion_MechCoInAA, Dial_RedBattalion_10T62sSeen, and Dial_RedBattalion_FlowingFor2Hrs. The idea is that by reviewing these three points of the initial situation description (ideally, getting the student to generate them), the argument for believing there is an enemy battalion east of the river will become apparent.

Tutor	Let's start by talking about the red forces and their situation.
Tutor	What sort of force do you think you're facing in your area east of the river?
Student	Enemy Mech is located at area east of the river.
Tutor	Can you characterize what sized unit you are seeing here?
Student	company is located at area east of the river.
Tutor	How much do you think you're facing in terms of armored forces?
Student	10.0 t-62s is located at area east of the river.
Tutor	Yeah, the host nation recon patrol said they had counted 10 T62s cross the bridge in the last half hour.
Tutor	How long has the enemy been moving forces across the bridge?
Student	Move has been going on for 2.0 Hour.
Tutor	Good. The host nation recon patrol reported they were forced back across the river under fire 2 hours ago. So enemy forces have been moving in for something like 2 hours.
Tutor	With reports of at least a company of mech, and most of a company of tanks, would you say you might be facing a battalion?
Student	Yes.
Tutor	Yes. Given that you've seen at least a company of enemy mech in what was supposed to be your assembly area, and given reports of at least 10 T-62s, plus the fact they've been moving over the bridge for at least 2 hours, it seems reasonable to assume that there's something on the order of a red battalion already east of the river.
Tutor	What do you think the enemy is up to?

When the tutor starts on these recursive nodes, its first utterance would be, "How much do you think you are facing in terms of armored forces?" This is the Introductory question for the second sub-node, Dial_RedBattalion_10T62sSeen (see Figure 22). The system effectively skips over the first sub-node Dial_RedBattalion_MechCoInAA (see Figure 21) because the student has already said that there is an enemy Mech company east of the river; that statement pre-satisfies the target Pattern for the node and so, following the flowchart, the system goes directly to the success end-state, skipping over all Presentations.

Dial_RedBattalion_MechCoInAA	
Entry:	
Target:	(AtLocation (subject (& Company (! Mechanized BMP))) (object =EastOfRiver))
Setup:	
Intro:	How much mech do you think you're facing?
Hinting:	What did your scouts tell you was sitting in your assembly area?
Leading:	
Success:	Right. The scouts said there was at least a company of enemy mech in your assembly area, possibly more moving into the area.
Failure:	What the scouts actually said was there was at least a company of enemy mech in your assembly area, and possibly more moving into the area.

Figure 21. EOTB Dialog Node Dial_RedBattalion_MechCoInAA.

Dial_RedBattalion_10T62sSeen	
Entry:	
Target:	(AtLocation (subject (! (& Company (! Armor Tank)) (Tank (count (> 9))))) (object =EastOfRiver))
Setup:	
Intro:	How much do you think you're facing in terms of armored forces?
Hinting:	What did the host nation recon unit tell you they had seen cross the bridge?
Leading:	
Success:	Yeah, the host nation recon patrol said they had counted 10 T62s cross the bridge in the last half hour..
Failure:	The host nation recon patrol actually reported that they had counted 10 T62s cross the bridge in the last half hour.

Figure 22. EOTB Dialog Node Dial_RedBattalion_10T62sSeen.

The student could answer a number of ways to this question. They could base their answer on the report from the scenario introduction and say, "*10 T-62s* is located at area east of the river." The system would also accept, "*10* tanks is located at area east of the river." Finally, the system will also accept the inference-based answer, "company is located at area east of the river" (interpreting that as meaning, "armor company is located at area east of the river").

Any of these answers would satisfy the node's Pattern[7] and cause the tutor to follow the flowchart path to use its Success Summary: "Yeah, the host nation recon patrol said they had counted 10 T62s cross the bridge in the last half hour." Other answers would lead the tutor to try again with its Hinting question, "What did the host nation recon unit tell you they had seen cross the bridge?" A correct answer here would lead to the same Success Summary. Since the node has no Leading question, an incorrect answer to this second question would lead to the Failure Summary: "The host nation recon patrol actually reported that they had counted 10 T62s cross the bridge in the last half hour."

However the node Dial_RedBattalion_10T62sSeen completes, the system will move on to the Dial_RedBattalion_FlowingFor2Hrs node (see Figure 23) and ask its Introductory question, "How long has the enemy been moving forces across the bridge?" Again, this is a relatively straightforward factual question, so we expect the student will get it right. Still, the system is prepared with a Hinting question and a Failure Summary as backups.

Dial_RedBattalion_FlowingFor2Hrs	
Entry:	
Target:	(Move (duration (?tu TimeUnit (amount ?n))) (.units ?n ?tu (> 1.9) =Hour))
Setup:	
Intro:	How long has the enemy been moving forces across the bridge??
Hinting:	The host nation recon patrol told you when they were forced back east over the bridge. So how long would you say the enemy has been coming across?
Leading:	
Success:	Good. The host nation recon patrol reported they were forced back across the river under fire 2 hours ago. So enemy forces have been moving in for something like 2 hours..
Failure:	Actually, the host nation recon patrol said they were forced back across the river under fire 2 hours ago. So enemy forces have been moving in for something like 2 hours..

Figure 23. EOTB Dialog Node Dial_RedBattalion_FlowingFor2Hrs.

Here we assume the student recalls the relevant fact from the introductory briefing and correctly answers, "Move has been going on for 2.0 hour." This elicits the tutor's Success Summary: "Good. The host nation recon patrol reported they were forced back across the river under fire 2 hours ago. So enemy forces have been moving in for something like 2 hours." The tutor has now completed processing the three recursive nodes beneath Dial_RedBattalion, and continues through the flowchart for that node.

[7] The system will actually accept too many things as "correct" answers. For instance, since the Pattern is specified at the level of "Tanks" it would accept "10 T-72s" as well as "10 T-62s" because it knows that T-72s are a kind of Tank, even though they don't appear in this Scenario. Likewise, it would accept "10 M1A1s" which are also a known kind of Tank (even though the M1A1s are our tanks). In fact, it will accept any number of tanks greater than 9. So if a student wanted to say there were "9.1 tanks" or "157 tanks" it would accept those answers as well. These deficiencies seem likely to become issues only if students start "gaming" the system rather than taking it seriously. Further, most of these issues can largely be addressed by more careful authoring, including tighter Pattern conditions and additional disambiguation questions. Experience with actual students is expected to reveal when this kind of sloppiness is likely to become an actual problem.

It finds the Leading question, "With reports of at least a company of mech, and most of a company of tanks, would you say you might be facing a battalion?" We assume the student agrees to this argument, answers "Yes." and receives the tutor Success Summary: "Yes. Given that you've seen at least a company of enemy mech in what was supposed to be your assembly area, and given reports of at least 10 T-62s, plus the fact they've been moving over the bridge for at least 2 hours, it seems reasonable to assume that there's something on the order of a red battalion already east of the river."

Of course it is possible the student is not convinced by the detailed analysis of the situation and insists on answering, "No." This is probably the situation where the system most seriously diverges from the behavior of live expert tutors. A clever person would be able to explore what the student was thinking and why they were not accepting what seems like a logical conclusion. Here, ComMentor simply has to insist on its interpretation and move on[8]: "Well, given that you've seen at least a company of enemy mech in what was supposed to be your assembly area, and given reports of at least 10 T-62s, plus the fact they've been moving over the bridge for at least 2 hours, it seems reasonable to assume that there's something on the order of a red battalion already east of the river." The question, "What do you think the enemy is up to?" represents the tutor moving on to the next node under Dial_RedPicture; it is the Introductory question from Dial_RedOffensive.

Having explored an example of Dialog node processing in detail, we can now turn to how larger-scale topics are introduced as various Evaluations are triggered. Here we first consider the discussions that follow from several Evaluations that fire during the transcript from Appendix C, and then look at alternate topics that might have been introduced by the tutor under different conditions.

Tutor	What sort of risks do you see in your move against the assembly area?
Tutor	Does a night fight in close quarters give you any qualms about the possibility of fratricide?
Student	Yes.
Tutor	Good call. The fact that you're going to be fighting in the dark, in an area that's small compared to the range of your tanks' main guns means that you need to give extra consideration to the danger of fratricide if you're going to be maneuvering forces in from more than one side.

This (condensed) discussion is tied to the Evaluation Eval_AttackingAAMultiCos. The tutor only delves into the possible risk of fratricide between Blue attacking forces because the student has tasked two different companies (including at least one tank company) to attack the assembly area. With only one company attacking, or two mech infantry companies attacking, this discussion would not be entered.

---

[8] Actually, if there were another argument for why the student ought to accept that the enemy probably had a battalion east of the river, it would be possible to have another Dialog node with the same target Patten ready to try again after Dial_RedBattalion. The system would try that other line of reasoning only in the case where the target Pattern had not been satisfied in the pursuit of Dial_RedBattalion. Likewise, if the SME considered that there was a plausible alternate interpretation of the situation, they could author a following Dialog node whose entry Pattern was the negation of Dial_RedBattalion's Pattern ready to pursue that possibility.

Tutor Where is your scout platoon?

Tutor Yes. Given their current location, your scouts are likely to get shot up by your own assault force.

Tutor Why don't you give the Scouts some orders about what they should be doing--especially where they should go.

This snippet is tied to the Evaluation Eval_AttackingAANoEvac (which is nested under Eval_AttackingAAAlongWhartonRd). The tutor only delves into this second fratricide risk if the student sends a company to attack the assembly area by a route that takes them down Wharton Rd, *and* if they also fail to issue any orders to their scouts to move out of the way (since they are currently sitting on Wharton Rd and are engaged with the enemy).

The snippet above ends with a prompt to issue orders that move the Scouts. A set of alternate Evaluations await such orders, activating different discussions depending on where the student sends the scouts. Appendix C shows a run through Eval_AttackingAAEvacScoutsSouth as synopsized below:

Student Who: Scout Platoon Where: Wharton Hill Route: road south of the assembly area, Wharton forest, and Wharton Hill.

Tutor What is the primary value of moving your scouts to the south?

Tutor You are coming from the east, and the enemy is certainly to the north. Where might the scouts develop new information?

Student Scout Platoon being at Wharton intersection TacticallyEnables Observe by Scout Platoon or (location) intersection south of bridge.

Tutor Yes. Moving your scouts to the west would also get them out of immediate harm's way and might help develop more situation awareness for you.

The tutor is, however, also ready with two other lines of discussion attached to the Evaluations Eval_AttackingAAEvacScoutsEast and Eval_AttackingAAEvacScoutsWest. If the student sends the scouts to the East, it is worth talking about the fact that they are being sent back into the path of the oncoming assault forces. If they are sent to the West it is worth noting that there is the potential for them to gather useful information about a possible route to the bridge (a point which is approached in a different way in the piece of Eval_AttackingAAEvacScoutsSouth above.

There are other points at which the EOTB Scenario includes Evaluations ready to launch discussions of likely alternatives. For instance, there is an Evaluation (not triggered in the Appendix C transcript) ready to discuss the student's rationale for taskings that call for an attack on the town of Hamlet itself. Likewise, there are a set of Evaluations tuned to respond to alternate COAs aimed at taking back the bridge. Appendix C includes a section associated with the Evaluation Eval_AttackingBridgeBySOfWharton, but two other lines of discussion are ready for students who send their forces by the Wharton Rd (Eval_AttackingBridgeByWharton) or by Shaffer Farm and Hill (Eval_AttackingBridgeByShaffer). Finally, there is an Evaluation that watches for students who don't consider the need to project forces across the bridge to establish a defensible bridgehead.

At both the Evaluation and Dialog levels ComMentor is capable of adapting to the behavior of the individual student. This section has presented examples intended to give some sense for how the system makes such decisions, and what the student experiences as a result. Developing tutoring scripts that produce appropriate behavior is heavily dependent on the authoring process which, in turn, relies on the authoring tools described in the next section.

Final Prototype Authoring Capabilities

The Authoring Process. Authoring a ComMentor scenario involves a basic sequence of core tasks, supplemented by a set of supporting tasks performed on an as-needed basis. The core task sequence is as follows:

1. Design the scenario problem and prepare basic media. A TDG scenario starts in the mind of an expert tactician and instructor, often based on some personal, anecdotal, or historical experience, or on a desire to highlight particular issues for students. The scenario must be fleshed out with a basic set of supporting media, in particular, a situation map and an introductory briefing. Typically the briefings run about 1 or 1-1/2 pages in length, however, in the case of FCS scenarios that involve a large amount of unfamiliar material, the briefing may run to 3 or 4 pages. The map should be a GIF or JPG file, preferably sized to something between 640x480 and 800x600 pixels. It may be a scan of an actual map or a hand (or computer) drawn map sketch. A set of military unit and/or vehicle icons must also be provided to place (and move) on the map. We estimate this task to take 0-3 days (0 in the case where the SME already has a relevant TDG lying around, as was often the case in this project).

2. Convert the introductory briefing to an on-line presentation. In the current system we use an HTML/GIF slide show with voice-over as the on-line media format for the initial scenario briefing. We create a series of GIF files by breaking the introductory briefing text into a set of PowerPoint slides that include graphics such as force-structure diagram and an evolving situation (map) display. For each slide, we create a WAV file recording a spoken rendition of the slide's text. We use the PowerPoint "Save As... (GIF)" option to turn the slides into a series of GIF slides. The GIF and WAV files are then linked into a sequenced presentation in ComMentor. This is accomplished most easily as one of the effects of using ComMentor's New Scenario Wizard, but the presentation can be wired together (or updated) by hand. We estimate the task of preparing the media to take 2-4 hours.

3. Create the new scenario in ComMentor (New Scenario Wizard). The ComMentor authoring suite includes a New Scenario Wizard to speed creation of the common structure required for most typical scenarios. In addition to wiring together the initial briefing presentation, this includes creating an appropriate directory structure, linking to the map graphic, creating an initial order-collection and discussion "scene," and establishing an appropriate sequence of startup events. The wizard also prompts the user to carry out the next two core tasks (but these can be done or refined separately later). Not counting the next two tasks, working through the basic wizard takes a few minutes.

4. Define the scenario force structures for Blue and Red. The force-structure tree displayed in ComMentor's runtime must be constructed as part of scenario authoring using the Situation Authoring Tool. The trees are built up using a conventional tree-structure GUI through

a series of right-click menu options and drag/drop operations. The attributes of each unit added to the tree can be controlled in detail by right-clicking to bring up an editing dialog. Important attributes include: internal (author visible) name, external (user visible) name, unit type, unit icon, and system responses to dragging or right clicking the unit's icon. For a typical scenario, it might take up to 2 hours to define the required force structures.

5. Annotate the map with meaningful regions. Initially, the map is just a raw graphic file so far as ComMentor is concerned. In order to enable the system to recognize when the student has pointed to (or placed forces on) map regions of particular interest, those regions must be defined on a set of map overlays, again authored using the Situation Authoring Tool. We recommend creating map layers for clusters of features such as relief, vegetation, water features, roads/intersections, cultural features, and large-scale regions. Each region is created by designating a polygon on an appropriate map layer (by a series of mouse clicks) and then filling in key attributes in a dialog: internal name, external name, and type. For a typical scenario, this task might also take up to 2 hours.

6. Define an initial set of scenario-specific Question Templates. One additional scenario set-up task which has not yet been incorporated into the New Scenario Wizard is the definition of a set of scenario-specific questions that the student can ask at any time for clarification. There is, however, a separate Scenario Question Wizard available from the Template Authoring Tool that makes it easy to link subsets of the slides from the introductory briefing as answers to questions about the scenario. With the help of the Wizard, this task should take less than 1 hour.

All of the initial design, media preparation, and situation definition tasks discussed so far are expected to take a total of 1-4 days.

7. Develop an initial set of discussion topics (Evaluations). Developing an initial discussion topics list for a TDG scenario involves extensive analysis of the particular scenario. In our experience, the best way to start off is to play through the scenario several times with live human students (we used 4 2-hour sessions, so counting author and student, that amounts to another 2 days of effort; we also found it very useful to record the sessions and to produce transcripts of the resulting tapes, which we will count as another 1 day of effort). An additional 1 day to review session transcripts is recommended. Finally, assuming on the order of 20 Evaluation nodes are to be authored, we expect (at 10 minutes per node) it would take well less than 4 hours to formally enter and organize the resulting Evaluation nodes in the Tutoring Authoring Tool. The only challenging aspects of this final step are (a) creating the Pattern that will serve as the trigger for the evaluation (usually referring primarily to aspects of possible student plans), and (b) choosing a set of Curriculum nodes representing topics that the Evaluation is going to stress. Either or both of these sub-steps may usefully be deferred until the Dialogs to be associated with the Evaluation have been more fully developed.

As with the design time required for Task 1, note that the several days of scenario play/transcripting/analysis effort can largely be avoided in the case where the author is working with a TDG that they have already developed and used extensively with students. This step then is expected to take 2-5 days of effort, bringing the running total to 3-9 days so far.

8. Fill in Dialogs for each topic. This is the most complex, time-consuming task, and calls for the most specialized (system-specific) analysis and knowledge. For each Evaluation, the author must create a tree of Dialog nodes that represent the major points the student ought to come to understand about the situation, typically organized as a kind of "argument" structure, where nodes lower in the tree cover more basic facts that together lead to the conclusions higher in the tree. For each Dialog node, the author must write a set of (up to six) questions/statements that will constitute tutor contributions to the dialog under certain conditions, annotate the questions with instructions for appropriate modifications to the runtime user interface, optionally develop a set of response questions, and then tag the entire Dialog node with Pattern(s) and Curriculum nodes (as for Evaluations, above). Assuming an average of 6 Dialog nodes per Evaluation node[9], and 20 minutes to author each Dialog node, we arrive at an estimate of 5 days, bringing our running total to 8-14 days.

9. Prepare supplementary presentations. This task is optional, and the amount of time devoted to it depends on how fancy the author wants to get about the tutor presentations that the student will see during the interactive discussion. An example of a supplementary presentation would be a custom map overlay that shows the tutor's estimates of where Red forces might be, to be linked with one of the normal questions/statements associated with some Dialog node that discusses Red force dispositions. So far we have little experience creating or linking such supplementary presentations, so we will simply estimate 1 additional day of effort devoted to this task.

10. Test and refine authored tutoring. The authoring tool suite integrates with the ComMentor runtime (and offers some extra debugging and control options), so that it is possible to test pieces of the tutoring structure as it is authored. We estimate that an author would typically want to spend up to 5 additional days (much of it interspersed with the earlier analysis and data entry tasks) testing and tweaking the tutoring. This is likely to lead to reorganization of Evaluation and Dialog node trees, revision of Patterns and Curriculum nodes assigned to such nodes, and tuning of questions/statements in Dialog nodes

Analysis of these 10 tasks allows us to estimate the core effort devoted to getting a new scenario into a state suitable for initial student use at 14-20 days—roughly 1 person-month of effort. An initial testing period with a set of sample students—say 10 students at 2 hours apiece—would likely reveal that majority of issues with the newly authored Scenario. If we allow 1-2 hours to review each sample transcript and up to 4 hours to address the issues it raises, we would allocate another 2 weeks for Scenario review, and refinement before release to the general population. The estimates so far also leave out a significant set of non-core tasks aimed at refining representations used across all scenarios: the domain ontology, input-form templates, and the curriculum (with accompanying presentations). Effort devoted to authoring such cross-scenario resources ought to decline over time (as more scenarios have been authored and most common requirements have been met). Still it seems prudent to allocate 2 additional weeks to

[9] At an advanced stage in the development of "Enemy Over the Bridge", there were 22 Evaluations defined with the following number of Dialog nodes (note that 5 Dialog nodes are used in two different Evaluations):

Evaluation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	Tot
Dialogs	1	1	51	0	4	0	4	0	3	3	3	6	4	6	4	5	7	6	3	7	13	3	134

such super-scenario enhancements.[10] That suggests as a final estimate, around 2 person-months of authoring effort per scenario, which at billing rates typical for the highly skilled personnel involved comes to approximately $50,000/scenario.

To complete the picture of the authoring process, we briefly characterize the tasks associated with authoring the cross-scenario representations:

- Extending/revising the ontology. As described in the section on the Domain Model, the ComMentor ontology is a large term hierarchy (with upward branching) that is used to (a) define symbols that serve to name concepts, (b) organize concepts according to common properties, (c) also organize them according to how they should appear in menus in the user interface, (d) store basic data about key categories of entities, and (e) attach sets of English-language strings that might be used by students (or authors) to describe entities in a scenario. Extending/revising the ontology then potentially involves editing any aspects of this data: creating/deleting concepts, adding/removing attributes of concepts or changing the constraints on existing attributes, moving/multiply-linking concepts within the hierarchy, adding or modifying data about particular concept classes, adding or modifying string patterns assigned to concept classes. ComMentor's Ontology Authoring Tool provides graphical user interface support for all these operations, as well as other related tasks. Despite the relative ease of use afforded by that GUI, we do not expect every SME/author to be comfortable modifying the ontology.

- Creating/modifying input-form templates. Anything that the student might need to say to the tutor must be expressible through some input-form, button, or menu. If an appropriate input mechanism does not already exist, then an author can use the Template Authoring Tool to develop the Template that defines a new input layout. Most often authors will be defining new forms to be added to the main menu structure of the tutor's input panel. For each such form, the author has to decide what the input fields are going to be (along with the type of the field, the type of the filler and other display information), what connecting text is going to appear to help unify the fields into a coherent form, and what internal representation is going to be created when the user fills and submits the form. Again, there is a relatively friendly GUI for this tool, but again, we do not necessarily expect every author/SME to feel fully comfortable authoring new forms.

- Extending/revising the curriculum. Evaluation and Dialog nodes can be annotated with Curriculum nodes to indicate what points are covered when the tutor executes those nodes during a student session—what points to credit or debit the student on if they trigger or bypass an Evaluation node, or if they succeed or fail at a Dialog node. The

[10] Note that modifications to representations that are used across the entire set of ComMentor scenarios are potentially dangerous, in the sense that they could break previously authored scenarios. Making such changes would require going back and testing old scenarios. One solution to this problem would be a versioning system for ComMentor KB contents, so that old scenarios could continue to rely on old versions of shared KBs. Under sponsorship of a separate DARPA project, a refined implementation of the underlying GRIST KB package is currently under development that provides initial support for versioning, as well as some support for client/server configurations that allow distributed collaborative work on KBs.

creation and organization (again into a multiple hierarchy) of Curriculum nodes is supported by the Curriculum Authoring Tool. In addition, the tool supports interlinking of Curriculum nodes with pre-requisite relationships, and the attachment to Curriculum nodes of Presentations labeled by roles such as introduction, example, anecdote, etc. Except perhaps for specification of complex prerequisite relationships, it seems likely that any author willing to accept and pursue the breakdown of overall tactical skill into a set of identifiable sub-skills[11] should feel comfortable exploring and extending the curriculum tree.

Authoring Tools. Having discussed the authoring process, we can now usefully review the authoring tool suite. Figure 24 shows the Launcher—the central window for dispatching to all of the individual authoring tools in the ComMentor authoring suite. As outlined in the sub-section on "Final Prototype Architecture and Design" there are six primary authoring tools with which we need to concern ourselves here: (1) the Ontology Authoring Tool, (2) the Presentation Authoring Tool, (3) the Situation Authoring Tool, (4) the Template Authoring Tool, (5) the Curriculum Authoring Tool, and (6) Tutoring Authoring Tool.

The Launcher is dominated by a two-column list that shows available tools to the left, and which of the loaded KBs that tool is currently focused on (the KB where editing done with that tool will register). As discussed in the sub-section on "Final Prototype Architecture and Design," one of the major differences between GRIST and Protégé is the ability to simultaneously hold multiple KBs in memory, and to instantly see the effects on a KB of edits to another "included" KB. The Launcher helps to manage this feature, as does the drop-down menu of available KBs that appears at the top of every tool (see for instance, Figure 25 below). In addition to basic file loading and saving, the Launcher also provides menus giving access to the available Validators and Wizards.

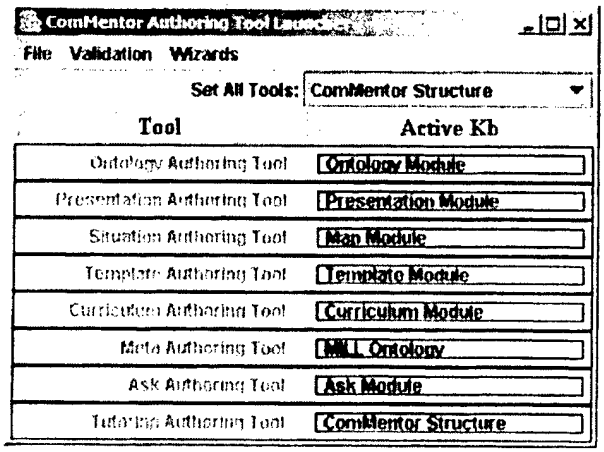


Figure 24. ComMentor's Authoring Tool Launcher Window.

[11] We note here that our primary SMEs on this project did *not* seem oriented towards analyzing tactical skill into discrete sub-skills. While they clearly had a tacit model of what made for a good tactical problem and good tactical instruction, their attitude towards evaluating students was more-or-less "I'll know a good tactician when I see one."

Figure 25 shows a representative screen shot of ComMentor's Ontology Authoring tool, displaying pieces of ComMentor's Military Domain Ontology including aspects of the system's vocabulary for describing "battlefield geometry" (tactically salient places), military relationships (emphasizing command and control relationships), and part of its list of scales or dimensions peculiar to the military world (e.g., damage assessments, military formations, probability of kill, etc.). As noted earlier, the ontology was developed specifically for ComMentor (drawing on earlier sources) but the authoring tool itself was built for a different project. This tool, which is patterned on the core Protégé knowledge entry tool is best developed for display and entry of frame types, but offers some rudimentary support for managing slots, instances, and rules as well (as suggested by the set of tabs near the top of the window).

The left column on this main "Types" tab contains a tree view that displays the type hierarchy and also supports editing of that hierarchy using drag-and-drop and right-click operations. Whichever concept is selected in the left column is shown in detail to the right. The detail view lays out according to the meta-type of the selected concept. The basic meta-type used in ComMentor allows each type to carry a common set of information, including for instance its doctrinal source (if available) and a set of English-language string patterns that should parse to the concept. Some concepts are assigned meta-types that enable them to carry extra information, for instance vehicle types can be assigned a range and a maximum speed (which can be used to answer student questions about the capabilities of particular types of vehicles).

The largest block in the right part of this screen is devoted to a table listing the slots and slot-constraints assigned to (or inherited by) the type being displayed. Given GRIST's native frame view of concepts, these Template-Slot definitions are critical to determining what kinds of assertions can be made about the concepts defined in the ontology. As in most such systems, it is possible to override inherited slot-constraints in order to make them more appropriate to the local context. Thus for instance a generic Action might take a generic Agent as the filler of its who slot, while a MilitaryOperation might require a MilitaryUnit as its who.

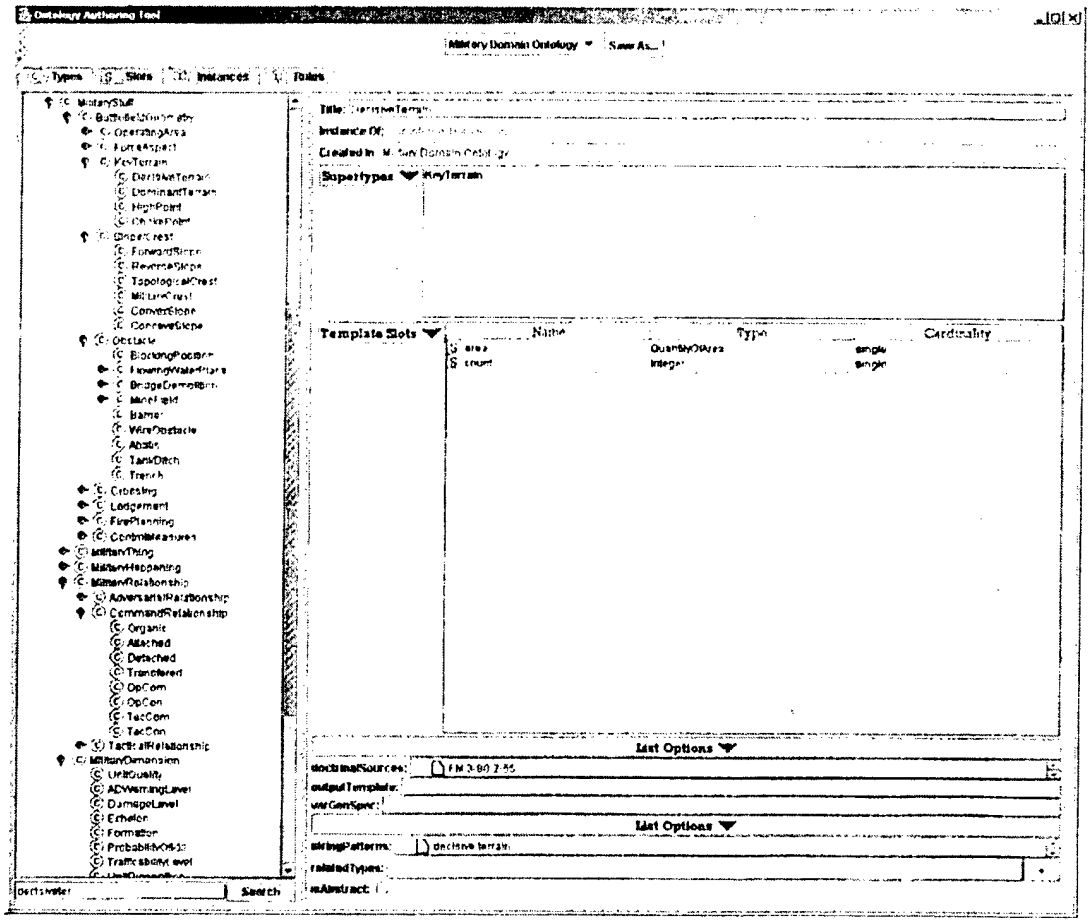


Figure 25. ComMentor's Ontology Authoring Tool Window.

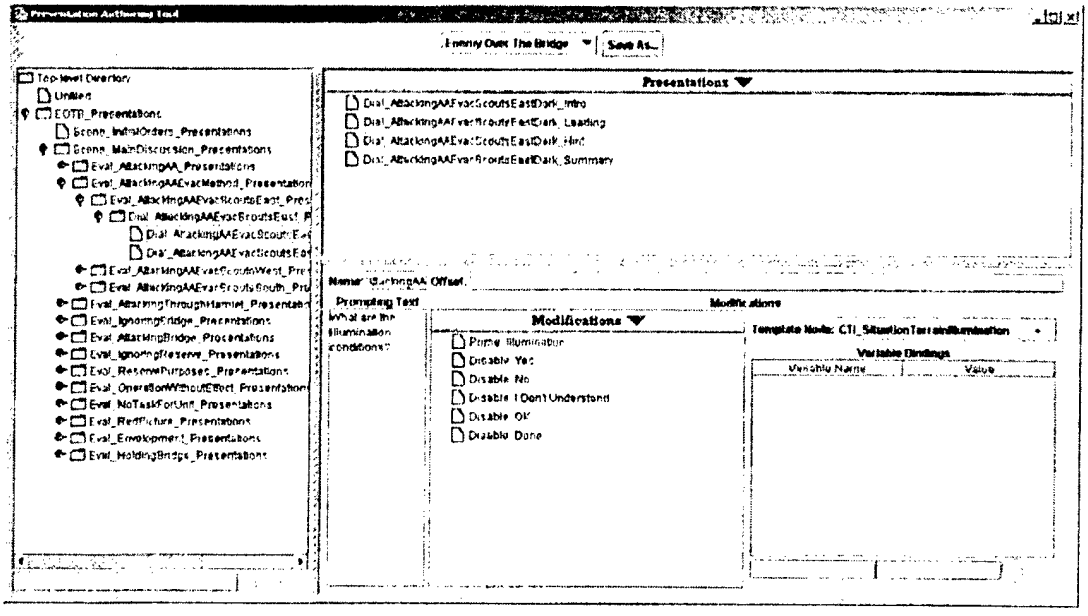


Figure 26. ComMentor's Presentation Authoring Tool Window.

Figure 26 shows ComMentor's Presentation Authoring tool displaying the hinting question Presentation from one of the Dialog nodes. Like the Ontology Authoring tool, this tool devotes its left column to a tree view useful for viewing (and searching for) the items it organizes—in this case Presentations instead of ontology Concepts. Here the left column is best thought of as a kind of filing system for Presentations, rather than as a hierarchical view of the Presentations themselves. In fact, the organization of this screen was borrowed from the 3-pane folder-tree/folder-contents/item layout used in many electronic mail programs. To the right of the tree view, the top pane shows the contents of the selected Presentation Directory and the bottom pane shows the details of the Presentation selected from that Directory. For Presentations created specifically to support Dialog nodes (which covers the bulk of the Presentations in ComMentor) the directory structure is created automatically to parallel the Evaluation and Dialog tree structures created in the Tutoring Authoring tool.

This particular screen shot shows that questions in Dialog nodes are actually compound objects built from a simple text Presentation ("What are the illumination conditions?") tied to a set of GUI modification Presentations. The GUI modification Presentations are responsible for arranging the forms-based input area in a way appropriate to the question. In this case, that means disabling most of the standard discourse buttons (you can't say "Yes" or "No" to this question), and priming the forms area with the form that provides a way for the student to describe the prevailing illumination conditions.

Figure 27 shows ComMentor's Situation Authoring tool, which, as suggested by the earlier discussion accompanying Figure 10, is one of the more customized tools in the suite. The custom aspects of a ComMentor situation include the map and its overlays, the force-structure elements and their wire-diagram relationships, and some day, a set of major event landmarks in the timeline display. These are the major domain-dependent visualizations available to both student and author to promote comprehension and expression of Scenario-relevant facts. The point of the Situation Authoring tool is to provide friendly ways to author facts about the Scenario using those visualization tools, as well as to author Presentations for use during tutoring on the Scenario that exploit those visualizations. The result is a set of editors that emphasize direct manipulation of the visualizations.

In the situation map pane the author can create, delete, and edit map layers, and control their visibility and focus. On a given map layer, the author can create, delete, and edit regions, as well as place and/or reposition force icons. Each region has a range of editable information associated with it, including an internal name (for use in specifying patterns), a set of conceptual types (for use in pattern matching), an external name (for generating tutor output), and a set of possible input string descriptions (for parsing student input), as well as a color and a set of Boolean properties controlling whether the region is deletable, selectable, user-visible, or included in descriptions of routes created when icons are dragged across the map at runtime. The left column of Figure 27 shows the region editor focused on the currently selected region: the assembly area (note the handles highlighting that region's border on the map).

In the force-structure wire-diagram pane the author can create, delete, and edit military units. Right-clicking on an item in the tree provides options to create a subordinate unit or edit the current unit. Units can be dragged around in the tree to change their relationships. In this case, editing a unit's properties is done in the pop-up dialog shown in Figure 28. Like map

regions, military units have an internal name, a set of conceptual types, an external name, and a set of possible input string descriptions. Units can be assigned an icon file if the automatic icon drawing code is not able to compose the correct icon based on the unit's assigned types. Units can also be assigned GUI actions (e.g., forms to be primed) when the icon corresponding to the unit is right-clicked or dragged on the map at runtime.

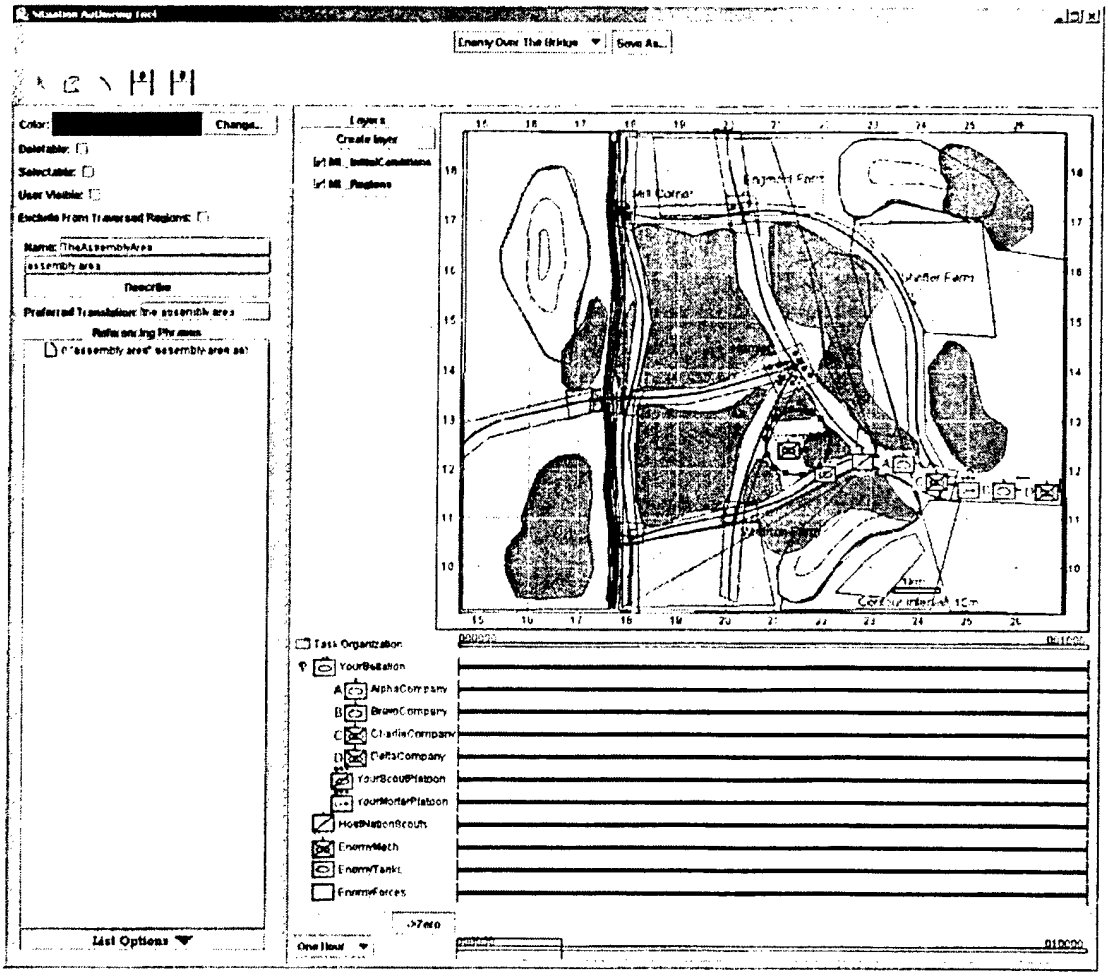


Figure 27. ComMentor's Situation Authoring Tool Window.

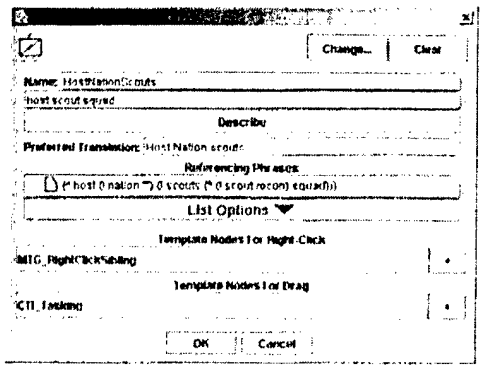


Figure 28. ComMentor's Situation Authoring Tool's Force-Structure Editing Dialog.

Figure 29 shows ComMentor's Template Authoring tool, displaying a major chunk from its main menu of student input forms in the left column, and the details of one selected form to the right—in this case, a form intended to describe where a military unit is located. In this window, the tree in the left column combines both the packaging structure and the bottommost packaged items (unlike the Presentation directory view). That is because the intermediate packaging structures ultimately appear to the end-user in the runtime GUI as menus or button-pallets, with their contents as menu-items (including nested menus) or buttons.

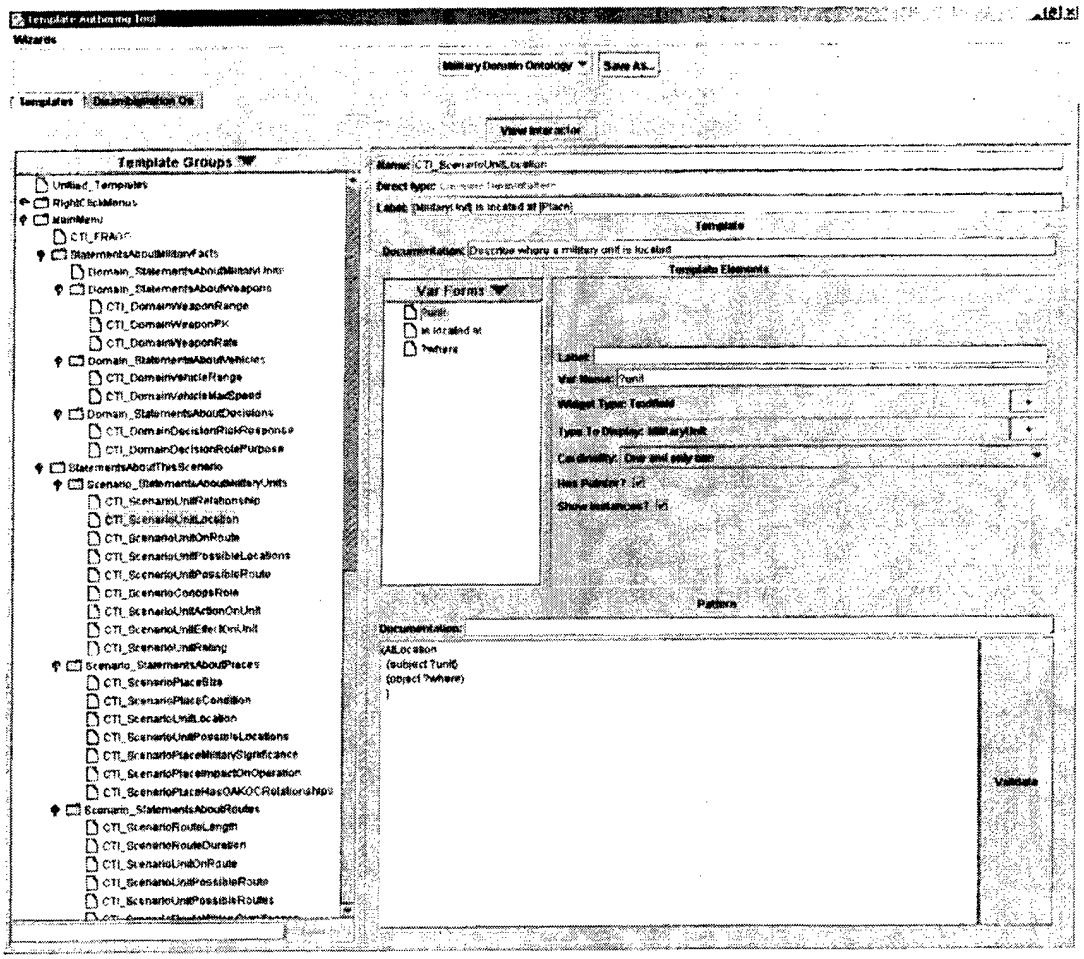


Figure 29. ComMentor's Template Authoring Tool Window.

The selected item in Figure 29 is itself the specification of an input form that contains two blanks and an intervening text string (to help tie the blanks together into a pidgin-English sentence). The column labeled "Var Forms" lists the pieces of the form, and the details of the selected item are shown to that column's right. The selected form element in Figure 29 is the first blank in the form. The details to the right indicate the name of the variable that will be bound internally to the user's input for that blank once they complete the form, what kind of interactive widget will be used to collect the data for that blank, which GRIST concept type is expected to fill the blank, how many fillers are required or allowed, whether or not the filler can be entered by pointing at one of the runtime visualizations, and whether the filler will ultimately be an instance of the specified type, or a subtype of that type. Finally, the bottommost part of the

window contains the pattern that will be fed to the GRIST pattern-driven structure creation mechanism (a make-pattern). Any variables appearing in that pattern should also appear in the Var Forms column, specifying where (from which form-blank) the variable is supposed to acquire its binding.

Figure 30 shows ComMentor's Curriculum Authoring tool. Again, the left column is devoted to a tree view that organizes the Curriculum points hierarchically. As with the type hierarchy in the Ontology Authoring tool, this network allows upward-branching (that is, a Curriculum point can appear in multiple places in the tree view). As usual, selecting an item in the left column causes the details of those objects to appear in the balance of the screen to the right. Curriculum points have two major pieces of information associated with them: a set of linked Presentations, and a prerequisite structure.

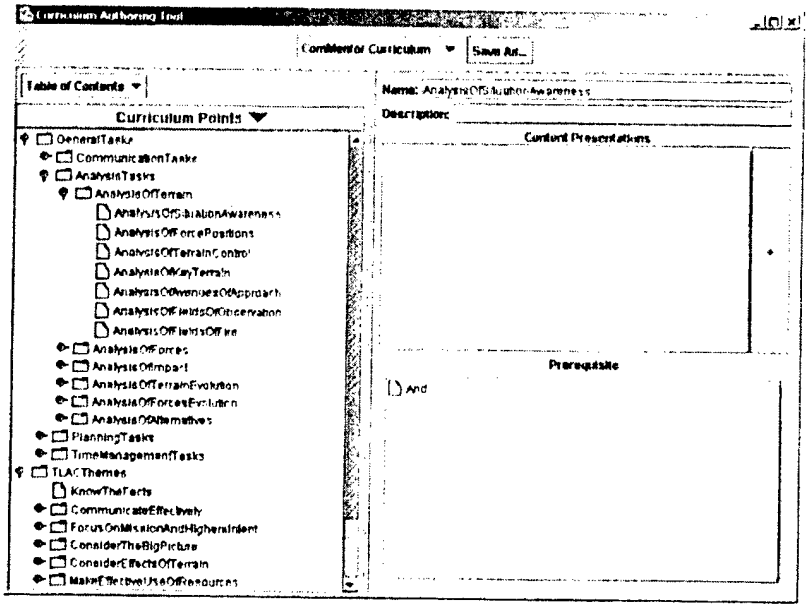


Figure 30. ComMentor's Curriculum Authoring Tool Window.

Figure 31 shows ComMentor's Tutoring Authoring tool. This is ultimately the heart of the authoring suite. Like the Situation Authoring tool, it is extensively customized to ComMentor's needs as a Socratic tutor relying on a particular style of representational structures and a particular set of algorithms. Instead of a single tree view filling a column on the left, this authoring tool uses a cascade of tree views across the top half of the window (similar in spirit to a SmallTalk style browser). The tab on the far left represents the Scenario. The first of three major columns represents the Scenes of the current Scenario. The middle column represents the Evaluation nodes within the currently selected Scene. The final column represents the Dialog nodes within the currently selected Evaluation.

What appears in the bottom half of the window depends on what is selected in the top half. There are layouts to display and edit Scenarios, Scenes, Evaluations, and Dialogs. Figure 31 shows the layout for a selected Dialog node. Figure 32 shows the layout for a selected Evaluation node. There are distinct layouts for each of the three kinds of Scenes that comprise a regular ComMentor Scenario.

The Dialog node display/editing layout in Figure 31 is the densest piece of layout in the system. It is roughly divided into a narrow left column that contains a mix of information—most notably the Curriculum points associated with the node, and the Patterns for entry and exit from the node—and a wider right column that contains room to author the six Presentations that can optionally be associated with a Dialog node: the Setup, the Introductory Question, the Hinting Question, the Leading Question, the Success Summary and the Failure Summary (see the discussion accompanying Figure 15). The three questions are compound Presentations that combine a text with a set of input-system Modifications. The other three are typically simple text Presentations.[12]

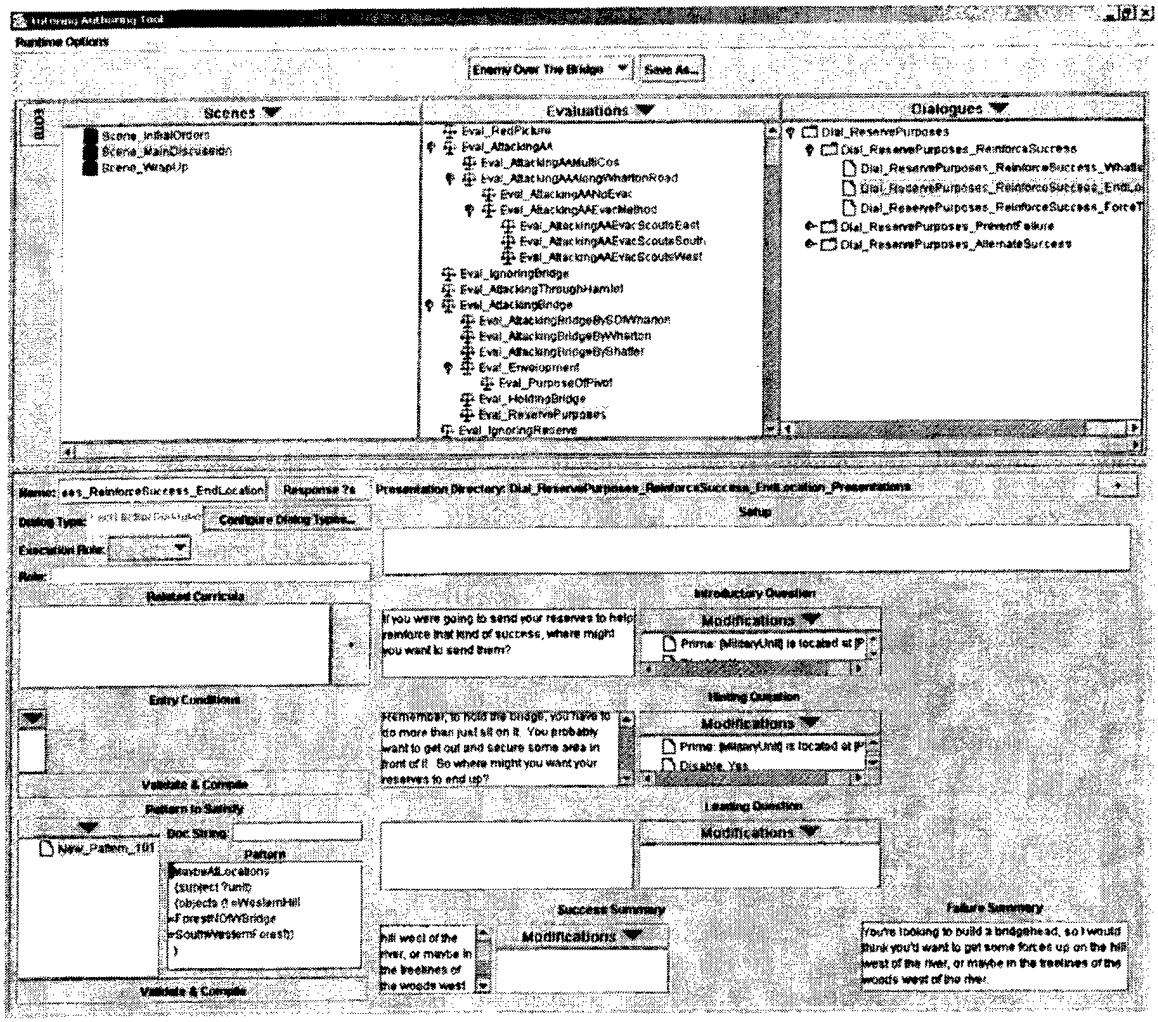


Figure 31. ComMentor's Tutoring Authoring Tool Window Focused on a Dialog Node.

[12] Figure 31 shows the Success Summary as a complex Presentation with Modifications. This is an artifact of the EOTB Scenario largely being authored with an earlier version of this tool that unnecessarily created the more complex Presentation type. Dialog nodes authored since the change-over to the new structure/layout will be simple text Presentations.

The Evaluation node display/editing layout in Figure 32 is substantially simpler than the Dialog node layout. It is comparable to just the left column of the Dialog display, primarily containing links to related Curriculum points and the Patterns that can trigger the Evaluation.

There is one other major piece of functionality embedded in the Tutoring Authoring tool: the ability to launch the runtime in order to explore the consequences of what has been authored. The right-click menu on the Scenario tab provides this option. The "Is Active?" Boolean checkbox on each Evaluation allows the author to control which evaluations could potentially fire during such a trial run. Menu-items in the Runtime Options menu offer a rapid way to enable or disable all Evaluations, and also provide the option of turning on rule and agenda tracing to help debug any strange tutor behavior. The net result of these embedded runtime and debugging tools is a substantially accelerated tutoring development process.

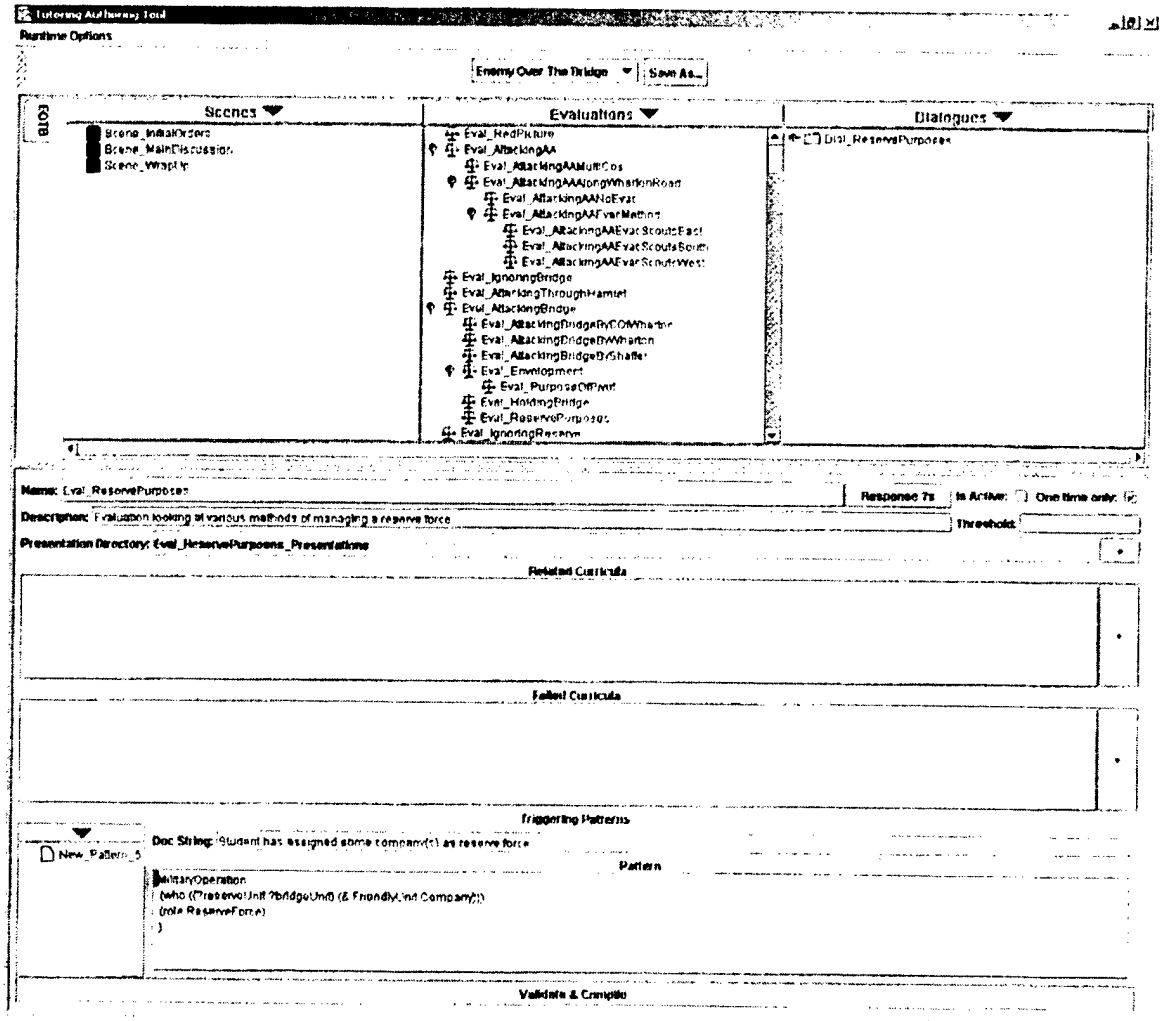


Figure 32. ComMentor's Tutoring Authoring Tool Window Focused on an Evaluation Node.

Authoring Trials. In September 2003, we ran a small trial exploring end-user authoring of a ComMentor scenario with one of our consulting SMEs. Over the course of five days our SME devoted approximately 20 hours to authoring pieces of the "Clash at Timpan-ni" (CAT)

Scenario.[13] A major caveat affecting this study is that the authoring tools, though they had been in active use for some time inside Stottler Henke, were still under development and were thus subject to various bugs and functional limitations. In addition, there was no written documentation available, so our SME's introduction to the tools constituted verbal briefings, demonstrations, and collaborative work on pieces of the CAT Scenario.

For this Scenario, the first step "Design the scenario problem and prepare basic media" had long ago been completed. Accordingly, our SME started on the second step, "Convert the introductory briefing to an on-line presentation." He based his work on the existing Scenario write-up and a PowerPoint file previously prepared for another Scenario. This task (along with gathering up the previously prepared graphics for map and icons) was largely carried out in about 2-3 hours (within the estimates from our earlier section) and constituted our first day's work on the preparing the Scenario. Our SME did not create the WAV files, because he did not have a reasonable microphone, and because the relatively mechanical task of narrating the newly created slides was considered a poor use of our limited time.

On the second day of the exercise, with guidance from Stottler Henke staff, our SME worked through the next three steps "Create the new scenario in ComMentor" (using the New Scenario Wizard), "Define the scenario force structures for Blue and Red," and "Annotate the map with meaningful regions." We spent about 1 hour working together through the New Scenario Wizard and definition of the Blue forces. After that our SME was able to create the Red force structure on his own in less than 1 hour, though he ran into some problems with the authoring tool's screen layout (the lack of a scroll-bar in the force-structure pane) that, in combination with the Scenario's very large map, made it difficult for him to see and create forces in the appropriate order. Also, a glitch in completion of the New Scenario Wizard caused us to have to edit the resulting KB data by hand to patch a missing link between the Scenario proper and some of the related structures.

In the afternoon of the second day, while we worked on small fixes to the authoring tool, our SME spent about 2-3 hours studying the transcripts of the live tutoring sessions from a year-and-a-half earlier. During that time, we worked to fix authoring tool issues that had arisen (e.g., bugs in the map pane's ability to manage layers, and slowness in some screen refreshes). Once we turned the tools back to him and demonstrated creation of map layers and regions, our SME was able to complete annotation of the map in about 1-½ hours.

The third day was a short one in which the SME spent 3-4 hours continuing to review the old session transcripts. Because we skipped having him use the Scenario Question Wizard to execute the "Define an initial set of scenario-specific Question Templates" step (again in an effort to conserve his time for the more important Evaluation and Dialog authoring steps), by the end of the third day we had completed all of the preparatory steps. In total, our SME had spent

[13] During those five days our SME put in approximately 30 hours overall, but approximately 10 hours were devoted to other discussions of system design, runtime behavior, ontology issues, or to procedural matters and time lost to bugs encountered and eventual fixes. The estimate of 20 hours on the Scenario authoring task includes perhaps 3 hours being directly coached and supported (over the phone and sometimes using NetMeeting for remote desktop sharing) by Stottler Henke personnel, as well as a couple of hours duplicating authoring work that was lost in a system malfunction. The details are presented in the narrative below.

about 6 hours creating materials for the new Scenario and 6 hours reviewing old transcripts in preparation for the tutoring authoring work to come.

On the fourth day we started by spending ½ an hour talking about the interesting points of the transcripts and ½ an hour getting set up with the latest release of the authoring tool and NetMeeting. We then spent 1-½ hours working together on the first set of Evaluations and Dialog nodes. We started with a discussion of major phases that had been typical in the discussion found in the live session transcripts, and of the major classes of responses that students had offered. That discussion allowed us to sketch out a first six Evaluations intended to serve as the backbone of tutoring authoring for the Scenario. The Evaluation data entry took only a few minutes, in large part because we did not immediately enter the trigger patterns. The bulk of our time was spent defining the first few Dialog nodes for the first Evaluation topic—the discussion of the Red situation. From our experience with other Scenarios, we expected this to be one of the most elaborate sets of Dialogs (e.g., in EOTB, the corresponding Red situation discussion involved about 50 nodes—more than any other Evaluation by a factor of 4).

The Dialog node screen has many pieces, and a reasonable understanding of the tutor's algorithm is required in order to build a good mental model of how those pieces should be filled in. A number of useful lessons and suggestions derived from this session (as well as significant side discussion on representational issues bearing on the underlying domain ontology).

- Our SME requested a visualization tool to clarify the order in which Dialog nodes and especially their various Presentations would be encountered by the student.

- Our SME requested that there be a way to give students feedback not just on right and wrong answers, but on partially right answers. Such a mechanism was under design at the time and has since been implemented.

- Our SME recommended a processing mechanism for option→pro/con discussions. Most aspects of the required control are expressible with existing ComMentor capabilities, but the pattern should be packaged in a more usable format.

- Our SME made clear that, as we had expected, he was not comfortable authoring Pattern specifications in ComMentor's pattern language. For purposes of this exercise, to keep making progress and explore issues bearing on other aspects of authoring, we reserved pattern development for a member of Stottler Henke staff. We have developed plans for easing initial pattern entry (see "Future Research and Refinement" sub-section on "Authoring Tool Refinements"), but the issue of generalizing and polishing the final pattern remains a challenge.

- We observed that there were some simple convenience features we could add to the Dialog Node screen to reduce redundant data entry (e.g., automatically generating sensible names for new nodes, and automatically copying Modifications from the initial Introductory Question to the Hinting Question). Those have not been implemented.

- We identified several representation issues, including how to represent places that do not have some property, how to represent coordinated action between multiple units, and how to express the properties of a good coordinated attack.

After our 1-½ hours working together, the SME worked for another 1-½ hours on his own elaborating on the Dialogs for the initial Evaluation topic. Unfortunately, at the end of that period, he experienced an authoring tool crash and lost his work. The balance of the afternoon (about 3 hours) was spent by the SME working solo on Dialog authoring to reconstruct and extend what had been done earlier in the day.[14] While annoying and regrettable, the loss of the earlier Dialog work did allow us to end up with a version of the Red situation discussion that was solely authored by the SME, rather than including the earlier contributions of Stottler Henke staff from the phase where they were introducing the SME to the authoring tools and process.

Finally, the fifth day of our initial study was again a short day. The first hour was spent reviewing the SMEs work from the previous day and discussing additional representational issues. The remaining 1-½ hours was spent with the SME again working solo, continuing to flesh out the Red situation Evaluation discussion. By the end of the first 5-day exercise, our SME had spent 6 hours preparing the Scenario basics, 6 hours reviewing live session transcripts, and 8 hours actually authoring Evaluations and Dialogs (including perhaps 2 hours of lost work due to a system crash).

After a two-week gap, our SME was again available to pursue a couple of additional hours of authoring for the CAT Scenario. In the interim, there had been additional enhancements to the authoring tools—in particular, the introduction at Dialog nodes of separate success and failure summary Presentations, and the addition of mechanisms for authoring contextually appropriate response questions at Evaluation and Dialog nodes. In this last day we concentrated on using the embedded runtime tools to experiment with and refine the pieces of Dialog that had already been authored, and to experiment with more responsive Evaluation structures. Again, a number of lessons and insights were generated by this session, including some new conventions or guidance for authors that should lead to better tutoring in the system:

- One authoring issue is finding ways to make the tutor's utterances feel more connected to one another when processing switches from one Dialog node to another. When entering a new node, the tutor's questions can be made more contextually appropriate by paying attention to which other nodes must have been explored beforehand, versus which other nodes might have been touched upon; the rule in the common case is that sibling nodes listed earlier will have been discussed, whereas later siblings and children of the earlier ones may not have been. When returning to an old node after descent into a nested set of elaboration nodes, again it is important to keep in mind what in those nested nodes must have been discussed; here the common case is that the immediate children will have been discussed, whereas nodes nested further below those children may not have been.

- A second authoring issue is to find ways to make the tutor's line of argumentation more tightly responsive to the student's utterances. We experimented in the case of the Red

[14] This experience suggested yet another possible system improvement—the addition of an auto-save backup feature to protect against extensive data loss due to system instability.

situation discussions for CAT and developed a structure using nested Evaluations that appeared likely to produce the desired effect. A small Dialog associated with a top-level Evaluation asks a relatively open question, and then a series of nested Evaluations are ready to respond to different aspects of different kinds of possible answers. This is likely to require extra authoring effort, but some authoring tool features exist (in preliminary form) to speed the process (e.g., copy and paste of Dialog sub-trees).

- We also hit upon a realization about the structure of ComMentor's Dialog nodes: the recent introduction of a more flexible follow-up question capability may come close to making unnecessary the inclusion of the "Hinting Question" in the basic Dialog node layout (and thus its processing in the basic Dialog node algorithm of Figure 15). There is the potential here for a simplification and generalization of the system's capabilities.

- Our SME again offered useful feedback on the authoring tools. The ability to launch the runtime from the Tutoring Authoring tool was considered a very useful feature. Additional features would be useful, for instance the ability to back up while running a Scenario so as to retry part of the interaction after a modification, or to test an alternate path. Also better support for copy/paste and related tree editing would be very helpful. Our SME remains concerned about the amount of authoring effort required to build the tutoring for a Scenario, and we agree, but see the current state of the system as a necessary phase in the evolution of our understanding of what a Socratic tutor must be able to do. Only once we can get the computer to produce appropriate behavior with heavy human scripting, and only when we are able to review accumulated Scenarios to better characterize the properties of effective tutoring scripts, can we plausibly aim to transfer more of the burden to the authoring tools or the runtime system.

- Finally we (again) had some interesting discussions about the system's mechanisms for collecting student input. There were some minor annoyances such as an inability to parse some noun-phrases our SME wanted to use, and discovery of a related bug that caused system lock-up. The interesting discussion, however, focused on the general question of how the affordances of our forms-based interface were different from normal person-to-person verbal dialog. In particular, our SME identified two major issues that can actually affect tutor behavior: (1) normal dialog allows the student to be more vague, or ambiguous, and to pack more alternatives into an utterance, and (2) normal dialog allows the student to hedge. Actually, ComMentor's forms are in some ways pretty good at allowing vague (general as opposed to specific) input, and we have recently introduced a limited capability to allow students to say multiple things on one turn; however, the fact remains that these features are harder to use than in speaking, and we also believe that the very fact of input being solicited through a computer-based form encourages users to assume a requirement for completeness and specificity. On the issue of hedging, we have started to consider ways to introduce such a capability into our forms.

Future Research and Refinement

In this section we devote a subsection to each of the following seven issues related to the future of ComMentor: (1) "Runtime Effectiveness and Evaluation," (2) "Cost/Benefit Studies of

Authoring/Tutoring," (3) "Authoring Tool Refinements," (4) "Strengthening and Generalization of Domain and Tutoring Knowledge," (5) "Applications for Project Infrastructure," (6) "Transition of ComMentor to Army Applications," and (7) "Other Possible Markets for High-Level Socratic Tutors," In addition, Appendix B, "Detailed List of Potential System Refinements" provides an extensive list of possible future developments.

Runtime Effectiveness and Evaluation

The most important piece of future work is to get a better sense of what in the current runtime system is working well, and what is working poorly, from the perspectives of student satisfaction and learning effectiveness. Insight into these questions could help inform decisions as to whether and how to push further development of ComMentor. It should be possible to pursue the answers we need by working with a small group of test subjects (between 6 and 12) using the current system, with no further development of software or content.

Assessment of student performance by more senior colleagues (ideally compared to prior performance) is probably the most revealing and relevant sort of test to which a training system like ComMentor can be subjected. That sort of evaluation raises many issues such as (inter) rater reliability, cost and availability of judges, and specificity/magnitude of any potential effect. We are, however, not aware of any more affordable solid metrics of learning effectiveness for battlefield command reasoning skills, or professional-level decision skills generally.

It may be possible to build instruments to measure in relative isolation some of the component skills believed to be critical to battlefield command reasoning (i.e., those identified in the ComMentor Curriculum model and labeled as covered by particular ComMentor scenarios). For instance, experts might develop a set of TDG-like set-ups and students might be required to characterize the opposing Red forces with respect to their type, size, disposition, and intent. An answer key that rates the typical range of student responses could probably be prepared; in effect that is one kind of information that is typically encoded in ComMentor scenario representations. Turning this approach into an actual formal evaluation might require authoring of additional ComMentor scenarios, but if evaluation was focused on a small set of issues (such as Red force assessment), it might be possible to quickly author the tutoring structures for a small subset of what might normally be in a full scenario.

It is much easier to invent assessments that address issues of ComMentor's usability, and that can more directly guide future development by suggesting opportunities for improvement. How usable is the latest iteration of the user interface? How comprehensible, expressive, and well organized are the current set of input templates? How coherent and useful are the tutor's general lines of discussion? How comprehensible and appropriate are the tutor's specific utterances? How well is the tutor doing on such specific functions as disambiguation or correction of student inputs, or on providing incremental feedback on student performance? Does the current mechanism and inventory of digression questions provide sufficient student initiative and situation clarification? Pursuit of such questions does not directly address the top-level issue of training effectiveness, but it is much easier to develop usable instruments for these questions and to get answers that can guide us towards a better system.

We also hypothesize that at a high level, student ratings of ComMentor are likely to be a good proxy for whether something was actually learned. Army captains are mature professionals highly motivated to master the skills that will ensure their success in battle. While there are clearly limits to introspection and reflection, we expect they are generally good judges of whether an experience has helped them learn something. The on-line questionnaire embedded in each scenario's closing reflective session should provide much of the most critical feedback we need on system performance.

Cost/Benefit Studies of Authoring/Tutoring

Assuming the results of the first evaluation study are promising, a second useful follow-on study would aim to firm up cost and quality data. With regard to authoring costs, we have arrived at an estimate of approximately $50,000 to build a new Scenario from scratch, to get it to the point where it has been tested by an initial set of users and revised accordingly. The extensive set of potential authoring tool improvements discussed in the next subsection and in Appendix B suggests that these costs might be lowered somewhat through more efficient tools, or alternately, that it might be possible to increase tutoring quality by getting more done for the same money. We also expect efficiencies will ultimately be gained from having a small set of experienced authors rather than individuals who are always new to the process.

To help resolve some of this speculation, it would be useful to observe authoring of a corpus of 6-12 scenarios under conditions that better approximate those expected to obtain in a production authoring environment. Deciding on exactly what those conditions should be, and making sure all the conditions are in place to establish them would likely require some additional research and development effort. For instance, which of the proposed authoring enhancements referenced above (or in more detail in Appendix B) ought we to get into place before we attempt even this limited scaling up and cost-baseline establishment study?

Ultimately, in addition to authoring costs, we will have to consider maintenance costs. One possible model would limit Scenario modifications, once released, to pure bug-fixes. We do not have a good basis for estimating the defect rate in Scenarios that have been authored and tested as described. Nor do we have a good basis for estimating the costs of fixing such defects. The most likely source of such quality data would be an extension to the authoring exercise just sketched. In this extension, authoring would be followed by a comparable runtime evaluation exercise with several students using each Scenario. Analysis of the resulting transcripts would then include a search for problems or disfluencies traceable to errors during authoring.

An alternate maintenance model would involve updating Scenarios from time-to-time to keep up with changes in doctrine or facts of the world. From the perspective of pure tactical reasoning, we might expect that good scenarios in some sense never go out of date. We can still learn valuable lessons by putting ourselves in Caesar's, Napoleon's or Washington's shoes. On the other hand, up-to-date scenarios provide opportunities to practice applying a broad range of specific doctrinal and factual skills and information that are not nearly so timeless as the core of battlefield reasoning. A reasonable stance might be to expect most Scenarios to stay unchanged most of the time, but that major revisions to tactical doctrine (such as those likely to accompany introduction of FCS) might trigger review and revision of old scenarios or authoring of a corpus of new scenarios. To explore this model, we could create an exercise where an existing scenario

(e.g., "Enemy Over the Bridge") is updated to accommodate a doctrinal or world change (e.g., a shift to the Interim Brigade concept).[15]

Finally, when it comes to quantifying benefits, we are up against an even harder version of the issues raised in the "Runtime Effectiveness and Evaluation" subsection above—not only to show benefits, but to quantify them in terms of dollars. Our best tack here may be using the justifications for other training initiatives as guides and baselines, and soliciting testimonials from users and their superiors. The more specific steps suggested in the earlier paragraphs of this section should provide ample examples and exposure to win such endorsements.

Authoring Tool Refinements

The existing authoring tool suite offers a large number of useful features in a highly integrated framework. Furthermore, that framework—based on the GRIST and MILL infrastructure layers—makes it relatively easy to modify the tools suite and add new features. We have envisioned a large number of possible refinements to the authoring tools. A detailed list is presented in Appendix B. Here we focus on a higher-level characterization that touches on a smaller set of more challenging or pervasive extensions, likely to significantly change the feel and efficiency of ComMentor authoring.

- Making it easier to author Evaluation/Dialog trees – When faced with the task of authoring a complex structure it is almost always easier to select an existing structure that is close to what you want and then tweak it until it is exactly what you want.[16] We expect there will be recurring patterns of Evaluations and Dialogs within and across Scenarios, and we would like to move beyond the current limited capability to copy, paste, and edit Evaluation and Dialogs in the Tutoring authoring tool. Right now we can associate Presentations with Curriculum points, but what we would like to be able to do is similarly associate generalized Evaluations and Discussions. Then, like the Curriculum, these tutoring structures would be accessible when authoring any Scenario. To further strengthen the system, we would have to add new representation that highlights the generalization points of these reusable objects, and ideally built-in procedural support for specializing those generalized points to a new context.

- Making it easier to enter Scenario-specific knowledge – Currently, there is no place to enter the tutor's beliefs about a Scenario or situations that arise within a Scenario. That knowledge is implicit in the structure of Dialog nodes that ask particular questions and seek for particular patterns of answers from the student. Supporting such situational representation, and furthermore gathering it and presenting it in a comprehensive and consistent way would potentially make it easier to author many classes of Dialogs (e.g., enumeration of places the enemy could be where they could fire on you). Knowledge gathered for one purpose could then more easily be applied for other purposes that arise in the tutor's discussions.

---

[15] Note that some changes in doctrine or world facts are significant enough as to undermine or completely destroy the plausibility and usefulness of a scenario. The FCS capabilities so change what a commander can know and do that the "Enemy Over the Bridge" map and situation simply might not make sense any more.

[16] This is essentially the core insight underlying the entire Case Based Reasoning approach within AI.

- Making it easier to author Patterns – In the current authoring suite, Patterns are by far the most technically complex thing that users are called upon to author. There is a wide range of enhancements that could be made to ease this difficulty. The most effective is likely to be combination of (1) a small modification that allows the standard ComMentor user interface (e.g., the existing template-driven forms-based interface) to output simple Patterns reflecting author input (e.g., Patterns without any disjunction or negation), and (2) a syntax-driven editor for the Pattern language that allows modification or even aggregation of such simple patterns to better capture a range of likely or acceptable student inputs.

Strengthening and Generalization of Domain and Tutoring Knowledge

In the current implementation, ComMentor applies its general domain knowledge in a few quite specific ways. The domain ontology shapes the representation of all the authored patterns and user inputs for all Scenarios, and the set of general input templates determines what is easily sayable across Scenarios. Likewise the ontology of Scenarios, Scenes, Evaluations, and Dialogs drives the overall tutoring process, with subtypes and patterns serving to speed configuration of recurring patterns of behavior. Beyond enforcement of basic semantic constraints across these representations, the system does small amounts of inference on tasks such as determining possible metonymy relations among places and military forces, conversion among units of measurement, and filling in missing input for a limited range of common-sense situations—for instance when a task implies an intent.

Some of the suggestions for improved authoring hinge on strengthening the system's exploitation of its representations, in particular (a) Place/Unit property/relationship matrices, (b) reusable (specializable) Evaluation/Dialog trees, and (c) Evaluation/Dialog-specific knowledge elicitation dialogs. Stronger domain knowledge can be applied at either authoring or runtime. Runtime inference may eliminate the need for explicit authoring of tutor reasoning and behavior that is not actually Scenario-specific. Alternately, authoring-time inference can enable the tools to prompt the author with plausible material that, as an expert practitioner, they can then accept, modify, or reject for inclusion in the Scenario script. Using inference for authoring support has the two significant advantages that it can save the author from having to start a given piece of work from a blank slate, and it allows the author to catch inappropriate inferences before they mar the instruction provided to an actual student (who may not be in as good a position to judge the inappropriateness of the general inference to the specific situation).

The basic strategy here is inductive—to be driven by regularities that appear in the data, where the relevant data is the materials that are authored for a growing set of Scenarios. Once we see patterns repeating across Scenarios we have warrant to start looking for ways to encode the regularity to the benefit of future authors. Pursuing this strategy will raise the cost of authoring early scenarios, but is potentially justified by savings in the authoring of further related Scenarios, or by gains in understanding of Scenario-based instruction in general.

Applications for Project Infrastructure

As noted earlier, much of the software infrastructure developed during the ComMentor project has already found application in other projects. Here we refer specifically to the GRIST,

MILL, and GRAIN layers (see "Results" subsection "Final Prototype Architecture and Design"). We expect to get tremendous use of these packages going forward. Furthermore, we expect them to see continued development in the future, making it possible for ComMentor to effectively gain new or improved capabilities for free or at substantially reduced cost (e.g., the example of concurrent distributed multi-user authoring cited in Appendix B).

Broadly speaking, GRIST is a useful package for any knowledge-based application where symbolic representation and reasoning is appropriate. Likewise, MILL is a useful package for any such knowledge-based system that has a significant GUI component. Finally, GRAIN is a useful package for any ITS application that is conceived along the lines of typical Stottler Henke designs. All of these packages are written in Java and thus apply most directly in the context of Java development, but in a separate ongoing project we are developing experience applying these packages in the context of distributed multi-environment Web services.

Transition of ComMentor to Army Applications

An excellent path to advance transition of ComMentor to Army application would be to continue working with ARI and to begin forging contacts with the Fort Knox Armor School. Experimental use with officers attending the Armor Captain's Career Course would start to provide some of the data that is needed to make the case for further development and broader experimentation. If funding can be found to pursue initial system evaluation, preparation of a more substantial Scenario corpus, and large scale cost/benefit evaluation, and if the outcome is positive, then there is tremendous scope for scaling up and broadening the range of ComMentor's application in the U.S. Army and the larger U.S. DoD community.

The level of interaction sophistication that ComMentor is currently capable of supporting is probably most appropriate to junior grade officers. Between the active duty Army, the Army National Guard, and the Army Reserve there are approximately 62,000 Captains and Lieutenants. The majority of these officers could benefit from the sort of TDG-based training we observed and ComMentor seeks to duplicate.[17] Assuming ComMentor becomes an ongoing part of their continuing professional development, the software, broadened with scenarios to cover infantry, engineering, and so on, could be providing hundreds of thousands of Scenario sessions per year in the Army alone. Clearly, accomplishing this goal would require a lot of additional work and refinement, but we remain optimistic that the goal is achievable (see "Conclusions" subsection "Prospects for Battlefield Command Reasoning ITSs").

Other Possible Markets for High-Level Socratic Tutors

As suggested elsewhere in this report, we believe a reasonable way to characterize what ComMentor is about is to couch it in terms of professional-level decision-making—reasoning skills employed on complex problems in open-ended domains where simple rule-application does not suffice to produce acceptable performance because context is so critical. In these domains, common sense and common practice suggest that Socratic exploration of possible courses of action and the rationale for alternate decisions in particular cases is an excellent way to facilitate improvements in performance.

[17] In the course of this project we met a wide range of Captains, including a non-negligible number who seemed to have no tactical background, nor much need for tactical training, e.g., medical or logistics specialists.

This framing suggests that the basic technology (if not the particular representations of domain and curriculum, or the user interface components such as maps and timelines) should apply in domains as varied as business, medicine, law, engineering, and education. Social values and structures being what they are, we foresee our best opportunities for further development of the technology in military and medical applications. We suspect the best opportunities to profit from a more fully developed and proven technology may be found in business applications.

Conclusions

This Conclusion section starts by reviewing project goals and accomplishments. In addition it discusses one of the major issues that arose in the course of the project: the problem of designing an effective input mechanism—in particular one that makes effective use of natural language. Finally, we close with an assessment of the prospects for battlefield command reasoning ITSs.

Project Goals vs. Project Accomplishments

In this sub-section we characterize our project accomplishments in light of the original eight project objectives (see Table 2 for a summary). We believe we have done a solid job on all objectives save item 6. We have also satisfied at least one additional objective that was not stated in the original proposal: development of a software system with a clean architecture that includes a wide range of reusable components capable of supporting further ITS research and development.

The Domain and Curricular models developed in the course of the Phase II project are substantially more detailed than what was produced during the Phase I effort. They are also more detailed and more applicable to the ComMentor problem than any pre-existing sources we could find. For example, DARPA's HPKB ontology did not include sufficient description of terrain and tactical principles while including large numbers of irrelevant vehicles and weapons. Likewise MITRE's CCSIL similarly lacked depth in describing tactics and tactically significant situational factors while including unnecessary detail about techniques and procedures. Both domain and curriculum have been subject to review and have been modified in accordance with input from our SMEs. Both have now proven themselves through their support of scenarios authored to date. We expect that these models will serve as useful starting points for future efforts at tactical training, including one project that has already started in Stottler Henke's Boston office.

During the Phase II project we explored textual language input techniques and ultimately determined that we were not yet far enough advanced in our analysis of ComMentor's language requirements to support their effective use (see "Conclusion" subsection "Student Input: The Problem of Natural Language" immediately below). Our explorations likewise determined that speech-based input was not feasible without a better advanced characterization of situation-specific student utterances. As suggested in the proposal, we turned to structured forms-based input mechanisms as an alternative to straight language-based approaches, and as a way to concretize much of the knowledge we would need to succeed at language processing. We explored and, through a series of iterations, successfully built a template-driven forms-based input facility. In the context of that new input framework, we successfully integrated some

forms of gesture-based input (e.g., pointing, sub-selecting, multiple-pointing, dragging), exploiting the existence of the situation map, force-structure diagrams, and more recently have made a start on building and using a timeline representation as well. On the output side, we have established the ability to present sequences of presentations, allowing mixtures of text, images, html, sound, and map layer manipulations.

Table 2

Phase II Project Objectives and Accomplishments

Objective	Results
1. Continue to identify and encode tactical analysis knowledge.	Phase II Domain and Curricular models go well beyond Phase I results, and are also more detailed and more applicable to the ComMentor problem than any pre-existing sources we could find. These models will serve as useful starting points for future efforts at tactical training, including projects already started in Stottler Henke's Boston office.
2. Continue to explore and evaluate candidate interaction techniques.	General text language input proved beyond our resources. We built a forms-based input facility to reduce language processing issues. We integrated gesture-based input (e.g., pointing, sub-selecting, multiple-pointing, dragging), from the situation map, force-structure trees, and timelines. On the output side, ComMentor can present sequences mixing text, images, html, sound, and map layer manipulations.
3. Continue to identify, classify, and implement potentially automatable tutoring strategies.	Substantial effort was devoted to observing and analyzing live tutoring sessions, resulting in a separate report and significant data. ComMentor uses many of the applicable strategies, some built into the structure of the system, others coded in choices made while preparing scenarios.
4. Refine our methods for authoring ITS scenarios.	The existing ComMentor Authoring Tool Suite is a flexible, usable (and reusable) base, that clearly points to a variety of incremental extensions and improvements, as well as some more significant reformulations.
5. Develop a fully functional software prototype.	The final prototype illustrates key ideas related to tactical analysis knowledge representation, student-mentor interaction methods, and interactive tutoring strategies. ComMentor can sustain an extended tutoring dialog with a student, and its dialogs exhibit local coherence while focusing on issues and themes that experts consider important.
6. Demonstrate the effectiveness of the tutoring system prototype.	It remains an open issue whether the current system or some extension, will serve as an effective learning aid for Army officers. We hope to gather preliminary data bearing on usability and student response during the contract extension period originally granted for review of this report.

Table Continues

7. Develop guidelines and baseline data on scenario authoring.	We have extensively documented the authoring process and tools in this report. Despite complicating factors, we are comfortable with our current estimate of $50,000 to develop a new scenario from scratch, including initial trial student use and consequent tuning.
8. Develop a preliminary analysis of the issues likely to bear on long-term system acceptance, effectiveness, and maintainability.	Authoring costs will have to be traded off against measured or perceived benefits, factoring in user community size, scenario longevity and maintenance costs. A major question is whether a scenarios' primary value lie in application of relatively timeless tactical principles or in practice applying more temporally bound doctrinal and factual skills? Delivery format will also become an important deployment issue.
9. Develop software with reusable components that can help support further ITS research and development.	ComMentor is built on top of a series of infrastructure layers, where each layer is itself a configurable assemblage of tools. The GRIST, MILL, and GRAIN are all useful and reusable packages in their own right. GRIST and MILL have already been used in two other projects, and along with GRAIN are currently being designed into a third project. We expect these packages will see wider application going forward.

The identification and classification of tutoring strategies that might be implemented in ComMentor occupied a substantial portion of the project effort and produce substantial results (reported in the "Results" subsections "TDG Data and Preliminary Analysis" and "Instructional Model"). We collected a large amount of data, analyzed the resulting transcripts, organized the results with respect to different frameworks and theories, and classified identified instructional strategies with respect to their applicability to ComMentor. The Phase II prototype system makes use of many of the applicable strategies, some through the structure of the system, and others through the choices made by authors in preparing particular scenarios.

The refinement of our methods for authoring ITS scenarios also received substantial attention, and our results (see "Results" subsection "Final Prototype Authoring Capabilities"), while certainly not the final word on authoring of tutoring for tactical scenarios, is a solid and usable base on which to build. The existing ComMentor Authoring Tool Suite clearly points to a variety of incremental extensions and improvements, as well as suggesting some more significant reformulations (see "Future Research and Refinement" subsection "Authoring Tool Refinements"). Furthermore, by virtue of its reliance on the GRIST representation and inference layer, and the MILL GUI framework, the entire suite has been designed and built in such a way that many of its pieces are reusable, and it is relatively easy to both to revise aspects of existing tools and to insert new tools into the suite that interoperate smoothly with existing tools.

The development of a fully functional software prototype that illustrates the key ideas related to tactical analysis, knowledge representation, student-mentor interaction methods, and interactive tutoring strategies, absorbed the bulk of the effort throughout the project. We believe we have done a solid job, given the challenging nature of the training task and format (see "Results" subsection "Final Prototype Runtime Capabilities"). ComMentor is now able to sustain an extended tutoring dialog with a student, and its dialogs exhibit local coherence while focusing on issues and themes that experts consider important for students to dwell on. Tutor behavior with respect to larger scale session coherence and more focused student feedback have

also developed somewhat since feedback from our last test user, but have not yet been subjected to trial use.

It remains an open issue whether the current system or some extension, will serve as an effective learning aid for Army officers (see discussion of the objective immediately below). We would very much like to continue working with ARI to see ComMentor through to the next steps of evaluation, refinement, extension, and deployment. We hope to gather some preliminary data bearing on usability and student response during the contract extension period originally granted to facilitate review of this report. If so, we will work to incorporate such late-arriving results into the final draft to the extent possible according to ARI review requirements.

With regard to authoring, we have now accumulated significant experience (see "Results" subsections "Scenarios" and "Final Prototype Authoring Capabilities"). One complicating factor here is that our continual refinement of the Authoring Tools Suite (enabled to a significant extent by the underlying GRIST and MILL infrastructure) provides something of a moving target with regard to the details of authoring process steps and the time they take. Nonetheless, we are comfortable with our current estimate of $50,000 to develop a new scenario from scratch, up through initial trial student use and consequent tuning. We have fully documented the authoring process and tools in this report.

Clearly the authoring issues will affect ComMentor's deployment prospects, but whether $50,000 is a lot or a little depends on how many users over how many years may be able to use the scenario, and what benefit they may derive from it. How long a scenario may be used in its more-or-less original form before it has to be revised to keep current with a changing world will depend in large part on what the perceived benefit of the scenario is: does its primary value lie in application of relatively timeless tactical principles or in practice applying more temporally bound doctrinal and factual skills? There is also the question of whether Scenario authoring costs will always remain at the estimated level, or may decrease as improvements to authoring tools and methods are deployed. In any case, we have made a good start on identifying the relevant factors, gathering preliminary data, and framing questions to guide future planning.

As suggested above, we believe we have satisfied one additional objective: development of a software system with a clean architecture that includes a wide range of reusable components capable of supporting further ITS research and development. ComMentor is built on top of a series of infrastructure layers, where each layer is itself a configurable assemblage of tools (see "Results" subsection "Final Prototype Architecture and Design"). GRIST, MILL, and GRAIN are all useful and reusable packages in their own right. GRIST and MILL have already found application in two other projects, and they along with GRAIN are currently being designed into a third project. We expect these packages will see substantially wider application going forward.

Student Input: The Problem of Natural Language

Natural Language Processing (NLP) is one of the well-known "hard problems" of AI. It proved a significant issue in the ComMentor project, with iterative efforts to arrive at a workable input mechanism delaying demonstration of the Socratic tutoring algorithms. In this section we aim to discuss the prospects for longer-term NLP enhancements to ComMentor. Natural language output is adequately handled by a relatively straightforward system that relies primarily

on canned tutor utterances, and secondarily on substituting strings associated with represented objects in place of variables embedded in the basic tutor utterances. Our discussion here focuses on language input as the more significant problem at this stage in ComMentor's development. Our ultimate conclusion is that natural language input remains a reasonable goal for a more fully developed ComMentor, but that there will (for the foreseeable future) remain limitations on what the system can understand, and the capability will require both significant up-front development costs as well as ongoing maintenance costs.

All of our discussion is framed by the assumption that NLP is a process of mapping natural language to and from structured internal representations of a kind typical in symbolic AI. Such representations can generally be viewed as notational variants on formal logics, and have the primary benefit that they can be used to simplify authoring and comprehension of pattern matching and inference. This assumption is in contrast to other possible approaches to NLP that depend more heavily on statistical properties of the surface form, or that use non-inspectable internal representations such as neural networks.

ComMentor initially aimed to comprehend successive user utterances in an interactive multi-turn text dialog. During the course of the project we scaled back our ambitions to comprehension of text phrases (primarily noun phrases), and managing the higher levels of input through template-driven forms-based input. A more robust NLP solution would return to address the issues of comprehending complex text sentences and connected sets of sentences (discourse), supporting interactive multi-turn text dialogs, and eventually extending to speech input and spoken dialogs.

While NLP is a hard problem in general, there are workable solutions in limited circumstances. Beyond having command of a reasonable suite of NLP tools (e.g., morphology analyzers, syntactic parsers and/or pattern matchers, etc.), there are two critical preconditions to success: (1) working in a well-understood limited domain, and (2) knowing at any moment the sorts of things that might reasonably be said. The difficulties that arose in ComMentor are, we believe, primarily attributable to violations of these preconditions. The question is whether those violations are necessary permanent features of the problem, or whether they are transitional—having more to do with the evolving state of our understanding of what representation and input would be necessary in ComMentor.

With regard to the first precondition—a well-understood limited domain—our assessment is mixed. We have to consider three sub-issues: (a) how well does the development team understand the domain, (b) how well does anyone understand the domain (how much agreement is there in the larger community on the critical concepts and terminology), and (c) how large or small is the domain, or how well does the problem allow for placing of limits on what must be considered. We consider each of these in turn:

(a) Developer understanding – Here the news is good. How well any development team understands a domain improves with time, and our team certainly knows much more about Army tactics, doctrine, and relevant operating conditions now than we did two years ago, or even one year ago.

(b) General agreement about a domain analysis – Here the news is mixed. At some levels, there exist good consensus analyses of important aspects of the military domain (as embodied in doctrinal terminology). Unfortunately there also seem to be many areas where either there is no solid analysis, or it is hard to ascertain how widely a putative consensus is actually shared. We observed differences between Marine and Army ways of talking and thinking, and vast differences in the levels of sophistication among Army captains (admittedly drawn from different specialties and having vastly different training and backgrounds). Ultimately, our primary need is for sufficient consistency among our target users (or perhaps their instructors). It would help to have a clearer notion of the precise target group (e.g., officers entering the Armor Captain's Career Course), and more access to representatives of that group.

(c) Raw domain size and ability to enforce limits – On this point the news is most challenging. The military domain is immense, and tactical exercises can potentially require consideration of a huge range of possible situational features and lines of causal reasoning. Of course this is exactly what you would expect for any training problem addressing professional-level decision-making.

The potentially open-ended nature of professional decision-making raises the issue of the second major precondition: can the system have a reasonable set of expectations about what might be said at any given moment? Just as the core ITS assessment algorithms must back away from exclusive reliance on general domain knowledge in favor of scenario-specific knowledge, so too a language-analysis component will probably have to find a way to incorporate scenario-specific—and even more narrowly context-specific—expectations. This suggests that just as authoring tools for scenario-specific tutoring knowledge are critical to enabling a complex-skills ITS to be built, so too, must an ITS have authoring tools for scenario-specific language processing expectations if it is to have a natural language front end. Just as we expect eventual convergence (or slowing in growth) of the domain ontology, inference rules, templates, and dialog structures, yet expect to continue to rely on authors to deploy these resources appropriately for each new scenario, so too we should expect a similar convergence in vocabulary, and (semantic) grammar constructs, while continuing to rely on authors to vet attachment of expectations about these items to scenario contexts.

Another way of looking at the problem suggests we have made substantial progress in the course of this Phase II project. We started out looking at transcripts of live tutoring sessions and trying to induce semantic and linguistic patterns that might cover essentially anything that was said. Our initial aspiration was to reproduce as much of the observed tutor behavior as possible. Now we have an actual system that reproduces what we believe is a useful subset of that behavior. To make that system work, we had to define a repertoire of input templates so that students could say the kinds of things to which the system was prepared to respond. One way to proceed is to treat that evolving set of templates as specification of the semantic space that now actually would need to be covered by a language analyzer—a space that is certainly much smaller than everything we saw in the transcripts.

A successful natural language input system for an ITS like ComMentor, then, does not seem an impossibility, but likely depends on adopting the strategies implied above and accepting (1) some additional ongoing authoring and maintenance costs, (2) some limitations on what inputs will be successfully comprehended, and (3) some impact on runtime performance.

Prospects for Battlefield Command Reasoning ITSs

We believe that the ComMentor project has made substantial progress towards the goals it was set up to address and that the prospects for battlefield command reasoning ITSs remain promising. A relatively uncontroversial set of observations about learning and training combine to suggest that if you could increase both the supply and consumption of coached simulated tactical experience—especially intensive coached drill on key component skills—students would become better tacticians.

- Time on task – Aptitude and motivation play tremendous roles in who becomes competent or excellent at any particular skill, be it chess, baseball, piano playing, medical diagnosis, or military command. Still, it is a broad conclusion of education and training research that time on task is perhaps the number one predictor of learning. This observation has to be hemmed around with all sorts of qualifiers about the quality of the time spent and how the learning activity stacks up against the actual target task. But it seems safe to say that a student is likely to learn more if he or she works diligently on TDGs for two hours than if they work diligently on TDGs for one hour. ComMentor can make it easier for more students to spend more time working in a focused way on TDGs.

- Simulation vs. simulation with expert tutoring – Again, all things being equal, there is good reason to believe that students will learn more from working through exercises (simulated problems) if there is someone or something there to point out what they are doing right and wrong than if they have to rely solely on their own assessment of the naturalistic consequences of their actions in the simulation. Army officers get only very limited opportunities to work simulated (or for that matter real) exercises with expert guidance, feedback, and coaching (e.g., from senior officers). ComMentor is able to give some forms of expert guidance and can be made available to any student at any time.

- Study or exercises vs. intensive drill – The training conception underlying the entire Think Like a Commander research thrust (Lussier, Ross, & Mayes, 2000) suggests that expert reasoning skills can be enhanced not merely by having more experience, but by having particular kinds of focused experiences characterized by intensive drill on fundamental skills—a kind of effortful start-stop-comment-restart work, rather than fluid uninterrupted exercise of already practiced skills. ComMentor emphasizes analysis and discussions that focus on key issues and force thinking about (and rethinking) scenarios over continuous simulated play.

Our judgment is that the ComMentor project suggests that for a price measured in hundreds of thousands of dollars, you can broadly increase the supply of at least a basic level of intensive coached drill on key component skills of battlefield command reasoning. Over time we expect the level and the quality of that training experience can be improved. The results to date are probably not as encouraging with regard to whether you could increase the consumption of such training. We may not yet have achieved a level of usability that lets the potential of the system be reached.

As noted in the "Future Research and Refinement" section, we have no data quantifying the benefit that would accrue from time spent with ComMentor. Of course there is little data

quantifying the benefits that accrues from other forms of high-level decision-skills training. One hypothesis is that student ratings of such learning experiences are a good proxy for whether something was actually learned—after all these are professionals with some level of competence and a significant reflective capability to judge when a situation has surprised them, when they have gotten a good tip, or when they have improved on their prior understanding and performance. These are also busy professionals, which means that there is probably a good correlation between whether or not they choose to use a resource that is made available to them and whether they (and their community) assess the resource as having value. If ComMentor regularly produced blinding insights in its users, then they would likely put up with a good deal more difficulty in its user interface and use it anyway. The causality potentially cuts the other way however: a limited user interface may make it impossible to capture enough input to be able to provide blinding insights, and it certainly raises the bar for the level of perceived benefit required to entice users into putting up with potential software frustration.

As all this suggests, we see ultimate success here as a balancing act. No ITS will ever match the motivation and excitement we saw when captains were given the undivided mentoring attention of a major and a general for two hours at a stretch. But we do believe an ITS can provide a sufficient approximation to the content of such interactions to produce learning that is perceived as useful. Further we believe the impediments introduced by limitations on user interface technology can be made low enough to let that content value win out. Finally, we believe a version of this vision can be achieved for a cost that is not prohibitive, amortizing the cost of the technology across a wide range of professional-level decision-skills applications within (and beyond) the military, and amortizing the cost of scenario generation across all students needing training in a particular set of professional-level skills. Rather the costs will ultimately be seen as a bargain compared to other available alternatives, or to the lack of practical alternatives.

References

Brown, J. S. & Burton, R. R. (1975). Multiple representations of knowledge for tutorial reasoning. In D. Bobrow & A. Collins (Eds.), *Representation and Understanding*, New York: Academic Press.

Carbonell, J. R. (1970). *Mixed-Initiative Man-Computer Instructional Dialogues*. Unpublished Doctoral Dissertation, Massachusetts Institute of Technology.

Chandrasekaran, B. (1987). Towards a functional architecture for intelligence based on generic information processing tasks. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, 1183-1192.

Clancey, W. J. (1987). *Knowledge-based Tutoring: The GUIDON Program*. Cambridge, MA: MIT Press.

Collins, A. (1976). Processes in acquiring knowledge. In R. C. Anderson, R. J. Spiro, & W. E. Montague (Eds.), *Schooling and the acquisition of knowledge*. Hillsdale, N.J.: Lawrence Erlbaum, 339-363.

Crandall, B. W., Kyne, M., Militello, L., & Klein, G. A. (1992). Describing expertise in one-on-one instruction (Contract MDA903-91-C-0058 for U. S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, VA). Fairborn, OH: Klein Associates Inc.

Edelson, D. C. (1992). When should a cheetah remind you of a bat? Reminding in case-based teaching. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, 667-672.

Forgy, C. L. (1982). Rete: A Fast Algorithm for the Many-Pattern/Many-Object Pattern Match Problem. *Artificial Intelligence*, 19(1) 17-37.

Glass, M. (1999). *Broadening Input Understanding in a Language-Based Intelligent Tutoring System*. Doctoral Dissertation, Illinois Institute of Technology.

Gragg, C. I. (1940). Because wisdom can't be told. *Harvard Alumni Bulletin*, October 19, 78-84.

Guralnik, D. (1996). *An Authoring Tool for Procedural-Task Training*, Doctoral Dissertation, Northwestern University, Institute for the Learning Sciences Technical Report 71.

Lussier, J. W., Ross, K. G., & Mayes, R. (2000). Coaching techniques for adaptive thinking. In *Proceedings of the 2000 Interservice/Industry Training, Simulation, and Education Conference, Orlando, FL.*

Munro, A. & Pizzini, Q. A. (1995). *RIDES Reference Manual*, Los Angeles: Behavioral Technology Laboratories, University of Southern California.

Rose, C. P., Moore, J. D., VanLehn, K., & Allbritton, D. (2000). *A Comparative Evaluation of Socratic versus Didactic Tutoring*, University of Pittsburgh, LRDC Technical Report LRDC-BEE-1.

Ross, K. G., Battaglia, D., Hutton, R. J. B., & Crandall, B., (2003). Development of an Instructional Model for Tutoring Tactical Thinking. Klein Associates Inc., Report on subcontract to Stottler Henke Associates (Subcontract No. SHAI-COMM-01; Prime Contract No. DASW01-01-C-0039).

Schank, R., Fano, A., Bell, B. & Jona, M. (1994). The Design of Goal-Based Scenarios. *Journal of the Learning Sciences,* 3(4).

Schmitt, J. F. (1994). *Mastering Tactics*. Quantico, VA: Marine Corps Association.

Schmitt, J. F. (1996). Designing good TDGs. *Marine Corps Gazette*.

Steels, L. (1990). Components of expertise. *AI Magazine*, 11(2), 28-49.

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).

USMA Redbook (2001). *Academic Program (Redbook), AY 2001-2002*. Available on-line at http://www.dean.usma.edu/AAD/Redbook/

Whitehead, A. N. (1929). *The Aims of Education*. New York: Macmillan.

Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser, A. (1999). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In *Proceedings of The Ninth World Conference on Artificial Intelligence in Education* (AI-ED-99), Amsterdam: IOS Press, 535-542.

Zsambok, C. E., Crandall, B., & Militello, L. (1994). OJT: Models, programs, and related issues (Final Technical Report Contract MDA903-93-C-0092 for the U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, VA). Fairborn, OH : Klein Associates Inc.

Zsambok, C. E., Kaempf, G. L., Crandall, B., & Kyne, M. (1996). OJT: A cognitive model of prototype training program for OJT providers (Contract No. MDA903-93-C-0092 for the U.S. Army Research Institute for the Behavioral & Social Sciences, Alexandria, VA). Fairborn, OH: Klein Associates Inc.

Appendix A

Project Chronology

This appendix presents a condensed summary of the major activities and accomplishments during each month of the ComMentor Phase II project.

Month 1 (October, 2001). During the first month of the project we began the process of planning and organizing project efforts, and devoted substantial resources to reviewing relevant literature. Due to limited availability of our subject-matter experts (SMEs) during the first three months of the project we immediately had to revise our proposed schedule, especially for tasks touching on tutoring observations. We did, however, make headway on structuring the set of sample scenarios to be used during the observations so as to cover a range of situations of interest to armor officers in the U.S. Army, tentatively targeting our three waves of observations to focus on (1) current armor operations, (2) the Interim Brigade concept, and (3) a notional version of a Future Combat Systems (FCS) unit. Given that the Phase II efforts were planned as a continuation of the Phase I work, there was plenty of knowledge engineering, design, and implementation work to get started on while waiting to get new tutoring observations underway.

Month 2 (November, 2001). During the second month of the project we identified our first two sample training scenarios—"Enemy Over the Bridge," and "Clash at Timpan-ni"—and planned the initial wave of tutorial observations. We also began the Phase II design effort including identification and assessment of possible supporting tools, in particular the Protégé knowledge representation and acquisition system from Stanford and the Java Expert System Shell (JESS) rule engine from Sandia Labs. We started early implementation work on the simplest module—the graphical I/O module intended to support map interaction—as it had the fewest dependencies on other decisions that are yet pending. At the end of the month, we held our Phase II project kickoff meeting at ARI Fort Knox.

Month 3 (December, 2001). During the third month of the project we finalized detailed plans and schedules for the first wave of tutorial observations and for a face-to-face full team meeting (all to actually occur in January 2002). With the observation sessions now scheduled, the early schedule slippage was under control. We also devoted substantial resources to design and initial implementation of significant pieces of the Phase II runtime and authoring system. The most exciting work during the month focused on elaborating the tutorial goal structure and accompanying processing cycles presented at the November kickoff meeting. We developed a hierarchical structure for ComMentor scenarios (what eventually became the final structure of Scenarios, Scenes, Evaluations, and Dialogs), and began working out the implications of that structure for possible processing algorithms. Based on the initial cut at Scenario structure we launched preliminary work on authoring tools (that included screens reflecting aspects of what eventually became our final tool suite, including early tools for Tutoring Authoring, Situation Authoring, and Presentation Authoring).

Month 4 (January, 2002). The fourth month of the project was a busy one. Major events included (1) further early development of an updated runtime and new authoring tools, (2) a full team meeting in the Washington DC area, primarily devoted to reviewing the current system design and prototype, and (3) carrying out the first wave of tutorial observations covering two

new scenarios. Discussions at the team meeting led us to slow work on the authoring suite as we tried to incorporate comments from our SMEs into the design. Comments included (a) suggestions for additional features of and constraints on the GUI, (b) recommendations that we focus on a standard format such as the fragmentary order (FRAGO) for capturing student orders, (c) emphasis of the importance of introducing some representation of time (e.g., a timeline), (d) a suggestion to modify our concept of the system's diagnostic machinery from large-grained "cases" to smaller-grained "evaluations," and (e) a loosening of constraints on what eventually became Scenes. The first wave of tutorial observations went smoothly, and produced initial data that was encouraging with regard to both identifiable regularities and interesting variation in student and tutor behavior.

Month 5 (February, 2002). During the fifth month of the project several tasks advanced including (1) initial analysis of general evaluations and possible curriculum organizations, (2) related advances in curriculum representation coupled to implementation work on curriculum authoring, (3) a first foray into building test suites aimed at improving long-term system robustness (in this case, applied to the early language processing module), and (4) major analysis by our Klein subcontractors of the tutorial observation data generated in the previous month, including initial extensions to their tutorial strategies inventory. Other minor events included refinement of plans for the access to active duty Army officers in June, and submission of an Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) abstract reporting on the project progress.

Month 6 (March, 2002). During the sixth month of the project we worked on three major classes of tasks: analysis, design, and implementation. Analysis tasks included (1) refinement of general evaluations and possible curriculum organizations, and (2) start of second-round analysis of the tutorial observation data generated at the end of January (this time by Stottler Henke staff). Design tasks included (3) a reconceptualization of the scene mechanism to enable plugging together of different kinds of scenes that support different styles of interaction, and (4) revision of the general agenda mechanism underlying tutor control. In addition to initial agenda coding, implementation tasks included (5) introducing a new version of the order elicitation scene more in keeping with expert advice, and (6) addition to the curriculum authoring module of a prerequisite language patterned on aspects of the Sharable Content Object Reference Model (SCORM) standard, and editing capability for expressions in that language.

Month 7 (April, 2002). During the seventh month of the project work advanced in six areas: (1) planning, (2) observation, (3) analysis, (4) text processing, (5) pattern language, and (6) discussion control. Planning focused on the second wave of observations which began during the month, as well as the activities (including the third wave of observations) scheduled for June at Fort Riley. The observations conducted this month included four sessions devoted to the "Gap at the Bridge" scenario, and had two interesting aspects: first the scenario was a straight role-reversal of the earlier "Enemy Over the Bridge" scenario, so we gained a much deeper understanding of the military logic underlying the situation (and importantly, so too did the one student who played both scenarios), and second our SMEs experimented with playing the scenario with more explicit temporal component by having scenario time pass in real time. The main analysis task focused on working in detail through the "Clash at Timpan-ni" scenario to see how well it fit with proposed ComMentor scenario structures. To simplify the problem, we eventually decided to switch our focus to "Enemy Over the Bridge" because it presented less

complexity in terms of required terrain analysis. Design work generated new ideas about how to exploit information captured in the agenda to help resolve ambiguities in text input, and how to sequence discussions. Related implementation work added experimental capability to resolve some anaphoric references and ellipses, and extensions to the old Phase I pattern language and matcher.

Month 8 (May, 2002). During the eighth month of the project we completed the second wave of tutorial observations with four sessions devoted to the "Enclave" scenario (an urban warfare problem), and prepared for the third wave of observations by having our SMEs develop a pair of FCS-based scenarios: "Attack on Kalat" and "Covering Force at Kalat." Development of the new scenarios was part of our preparation for the week of observations and system trials scheduled for the following month at Fort Riley, during which we would have access to a dozen active-duty Army Captains. We also worked to prepare the prototype system for its debut in front of potential users, including implementation of new functionality, massive extension of the domain model and parsing knowledge, and initial work on encoding our first sample scenario, which in turn required extensions to the authoring tools. During the course of expanding the system ontology and related parsing knowledge we first noticed performance issues with the underlying knowledge representation system that became a significant motivation for the decision made several months later to replace that key infrastructure package with what eventually became the GRIST package.

Month 9 (June, 2002). During the ninth month of the project we traveled to Fort Riley to carry out our third and final wave of tutorial observations, as well as an initial formative evaluation of the partial Phase II prototype. We also traveled to ARI Fort Knox to participate in the first In-Process Review (IPR) along with CHI Systems, Inc., at which we gave a short demonstration of the existing prototype runtime and authoring tool, as well as a peek at the underlying ontology and parser results. At Fort Riley, the eight tutorial sessions using the FCS scenarios were very interesting and productive. Rather than the usual 1-½ hours, these sessions ran 2 hours largely because it took almost an hour to get past the introduction with its extended discussion of the Objective Force unit structure and capabilities, before students got into the actual scenario. During the scenario, students had to recognize that old doctrine simply did not apply, and start to invent new doctrine on the fly. We had access to three other subjects, and these we ran through a shortened version of the "Enemy Over the Bridge" scenario, restricting their input to typed text in order to verify that the essential TDG mentoring interaction could transfer over to a textual (as opposed to verbal) format, and to gather some initial data on the character of such textual input.

With two of these final three Captains we also had time to run the ComMentor prototype through its initial order entry phase and received useful feedback on the user interface design, and details of the displays and presentations. We had a cautionary experience when the first of these students turned out to be a tactical novice, and froze when asked to enter orders in response to the initial situation description. It became clear that there is a tremendous range in the quality and preparation of Captains at large, and that there is a lower level of student competence beneath which it will not be practical to expect ComMentor to function.[18] Fortunately, the final

[18] This phenomenon was essentially repeated in April of 2003 at Fort Knox when some of our subjects in a second round of explorations showed similar lack of relevant background knowledge and experience.

student was a very strong tactician, and responded more in the way we had expected. The system was in fact able to parse his first two orders satisfactorily, and missed this third order due to a known syntactic limitation. Overall, we were encouraged that for students in the range of ability we had anticipated, it appeared that the design of the system made sense, and that its ability to interpret input was approaching acceptable.

Month 10 (July, 2002). During the 10th month of the project we concentrated on scenario authoring tools, including design reviews with our SMEs, and design refactorings that began to point us towards what became our authoring tool framework (and eventually the Modeling Inspection and Linking Library [MILL]). We also finalized our I/ITSEC-2002 submission. Our SMEs agreed that there was a surprising amount of consistency across students running the same scenarios and that their mentoring improved across runs as they refined their own understanding of a scenario's key points, got a feel for common student problems, as well as what lines of discussion tended to work better or worse. They disagreed on whether our idea of structuring ComMentor tutoring largely around argument trees was appropriate, with one feeling that would imply too closed a view of any problem.[19] Neither of them found our early presentation of the Dialog tree proposal to be a close fit to their natural way of thinking, but we expected the introduction of Curriculum nodes would mitigate that issue. Based on lessons learned from the early work on the Tutoring Authoring tool in preparation for the June trip to Fort Riley we began the process of generalizing and modularizing that component of the ComMentor authoring suite. The new design did a better job of consistently displaying the structure of a scenario, and the context for any object being edited. It also made it easier to define new kinds of scenario structures and slot them into the editor; for instance, our taxonomy of discussion types was expected to continue to evolve, and inserting new discussion editors was simplified. Finally, this revision created a more consistent connection to presentations, which were given a tool of their own, but were also made linkable and launchable from within the Tutoring authoring tool (as well as from within the Curriculum authoring tool, which was next in line for a comparable overhaul).

Month 11 (August, 2002). During the 11th month we again concentrated on authoring tool development. Work focused on refining the Tutoring and Presentation authoring tools, their interaction in the general authoring tool framework, and the introduction of a fixed question/answer network authoring tool. We also revised our accepted I/ITSEC-2002 paper.

Month 12 (September, 2002). During the 12th month our work was split between analysis of tutoring session transcripts, initial clean-up/overhaul of user interface code, and continuing work on authoring tool implementation, emphasizing the Curriculum tool. Focusing on the "Enemy Over the Bridge" scenario sessions, we reviewed the transcripts to identify discussion topics and patterns of recurrence across sessions. The goals were to pick out possible Dialogs nodes and Curriculum points, and to think about the implications of various control mechanisms. We held an extended teleconference with John Schmitt to review our analyses and compare them with his recollections and reconstructions of the sessions.

[19] Needing to impose some structure to support computer management of dialogs we ultimately ended up with a simple generalized version of argument trees for our Dialogs, and rely on the ability to author multiple Evaluations (each with its own Dialogs) to recapture aspects of the open-endedness of the problems.

Month 13 (October, 2002). At the start of the second year of the project, the 13th month saw an increase in effort and the beginning of an important new line of work. While work continued on the authoring suite, focused on interoperability across authoring modules (e.g., drag-and-drop editing and linking), extensive design and initial implementation were carried out to enhance the system's underlying knowledge representation system both to address identified performance issues and to prepare for the introduction of inference capabilities. Effectively, these parallel efforts were the engineering that ultimately led to the MILL (user interface) and GRIST (representation and inference) packages. The main purpose of the representation work was to prepare the ground for future detailed design and implementation of both forward and backward inference capabilities over the new representational formalism (effectively supporting both case-based and rule-based reasoning). Klein Associates delivered their draft Instructional Strategies report for Stottler Henke review.

Month 14 (November, 2002). During the 14th month work continued to focus on coding, including getting the authoring suite to a stable state in anticipation of the port from Protégé to GRIST, and completing the first round of implementation work on GRIST. Work on the design of inference capabilities proceeded at a low level. Klein Associates' draft Instructional Strategies report was reviewed.

Month 15 (December, 2002). During the 15th month work continued at a somewhat slower pace on implementation of ComMentor tools. Major effort focused on refining the new GRIST API in support of porting the existing authoring tools from Protégé to GRIST. Work on the backward-chainer also advanced substantially. We attended the I/ITSEC conference to present our paper on ComMentor, which also afforded opportunities to meet with ARI and Klein staff. We delivered feedback to Klein on their draft Instructional Strategies report. We also had several good discussions with ARI staff about project status and direction. Our interpretation of the major guidance offered was that we should de-emphasize natural language processing issues (and complex input interpretation in general) in favor of development and application of interesting instructional strategies. This shaped the direction of our work over the balance of the project.

Month 16 (January, 2003). During the 16th month work advanced on the porting and enhancement of authoring tool modules and general ComMentor system support modules. In particular the graphical map module was enhanced, the language parser was ported, and the ComMentor domain ontology was also ported. Work on the basic backward chainer was completed. In retrospect, we should have realized at this point that while we were making good progress on replacing earlier pieces of system infrastructure with more scalable, capable, and extensible alternatives, we were incurring delays that presented a danger of missing our April release target for a useable version of the system.

Month 17 (February, 2003). During the 17th month we focused on pulling the second version of the ComMentor system together in preparation for initial field trials in April. Progress was made on enhancing the authoring tool suite to enable completion of the first scenario's content. Design of a very capable forward chainer was completed, and a solid start on implementation was made. In response to ARI's direction to de-emphasize natural language input, work on a template-driven input mechanism that would relieve the need for the parser to interpret complex sentences was started. The Instructional Strategies Report from Klein

Associates was delivered, and two abstracts based on the project (one from Stottler Henke and one from Klein) were submitted to I/ITSEC-2003.

Month 18 (March, 2003). During the18th month substantial progress was made on ComMentor authoring tools, runtime system, and initial scenario content in preparation for the April field trials. In particular, many authoring features were added to smooth and speed the authoring process based on experience using the tools, the forward chainer and template-driven forms-based input system were largely completed, and new representations and input forms were developed based on MITRE's Command and Control Simulation Interface Language (CCSIL). However, it became clear that with one week left before field trials at Fort Knox, problems with the robustness and efficiency of the scenario authoring tools, and gaps in the runtime system were making it difficult to ensure that a usable sample scenario script could be authored and tested in time. Our long-term efforts to carefully architect the system and build suites of test cases as we went had paid substantial dividends throughout the development process, but when it came to late-arriving features and user-interface functionality, systematic and pervasive testing necessarily broke down.

Month 19 (April, 2003). During the 19th month we made final preparations for the ComMentor release that we ultimately took to Fort Knox for use with a set of active-duty Army officers. Those sessions were less productive than originally hoped, and were ultimately curtailed once we had gathered as much useful feedback on the system as we could from our first six subjects. The six officers varied widely in background and ability, which provided an interesting range of opinions; each session lasted approximately 1 hour. Specific outcomes of these interactions included: (a) small refinements to the "Enemy Over the Bridge Scenario" (e.g., revisions in terms and symbology, as well as more up-front orientation to the tactical decision game and the scenario situation); (b) major overhauls to the input facilities focusing on disaggregating the original monolithic FRAGO template into a set of smaller forms (e.g., for tasking a single unit) to be attached to the existing visualizations (e.g., map and unit icons); (c) some accompanying revisions to the underlying ontology to better capture the Army officers' conceptualization of how orders are issued; (d) increased priority on using the visualizations to support output, and on providing better orientation to the overall use of the ComMentor system.

Following the sessions with these officers, the second project IPR was held during which we emphasized a presentation of the project plan, laying out what had been accomplished, and what remained to be done to fill gaps in the existing implementation. We also proposed a specific timeline with three new milestones to complete the final six months of the project. The new milestones were as follows:

- June 1: Existing functionality debugged, incorporation of major feedback from the just complete field trials (e.g. revised scenario presentation, disaggregation of input forms), and content for the "Enemy Over the Bridge" scenario.

- August 1: Addition of active evaluation disambiguation and handling of student-initiated question digressions, and content for a second scenario.

- October 1: Addition of student-model driven scenario selection and dialog prioritization, plus content for a third scenario, and experiments with SME use of authoring tools.

By the end of the month we had accomplished all of the major software extensions required for the June 1 delivery. In addition, we had made initial progress on the content tasks to be accomplished for June 1, and were well positioned for the final remaining effort of tweaking the ontology and templates, and entering the main tutor evaluations and discussions for "Enemy Over the Bridge."

Month 20 (May, 2003). During the 20th month we developed and tested tutoring behavior for the "Enemy Over the Bridge" scenario. This work called for extension of our authoring environment with new support for controlling and inspecting system activity—especially to get insight into the processing of patterns and rules. We also extended some of our authoring sub-languages and strengthened built-in validation checks to make sure that written patterns conformed to language requirements. These efforts were in line with our general theme of building testing and debugging features along with the main system, and of trying to make the authoring tools as user-friendly as possible. An assortment of improvements were made to the user interface and to the representations that underlie both that interface and the system's reasoning. Finally, as part of preparing for our June 1 delivery, we developed initial user documentation.

Month 21 (June, 2003). During the 21st month we delivered a working version of the tutor features as discussed in April. We also refined the tutoring behavior for the "Enemy Over the Bridge" scenario, and debugged aspects of the underlying tutor. With those refinements, we demonstrated the system for both internal and external review. In preparation for the next release (August 1), we developed a detailed task breakdown. The initial question-answering capability was implemented and a start was made on the time-line representation for events in a scenario. We also began work on preparing an introductory video for new students based on the "Tanks on the Farm" scenario.

Month 22 (July, 2003). During the 22nd month we prepared for the second of three deliveries discussed in April. This involved implementing two new promised features: student initiated digression questions, and tutor initiated disambiguation questions. It also involved implementing two additional features: force-structure coupled timelines for event visualization and time input, and a video introduction to the system, as well as a host of other small improvements and extensions to both runtime and authoring tools. We also significantly extended the "Enemy Over the Bridge" scenario and began work on the "Attack on Kalat" scenario. During this month, Klein's I/ITSEC paper was accepted for publication. Based on discussions with ARI about scheduling a review period for this project Final Report, and with ARI's concurrence, we requested a 2-month no-cost-extension to the project which was granted.

Month 23 (August, 2003). During the 23rd month we delivered and then gathered feedback on the August 1 ComMentor release, began work towards the October 1 final system delivery, and started preparations for the project Final Report. Feedback came from Maj. Schmitt and touched on issues such as the lack of ability for students to give multiple answers to questions, weak tutor feedback to the student on the quality of answers, lack of clear transitions among topics, and lack of a clear ending point to the entire session interaction. Meanwhile, work

on the system covered a wide range of issues including inferences (noticing overlapping regions and locations of forces), user input (ability to designate forces vicinity any location), authoring tools (ability to link curriculum points to evaluations and dialogs), authoring (miscellaneous scenario fixes), and other assorted improvements (e.g., error logging, bug fixes, KB updates, etc.).

Month 24 (September, 2003). During the 24th and final real working month of the project we focused on five tasks: (1) experimenting with SME authoring of scenario content, (2) authoring final scenario content for the October 1 delivery, (3) implementing final system features for that delivery, (4) exploring issues related to ComMentor's deployment in operational use, and (5) writing a complete draft of this Final Report.

Month 25 & 26 (October/November, 2003). During the 25th and 26th months of this project (the two-months of the no-cost extension), we held the final project review meeting, revised this final report, and tidied up other lose ends. As of this writing, it remains an open question whether there will be some kind of formal system evaluation to round out the project, and if so, what form it will take, as well as when and where it will be conducted.

Appendix B

Detailed List of Potential System Refinements

User Interface Refinements

One of the major issues faced by this project was development of a user interface that most closely approximated the experience of students in the eTDGs we observed. While the user interface of the final Phase II prototype has advanced considerably from its earlier forms, there are a large number of features that ought to be added to enhance usability and expressiveness. Most of these are primarily a "mere matter of programming," though many would benefit from iterative design, implementation, and trial use.[20]

- Enhancements to the Map Input Capabilities – The student ought to be able to make more kinds of annotations on the map, and the tutor ought to be able interpret those inputs (at least in the context of structured forms, if not standing on their own). In addition to dropping checkpoints, the student ought to be able to draw lines and arrows, and to designate regions. The student should also be given control of attributes such as line color and thickness, and potentially be able to assign semantic labels to their annotations (e.g., labeling an arrow as an axis of advance).

- Enhancements to the Map Output Capabilities – The student is not the only one who should be able to annotate the map in the ways described above. The tutor (or the author) should have more freedom to place annotations on map layer that can be used in Presentations that accompany tutor utterances. A capability to animate icons (including a pointer) would also be very useful. Finally, expansion of the system's capability to composite appropriate icons automatically would simplify authoring.

- Enhancements to the Timeline Input Capabilities – Like the map, the interactive timelines that now accompany the force-structure wire-diagram trees could usefully be enhanced with a wide variety of additional input features. Examples include the ability to reassign tasks to different units (i.e., by dragging a task from one line to another), to change the start and end times of events (i.e., by dragging the start or end of event bars), to reorder chains of events (i.e., by dragging entire event bars to new positions relative to old inter-constrained events), etc.

- Enhancements to the Timeline Output Capabilities – The timelines could also support additional output features such as showing Scenario landmark events (i.e., events described in the introductory briefing), providing a moving sense of "now," offering some visualization of ordering constraints on events, and temporally under-constrained events in general.

[20] Note that some of these features, such as use control of line colors or simple icon animations, actually appeared, in limited forms, in earlier iterations of system implementation, but were lost along the way as a focus on usability, robustness, and integration with the tutor caused us to pare back the features we could effectively support.

- Enhancements to the Forms Component – The template-driven form-based input facility could be improved in a number of ways including tuning the relationship between standard menus and discourse buttons on the one hand, and specific (primed) forms on the other, introducing new widgets (such as a combo-picktree), and providing better facilities for specifying output paraphrases of the data entered using forms (compared to the current playback of the form's layout).

- Enhancements to the Forms Repertoire – The system's repertoire of forms needs to be expanded. In part this will happen as a natural evolution driven by authoring of additional Scenarios. But there should also be a focused effort to develop certain generally useful classes of forms, such as ones for entering reports and requests.

- Enhancements to Natural Language Input – The issue of natural language input can be revisited at a number of levels ranging from better integration of free-text input into the existing forms (e.g., spell-checking and correction, or better feedback on partial interpretations), or better coverage of typical noun-phrases typed into existing form blanks, to better coverage of a higher-level utterance structures that can start to replace some of the current form-based structure.

Runtime Refinements

In addition to the user interface issues addressed above, the ComMentor runtime could be enhanced in a number of ways, some bearing on more flexible or appropriate exploitation of existing major classes of knowledge structures (e.g., Scenarios, Scenes, Evaluations, Dialogs, Questions, etc.), some bearing on invention and application of new structures and processes, and some focusing on more powerful application of general knowledge.

- Revisit the typology, sequencing, and even concept of Scenes – In the current implementation three distinct Scene types are available: (1) the order-elicitation Scene (with special logic for collecting a set of orders), (2) the main discussion Scene, and (3) the final reflective-session Scene (offering a special questionnaire screen for collecting blocks of input text from students on a fixed set of questions). The SME critiques suggest the order-elicitation Scene should probably be handled within the discussion Scene, which we believe will be possible once the input system's capability to accept multiple student forms on one turn is slightly generalized. A proposed terrain-analysis Scene might then also be folded in as part of the normal discussion Scene. The idea of the Scene as the unit at which alternate interpreters (likely with alternate interfaces) are swapped in may then have only vestigial relevance to ComMentor (though we suspect it remains relevant for other ITSs). Likewise, the idea that there should be some more elaborate transition logic among Scenes is probably moot for ComMentor (though still relevant for other ITSs).

- Enhancements to Evaluation node processing – Based on accumulating experience with Evaluation authoring we expect to find patterns that can serve to guide the invention of new structures, or perhaps of new authoring supports. To the extent that new structures result, then modifications to the runtime system will be required.

- Enhancements to Dialog node processing – Likewise, based on accumulating experience with Dialog authoring we expect patterns will emerge suggesting new Dialog capabilities. One example cited in the main text (see the end of the "Results" sub-section on "Scenario Model") was addition of a tutor utterance to be used when a Dialog node is considered for processing but its satisfaction pattern is already matched. Such a modification would require a change to the Dialog node structure and to the runtime (as well as to the authoring tools).

- Embedded help system – The student-initiated question facility in ComMentor is, in principle, able to handle most help-system requirements (save for students that need help figuring out how to ask questions). Before it can meet this need, the repertoire of system-related help questions and answers must be fleshed out.

Authoring Tool Refinements

There is a spectrum of possible refinements worth considering for the authoring tools ranging from addition of helpful features more-or-less within the current framework, to introduction of significant new authoring modes, to changes in what needs to be authored based on modification of the underlying runtime system. Here we present as a more-or-less graded list (following that spectrum) an assortment of potentially useful extensions we have considered, but that were deemed beyond the scope of what could be accomplished during this project.

- Retain memory of past configurations (windows, panes, pickers) – The most basic set of enhancements to the existing authoring suite would be to have it maintain a better memory of where you had been, both within and between sessions. This would allow the system, on restart to reload the KBs an author had been working on, and to reestablish the configuration of window sizes, positions, and contents that captured the prior work state. Within a session, the system ought to do a better job of remembering the browsing state of the many tree structured displays available throughout the system, so that, for instance, moving back and forth between Evaluations didn't cause you to lose your place in the Dialog trees belonging to those Evaluations.

- Allow for multiple simultaneous open windows of any single type – At a similar mechanical level, it would be useful if the system could support multiple windows of any given type at the same time (with cut/paste drag/drop, etc., between them). For instance, being able to look at the contents of one Scenario in one window while editing another Scenario in another window would likely prove a tremendous boon to authors who are learning how Scenarios go together, or who want to copy some features from one to another.

- Better default filler-picking mechanism – The default widget for picking objects to serve as slot-fillers in other objects does not present as organized a view of the available objects as would be desired in most circumstances. Often, a well-organized view of the objects exists in some other window; with drag-and-drop that view can in principle be used to find the desired filler. However, it would be better to allow embedding of such views in the pop-up picker so that users don't have to go to another window to get a well-

organized view of the objects they're searching among. Also, consistently embedding a "search" tool in all such views would provide additional help.

- In-place choosing of types and creation of instances for embedded objects – There are many screen layouts in the authoring tool where the main object being edited contains sub-objects and it is convenient to allow in-place editing of those sub-objects. It turns out there is often room for variation in exactly what kind of sub-object should go there, whether anything is needed at all, and whether the object should be a new one or a reuse of an existing object. For instance a Dialog node can contain a number of (optional) Presentations. The main Dialog node view provides embedded support for editing those Presentations, but it can only support the default Presentation types (e.g., plain text) and an author might want something fancier (e.g., a sequenced combination of text and map manipulations).

- Embedded help system – There is no help system for the ComMentor authoring suite, let alone a context-sensitive embedded help system. This is a simple usability issue that would need to be addressed in order for the tools to be more widely distributed.

- Additional, more helpful, and more flexible wizards – The authoring suite currently has two wizards, one to help initialize a new Scenario, and one to help in the creation of Scenario-specific student-askable (digression) Questions. Both of these wizards are currently useful, but they could use additional tuning to be more easily comprehensible and more helpful. We have envisioned other possible wizards to support other tasks that are more complex in general than they need to be in the most common cases. There are also other approaches to such complexity management, potentially including user experience modes, semi-hidden menu choices (as in MS Office), and simple screen layout choices that put some fields in the base layout and others in optional nested (possibly pop-up) sub-screens.

- Pattern validation *vs.* constrained structure editing – There are several places in ComMentor where the complexity of possible data structures is such that we have essentially invented small formal sub-languages to characterize the data. Examples include the Curriculum node prerequisite language, the Template form specification language, the concept string regular expression language, and most notably the concept Pattern language. Some of these languages (e.g., prerequisites and forms) are handled using constrained structure editing—complex custom user interfaces that ensure the structures built adhere to all constraints of the underlying language. Others (e.g., regular expressions and Patterns) are handled using a free-text type-in box and a parser/validator. For these later languages, it is possible to enter incorrect specifications and it is possible for edits made elsewhere (e.g., in the ontology) to cause formerly correct specifications to become incorrect. In general, we want to migrate more of our specifications towards full constrained structure editing so that it is harder to make mistakes and harder for the data to get out of synch.

- Pattern authoring by editing from normal input – By far the most complex language in the system is the Pattern language. A separate but related issue to the one just discussed

is to move towards a user interface where, whenever possible, patterns are not specified directly, but rather are created (at least drafted) through the more user-friendly services of available end-user input processing modules. That is, the author should be able to create the first draft of a Pattern intended to match expected student input by using the same user-input mechanisms the student would use: the template-driven forms-based interface, combined with the phrase parser, and visualization reference interpreters. This initial draft is still likely to require editing, and we would still like that editing to be structured and constrained, but it is almost always easier to edit something into the format you want than to start from scratch.

- Graph Visualization of Dialog Structure – During our authoring study, our SME suggested that he would appreciate a more graphical representation of the possible flow of student/tutor interaction—how the various setup/questions/summaries of a given Dialog node would relate to the presentation of other related Dialog nodes (those higher, lower, and at the same level in the tree). This kind of visualization might or might not also serve as an editing infrastructure for Dialog nodes, their Presentations and relationships to other Dialog nodes.

- Place/Unit property/relationship matrices – Moving up to still more customized knowledge acquisition techniques, we have considered developing a set of matrix views that would allow the author to quickly, and in one place, summarize what is interesting and important about a Scenario's places and units, and their interrelationships. For instance, the availability of lines-of-fire and lines-of-sight (or inversely, cover and concealment) from one region to another is often crucial. Capturing and organizing such facts for a Scenario in one place could potentially save effort and increase system coverage as compared to the current mechanism that embeds such facts in the context of particular Dialog nodes. Of course we quickly get into contextual factors that potentially make this approach break down for many applications. Whether one region allows fire on another region depends in large part on the fire systems available. Whether a region has any particular import to a particular unit depends in large part on what stage we are at in the time evolution of the scenario—what the unit's current intentions are, where it is, and where its adversaries may be.

- Reusable (specializable) Evaluation/Dialog trees – Based on our still quite limited experience, it is safe to assume that there are patterns of Evaluations and Dialogs that will repeat across Scenarios, and perhaps even within Scenarios. A good example is the "RedSituation" Dialog that occurs early in pretty much every ComMentor Scenario. The details always vary, but it would be helpful to have a general skeleton for such a Dialog that could then be edited to fit the demands of the particular Scenario. The current system allows for copy-and-paste of Dialog trees, but a better approach would include support for adapting what was copied to its new setting—hence the notion of storing a generalized reusable Dialog tree and providing specialization support. At a more abstract level (and focusing on Evaluations rather than Dialogs) we expect one common and durable pattern will be the Evaluation that discusses a plan flaw, which culminates in an opportunity for the student to revise their plan, leading to a set of nested Evaluations intended to respond to different typical revisions.

- Evaluation/Dialog-specific knowledge elicitation dialogs – Following up on the previous two points, we can envision versions of reusable Evaluations and Dialogs that also represent key aspects of the specialization process and how to find or elicit knowledge that is typically relevant to that specialization (e.g., the kind of knowledge that could be captured in the matrices described above). The result might be viewed as a family of smart configurable wizards associated with the generalized nodes.

Representation Refinements

- Author additional Scenarios – The main drivers for refinement to the ComMentor system will be (1) new demands (or emerging patterns) encountered in the process of encoding new Scenarios, and (2) user feedback which will be available in greater variety as we increase the number of encoded Scenarios. At some level, then, the most central representation extension is the creation of additional Scenarios.

- Extend and refine the domain ontology – The domain ontology is still missing a large number of concepts, from the concrete (e.g., representing real-world entities such as vehicles, weapons, etc.) to the abstract (e.g., representing tactical concepts and military conventions). In addition, the system should know more about the concepts it already has defined—that is it should have a richer set of inference rules bearing on those concepts.

- Extend and refine the Curriculum – The existing Curriculum taxonomy is based on analysis of (a subset of) the live tutoring sessions observed during the course of the project. It has been only lightly reviewed by SMEs. It has not been reviewed at all by representatives of any Army school that might potentially use ComMentor. Curriculum refinement then will require study of additional Scenarios, extensive SME review, and interaction with Army trainers and training psychologists.

- Extend the taxonomy of Scenes, Evaluations, and Dialogs – Extensions to the existing taxonomies of Scenes, Evaluations, and Dialogs may be suggested by experience with new Scenarios (as noted in the earlier discussion of "Runtime Refinements"). Some changes to the way these structures are used may require changes to the structures themselves.

Release and Packaging Refinements

- Clean up directory structure of distribution – The existing ComMentor distribution package still contains remnants of earlier experimental commitments that should be cleaned up. The packaging of resources required to run Scenarios is not yet completely consistent (e.g., too many icons are sitting in a large shared directory).

- Automatically launch system introductory video – An introductory video was developed to orient first-time students to the ComMentor system. That video should be automatically invoked when a student logs into the system for the first time. It should also be available to the student as an option on subsequent runs.

- Explore packaging of ComMentor application as a Java applet – ComMentor is implemented in pure Java, and as such should be convertible to applet format for Web delivery. Required work would include (a) creating the applet class to parallel the application class, (b) preparing a more compact runtime-only configuration of .jar files that eliminates all unneeded authoring-specific classes, (c) ensuring all file access (e.g., Scenario and Student-Model files) was routed to the server, and (d) exploring ways to sequence and stream large media objects (most notably the system-introduction video and Scenario introduction briefings).

- Develop methods for student record portability – Whether running in application or applet form, ComMentor must ultimately have (or tie into) some system for managing student records, in particular the Student Model. At a minimum, there must be a way for the student to transfer their records from installation to installation (e.g., by putting them on a floppy and taking them to a new machine). Especially when running as an applet, reliance on a central server for student record management is appealing.

- Develop methods for distributing system updates – If ComMentor is to grow and flourish, it must be possible to distribute system updates (e.g., new Scenarios or changes to shared knowledge-bases) with minimal effort. Again, a server-based system, where the most up-to-date versions of ComMentor data are provided from a centralized source on each system invocation is appealing. A multi-tier system that distributes updates as communication links become available would be even more robust.

- Develop methods for distributed and concurrent authoring – ComMentor knowledge bases are currently stored as files and can only be manipulated by a single user at a time. Making the files available on a network file server does not address the most important issue of controlling concurrent access so multiple authors do not overwrite one another's work. Moving to an application or database server configuration is required to usefully support distributed and/or concurrent authoring of ComMentor scenarios. Fortunately, infrastructure work to support this goal is already in process on another funded project.

- Develop methods for maintaining development and production releases and for tracking knowledge-base versions and consistent configurations – A centralized server, or multi-tier system, could be designed to combine support for distributed and concurrent authoring with record-keeping about which version of which knowledge-bases are required to work together in a consistent way, and to ensure that distributions to users were always consistent.

Appendix C

Sample Session Transcript

This appendix contains a transcript illustrating a run through the ComMentor system using the "Enemy Over the Bridge" scenario. The transcript is interleaved with figures that illustrate aspects of the user interface, and with commentary that discusses what the student and the system are doing at each point.

We start at with Figure 33 showing ComMentor's initial loading screen that displays a progress bar while the system is starting up. Figure 34 then shows the ComMentor splash screen, which provides two options for entering the system: (1) logging in a as a known student, or (2) creating a new student account. Figure 35 shows the creation of a new student account.

Figure 33. ComMentor's Initial Loading Progress Bar.

Figure 34. Splash-Screen with Log-In and New-Account Options.

Figure 35. New Student Account Creation Screen.

Best Available Copy

Once logged in, the system chooses a scenario for the student. In this case, the system starts the "Enemy Over the Bridge" scenario by beginning the scenario briefing sequence. Figure 36 shows the first slide in the briefing as it appears in the larger ComMentor screen layout. Along with this slide (and the ones that follow), the system plays a voice-over narration that reads the text from the slides, potentially allowing the student to focus on the graphics (e.g., force-structure diagrams, and situation maps) that appear on many slides. Figure 37 to Figure 46 show the full sequence of briefing slides for "Enemy Over the Bridge."

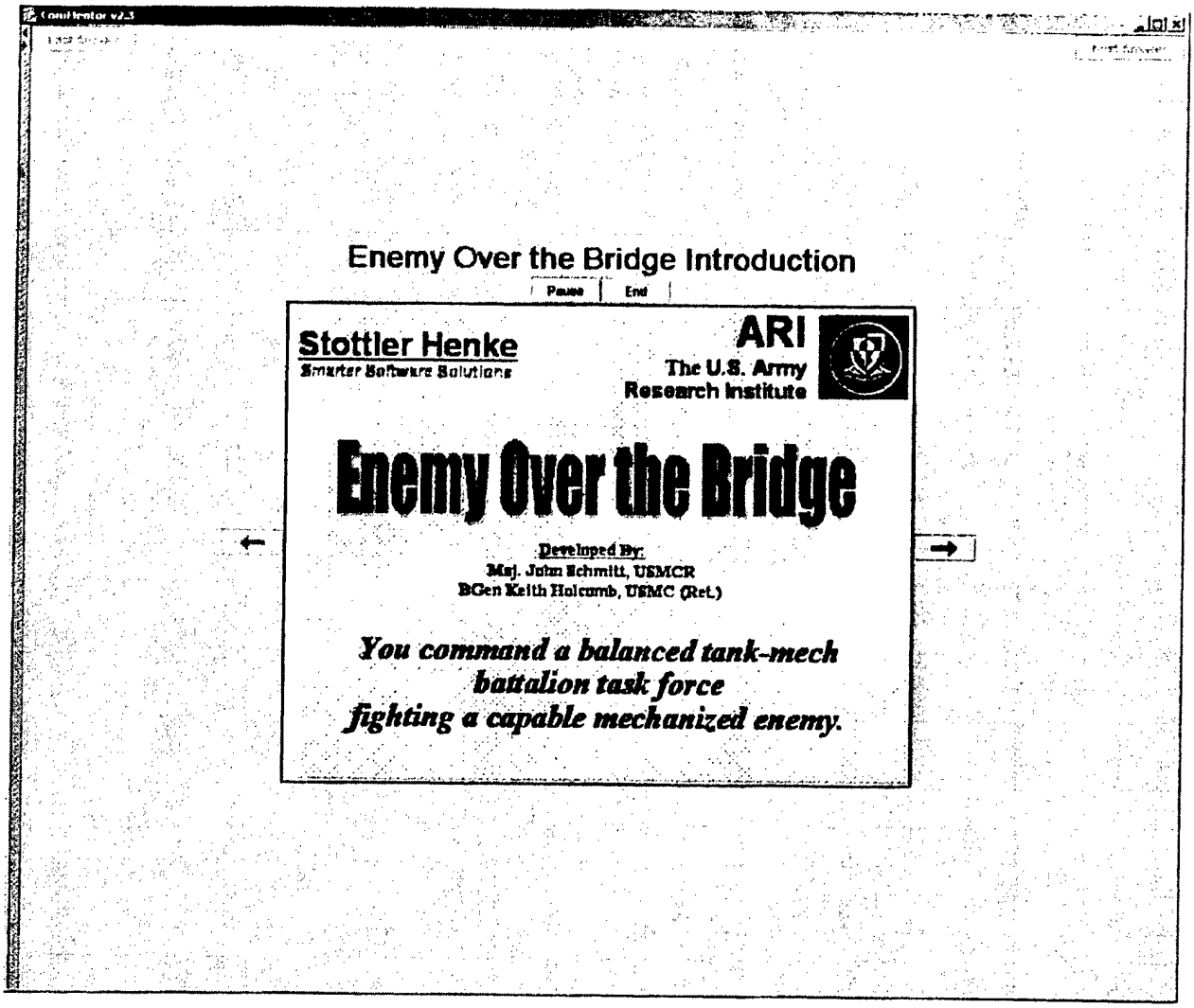


Figure 36. Start of Briefing for "Enemy Over the Bridge" Scenario.

Figure 37 is a standard introduction/disclaimer slide now included at the start of all scenario briefings based on experience with an earlier version of the system. When showing an earlier version of the system to active-duty Army officers, about half the subjects spent a significant amount of time pushing back about the details of the scenario, effectively avoiding engagement in the tactical problem. Our SMEs conceded this was a not uncommon reaction, but one that they could manage in live tutoring sessions, in part based on their authority as (retired) senior officers. The point of this slide is to orient students towards playing the game as given, rather than spending their time second-guessing the situation.

Tactical Decision Game Scenarios:

What follows is the set-up for a Tactical Decision Game. This scenario intentionally leaves out many details that might be available in a real situation, in order to more quickly get you into the action. For purposes of this training exercise, most missing details can be ignored, or you can fill them in with reasonable assumptions. In some cases you may be able to ask the computer for more details, or at least a refresher on this initial presentation.

The details provided—including force structures, terrain layouts, and the initial tactical situation—are meant to describe a plausible and realistic situation rather than one that is necessarily doctrinally ideal. Remember, command requires you to deal with events as they are rather than as you would want them to be.

Figure 37. Second Briefing Slide for "Enemy Over the Bridge."

There is a fairly standard progression to the slides in ComMentor's introductory briefings. Generally they start with an introduction to the forces. Figure 38 is a representative slide, presenting the basic facts about the forces that the student gets to command in the scenario. Note that we have given the student a battalion, even though we expect this scenario to be played by a captain. ComMentor tries to pull students out of their normal operating zone of comfort, challenging them to think about problems and in ways that go beyond their well-practiced habits. While a single slide suffices here to lay out the student's forces, in our FCS scenario the comparable discussion—covering a wide range of novel (notional) capabilities—ran on for over ten slides.

Enemy Over the Bridge: Your Forces

You command a balanced tank-mech battalion task force consisting of 2 tank-heavy company teams (A and B), two mech-heavy company teams (C and D), a scout platoon and a mortar platoon, plus your medical, support and maintenance platoons organized into combat and field trains currently moving together.

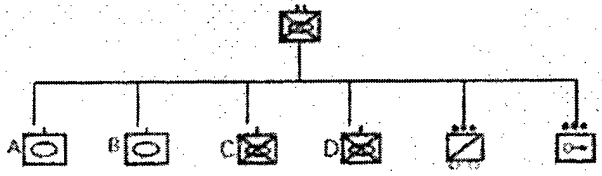


Figure 38. Third Briefing Slide for "Enemy Over the Bridge."

Figure 39 continues the normal progression by talking about the enemy forces. In this case, little detail is known about the enemy. In other scenarios, there may be moderately detailed intelligence about the enemy force structure, doctrine, and/or intentions.

Enemy Over the Bridge: The Enemy Forces

You are fighting a capable, mechanized enemy equipped with T-62s, BMPs and BRDMs and supported by towed and self-propelled artillery.

Figure 39. Fourth Briefing Slide for "Enemy Over the Bridge."

Figure 40 begins to lay out the big picture context surrounding the scenario. Where is this happening? What are we fighting for? What has been going on prior to the start of this scenario? This slide was also revised based on feedback from early presentations to Army officers, so that it now includes an explicit summary of the rules of engagement that will hold during the scenario.

Enemy Over the Bridge: The Context

U.S. forces have been deployed to "Host Nation" to support its effort to roll back an invasion by "Enemy Nation" which already occupies the western areas of the country. You have been engaged in conventional combat for several weeks.

The rules of engagement allow you to engage enemy forces at any time and place they are encountered, but call for efforts to minimize civilian causalities and damage to civilian infrastructure.

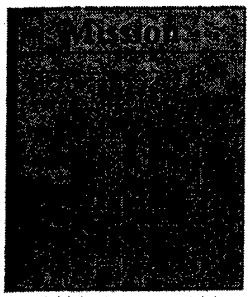
Figure 40. Fifth Briefing Slide for "Enemy Over the Bridge."

Figure 41 is a critical slide. It describes the current situation facing the student's own forces. It explicitly calls out the student's mission. It also introduces the situation map that will play a central role throughout the scenario. In this scenario, the student's battalion is moving in from the east approaching a river, across which they are to launch an offensive in 48 hours. Host nation forces are supposed to be holding the river line, and the bridge. The battalion is heading for an assembly area where they can prepare for the offensive. The mission is: "Attack west across river in 48 hours as advance guard in order to facilitate brigade's attack to destroy enemy forces in zone."

Figure 42 was another late insertion to the briefing. It is essentially a map key. It explains the color coding of fields, forests, and hills, points out the distance and contour scales, and adds a bit of useful trafficability information.

Enemy Over the Bridge: The Situation

Host-nation forces hold the bridge and the river line to your west. You have been told the river is unfordable. Reconnaissance elements are operating west of the river. In 48 hours, the division begins a major offensive west across the river to destroy enemy forces in zone, with the main effort in your brigade's zone. Your task force will spearhead the brigade's attack.



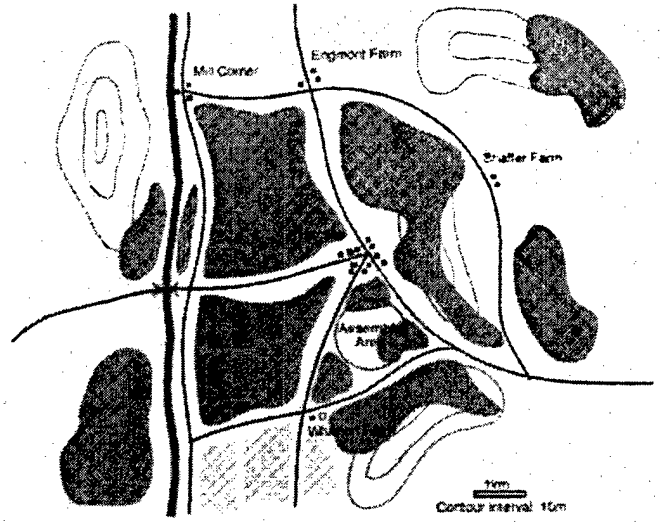


Figure 41. Sixth Briefing Slide for "Enemy Over the Bridge."

Enemy Over the Bridge: The Map/Terrain

On this map, the ochre-colored, geometrically-shaped areas represent farm fields and the small black squares are buildings. The green regions are wooded areas, which are generally obstacles to vehicle movement though not to infantry movement. Vehicles can tuck into the woods and hide, and even move some distance into the woods very slowly and carefully, but that's it.

The brown lines are contour lines for elevation, where the interval is 10m, so the hill masses are not significant obstacles to vehicles or people, although they may offer good fields of fire or observation. Also notice the 1km scale at the lower right.

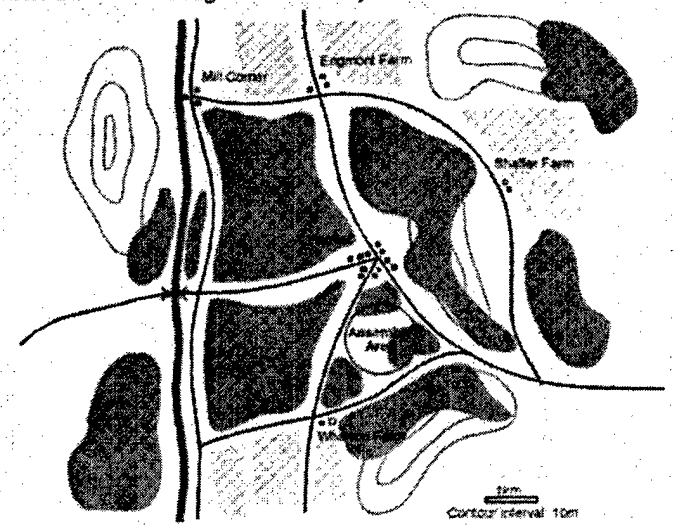


Figure 42. Seventh Briefing Slide for "Enemy Over the Bridge."

Figure 43 begins the sequence of slides that describe the specific situation that forms the crux of the scenario's problem. It is typical for scenario briefings to culminate in a slide sequence that describes a specific set of events, unfolding as the student's unit is in the middle of carrying out its planned activities, which call for some new time-critical response.

Enemy Over the Bridge: Initial Conditions

You have been instructed to occupy the assembly area shown on the map east of Hamlet in preparation for the 0500 attack the morning after next. You are moving to the assembly area as shown

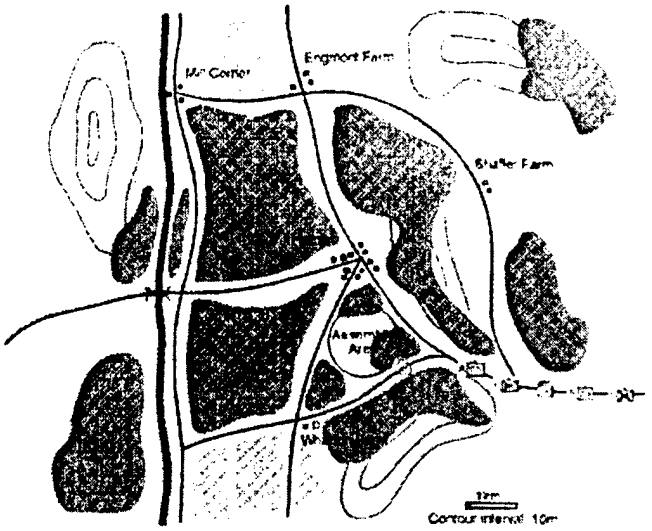


Figure 43. Eighth Briefing Slide for "Enemy Over the Bridge."

Enemy Over the Bridge: The Action Begins...

At 0100 your scout platoon, which is forward reconnoitering the route to the assembly area, makes the following report:

Have just made contact with a host-nation motorized reconnaissance patrol that was operating west of the river but about 2 hours ago was forced east across the river under fire. They came across the bridge and then via Wharton Farm. The reconnaissance patrol leader reports there is no sign of friendly forces holding the river line or the bridge and that enemy mech infantry and some tanks have been moving east across the bridge for almost 2 hours. He says he counted 10 T-62s in the last half hour; does not know how much mech. He says he has reported this twice to his higher headquarters. Over.

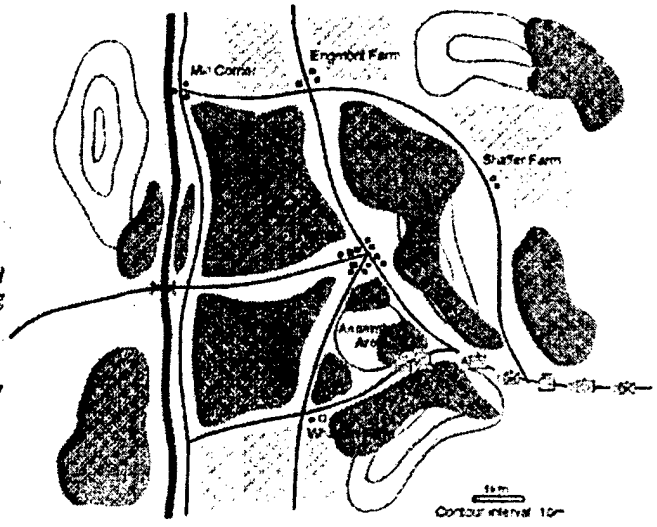


Figure 44. Ninth Briefing Slide for "Enemy Over the Bridge."

The evolution of the situation map from Figure 43 to Figure 44 to Figure 45 illustrates the way the specific situation (and the student's knowledge of the situation) is evolving in time. New force icons show up on the map as the situation unfolds.

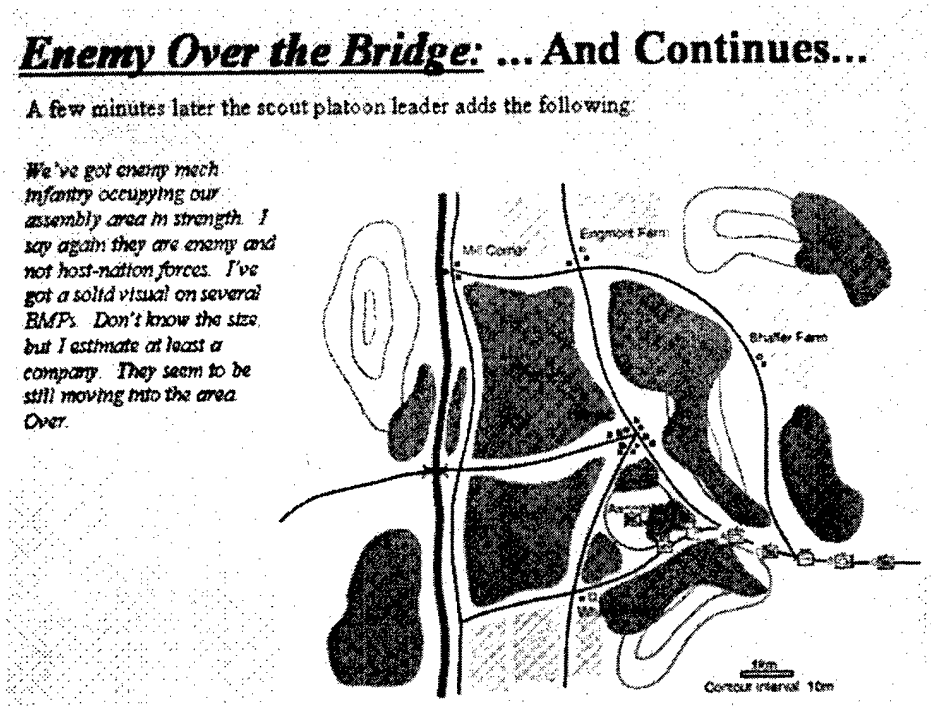


Figure 45. 10th Briefing Slide for "Enemy Over the Bridge."

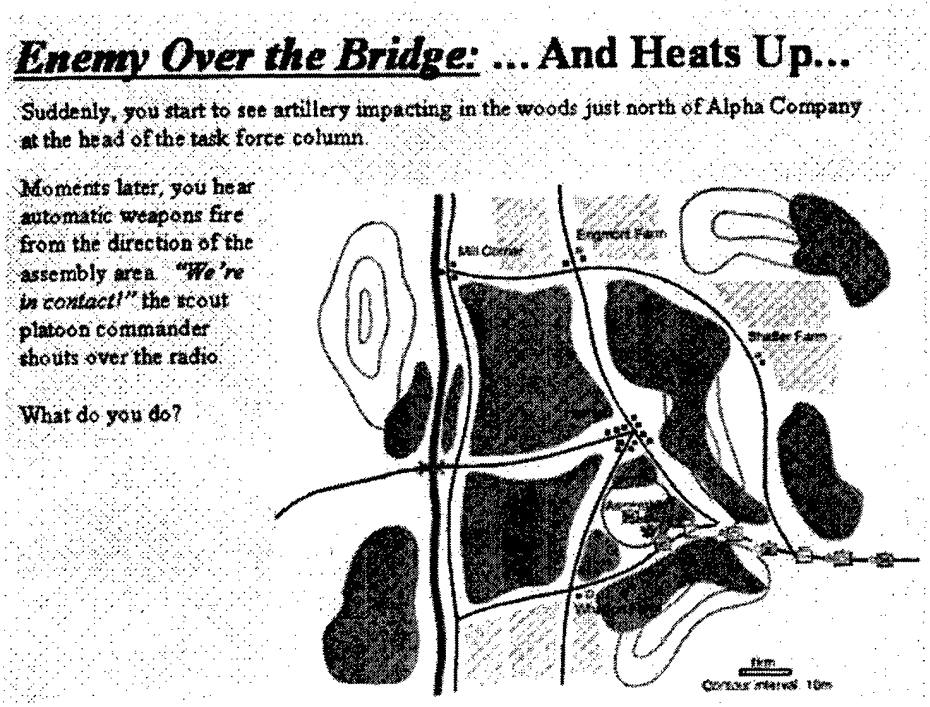


Figure 46. 11th Briefing Slide for "Enemy Over the Bridge."

Figure 46 represents the end of the introductory briefing. When the system transitions from this slide, it generates the tutor utterances below and moves to the main interaction screen shown in Figure 47. Note the situation map in Figure 47 matches with the state of the map in the final briefing slide.

Tutor Please enter your orders and reports.

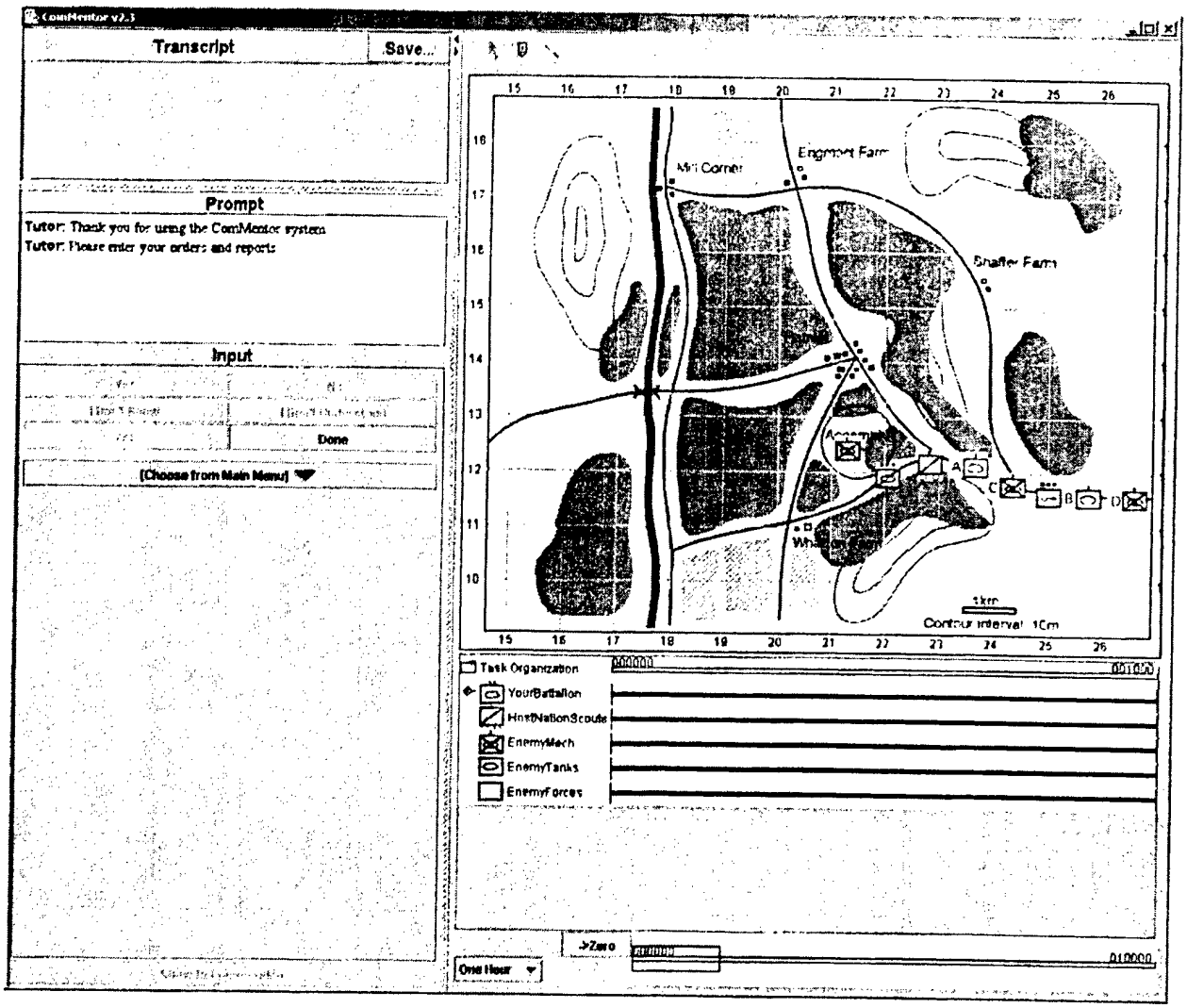


Figure 47. Main Interaction Screen with Initial Order-Elicitation Prompt.

In this sample run, we assume the student adopts a plan that is typical of what was observed in live tutoring sessions, in that the major response is to launch a sharp attack on the enemy forces in the assembly area. Figure 48 shows the student in the middle of issuing their first tasking: an order to Alpha Company to move into the assembly area from the south and attack the enemy mech forces. The Tasking form in the Input area has nine fields, three of which are filled in automatically—"Who," "Where," and "Route"—when the user drags a force icon on the map to initiate a tasking. With guidance from our SME, this form has been substantially simplified since its initial introduction, in order to improve usability and focus the student on the essentials of the tactical situation. The screen shot shows the student in the middle of filling in the "Enemy" using the mouse pointer; the arrow next to that field is gray, indicating it has been clicked on, and the pop-up menu on the situation map shows the choice the student is making by clicking on the enemy mech icon in the assembly area.

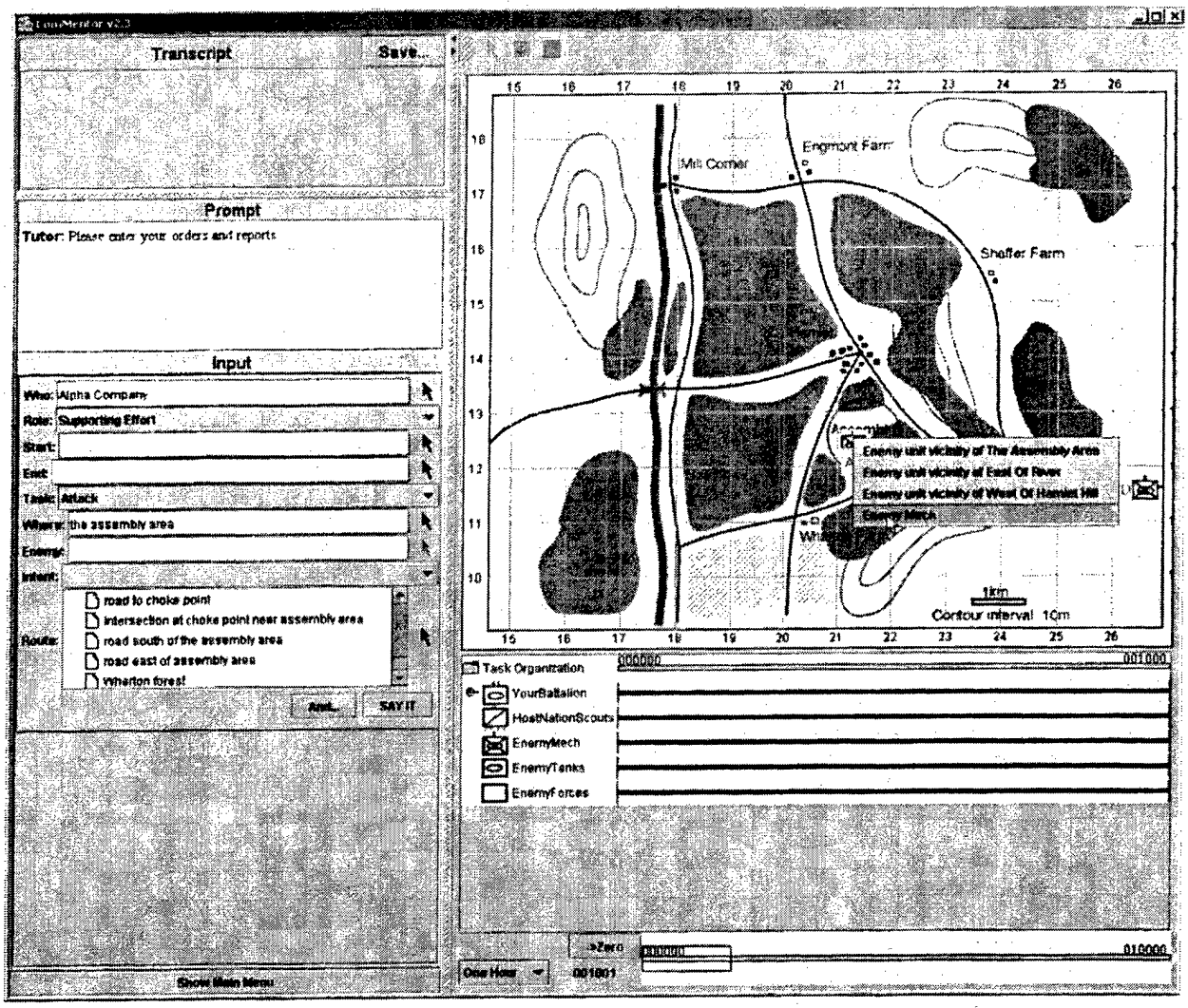


Figure 48. Initial Orders: Filling in Alpha Company's Tasking.

When the student completes and submits this tasking, the following two entries in the transcript are created, and the screen advances to that shown in Figure 49:

Student Who: Alpha Company Role: SupportingEffort Task: Attack Where: the assembly area Enemy: Enemy Mech Intent: Fix Route: road to choke point, intersection at choke point near assembly area, road south of the assembly area, road east of assembly area, and the assembly area.

Tutor Roger. Anything further?

Note that we have not invested any effort in dressing up the system's paraphrase of the student's input. Student transcript entries appear as a straight concatenation of the field names and their contents. The primary intent is to provide a way for students to review their input to make sure the system captured what they intended to say. The standard cycle is for new student input to appear in the Transcript pane, moving the Tutor's previous output into the Transcript at the same time, and making room for the new Tutor output in the Prompt pane.

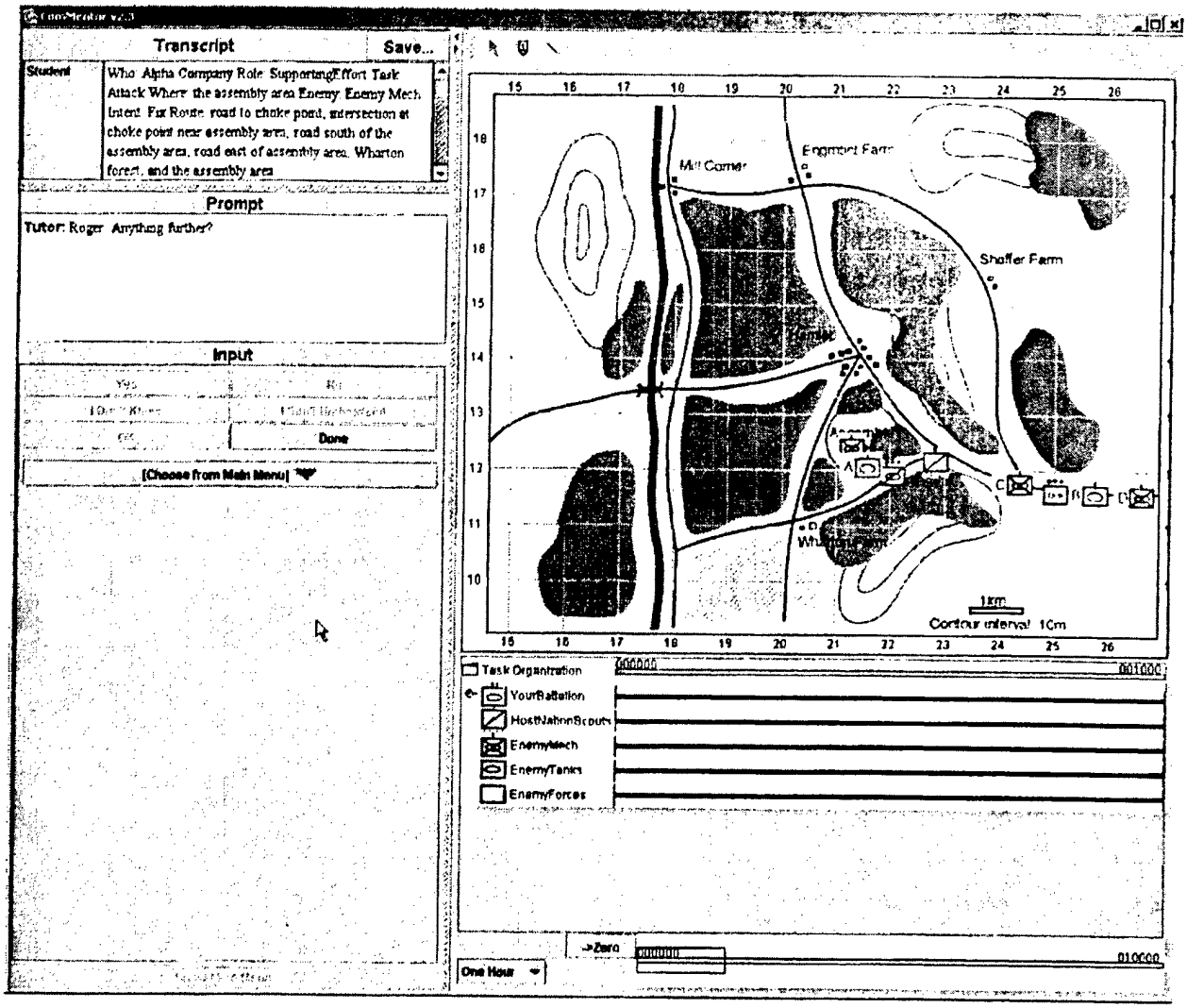


Figure 49. Initial Orders: Translation and Response to Alpha Company's Tasking.

The elicitation of initial orders is a simple loop that continues until the student clicks the "Done" button. In this run, the student enters four taskings, one for each of the major combat companies (had the student not done so, the system would have prompted them to enter orders for each company in an effort to elicit a more complete plan that makes better use of available resources). Figure 50 shows the state of the system after those four taskings have been issued. For this screen shot, the Transcript pane has been expanded so the entire transcript to date can be read without scrolling.

The student has assigned Charlie-Company to move into the assembly area from the east as the main effort, aiming to wipe out the enemy forces. Bravo-Company has been positioned at the top of Wharton Hill to cover the assembly area. Delta-Company has been designated as a reserve force and tucked into the south side of Wharton Hill.

Student Who: Charlie Company Role: MainEffort Task: Assault Where: the assembly area Enemy: Enemy Mech Intent: Destroy Route: eastern fork, road to choke point, forest on Hamlet hill, intersection at choke point near assembly area, and road east of assembly area.

Tutor Roger. Anything further?

Student Who: Bravo Company Role: SupportingEffort Task: Cover Where: Wharton Hill Enemy: Enemy unit vicinity of The Assembly Area Intent: Suppress Route: eastern road, south of Wharton Hill, and Wharton Hill.

Tutor Roger. Anything further?

Student Who: Delta Company Role: ReserveForce Where: south of Wharton Hill Route: eastern road and south of Wharton Hill.

Tutor Roger. Anything further?

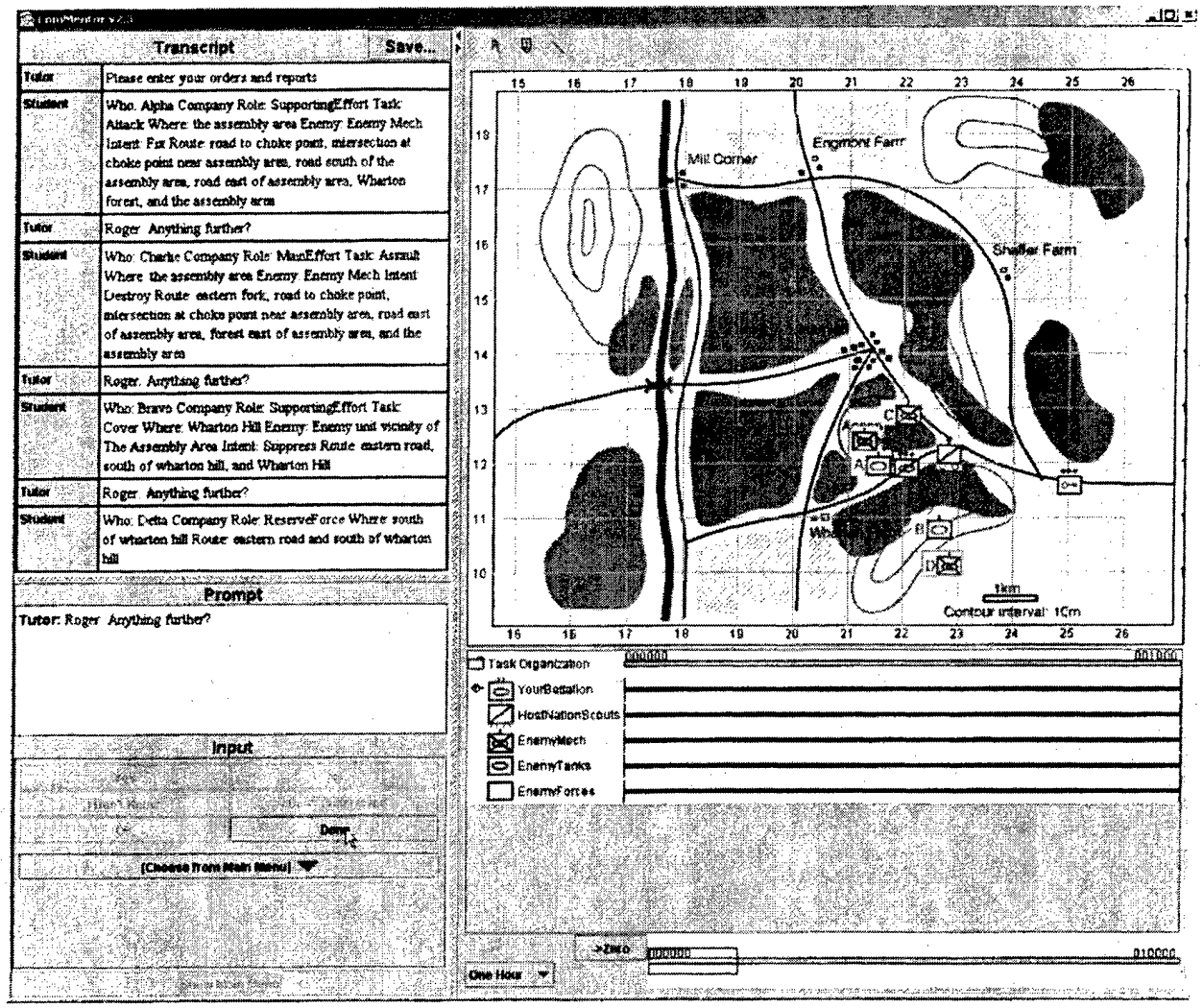


Figure 50. Initial Orders: Layout After Orders Issued to A-D Companies.

Student	Done.
Tutor	Alright. Let's move on.
Tutor	We're going to discuss this situation and what you've proposed to do about it.
Tutor	Let's start by talking about the red forces and their situation.
Tutor	What sort of force do you think you're facing in your area east of the river?

Figure 51 shows the screen after the student has clicked "Done" to indicate that they have no further orders to enter. The tutor responds with a series of short utterances acknowledging the end of the order-elicitation phase, and transitioning to the first topic of discussion. In this scenario (and quite likely in the majority of scenarios), the first topic of discussion is always an exploration of the student's understanding of the Red situation. What kind of Red forces are they facing? What are those forces up to? Where might enemy forces be positioned? The discussion starts with a basic assessment of the number and type of enemy forces in the vicinity.

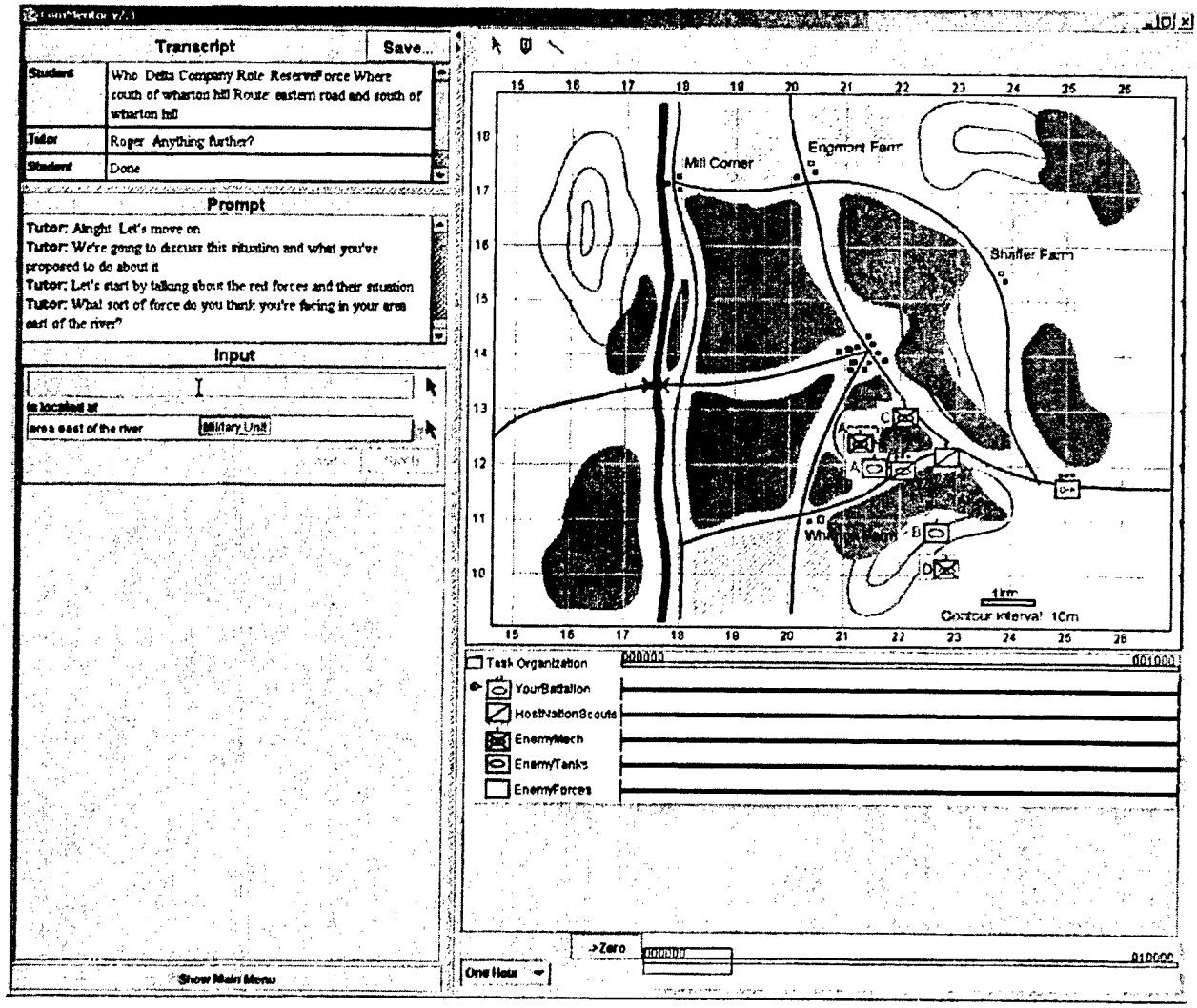


Figure 51. The Red Situation: The Tutor's 1st Question.

When ComMentor asks a question, it generally primes the Input pane with a form that provides a way to answer the question. In this case, it activates the "[MilitaryUnit] is located at [Place]" form, and pre-fills the [Place] field with "area east of the river" both to simplify the students input entry problem, and to clarify the meaning of its question. Note that the form's remaining empty field is highlighted in yellow and the "SAY IT" button is grayed out. This indicates the field is required, and the form is not yet complete. Both fields in this form are type-in boxes in which the student is encouraged to type a noun phrase that matches the desired type of information. Both fields also have arrow icons next to them, indicating that the student can use the mouse pointer to designate a filler (as was done when filling the first tasking form in Figure 48).

Figure 52 shows the student's completed answer. In this case, we assume the student has taken too narrow a view of the situation or the question, and has simply clicked on the one enemy force icon shown on the map to fill the phrase "Enemy Mech" into the form. When they submit this answer, the tutor comes back asking for clarification.

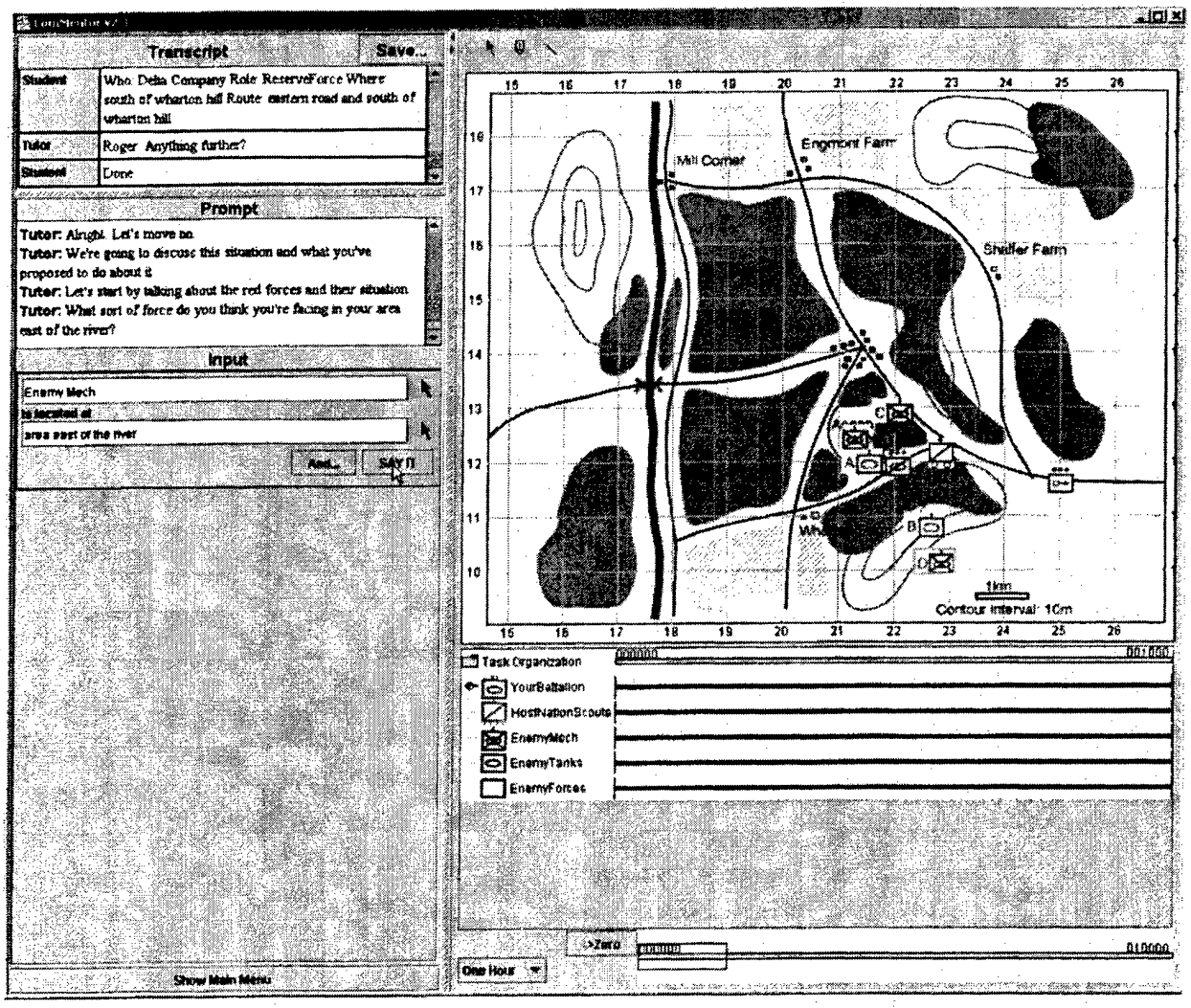


Figure 52. The Red Situation: The Student's 1st Answer.

Student Enemy Mech is located at area east of the river.

Tutor Can you characterize what sized unit you are seeing here?

The tutor has been scripted to watch for certain types of incomplete answers. It recognizes that while the student has indicated a type of force, no size has been specified. The tutor thus explicitly asks what sized enemy unit is being seen here, and combines the follow-up answer with the original answer. This time, as shown in Figure 53, the student realizes that the tutor is asking about all the enemy forces east of the river and types in the word "bat" (as an abbreviation for battalion) which the tutor accepts as a reasonable answer. Had the student answered something else, the tutor was prepared to engage in a more detailed discussion of the various reports received during the briefing (e.g., at least a company of mech in the assembly area, and at least 10 T-62s somewhere east of the river). In this way, the tutor would work through the argument in favor of concluding that there is probably an enemy battalion moving east of the river.

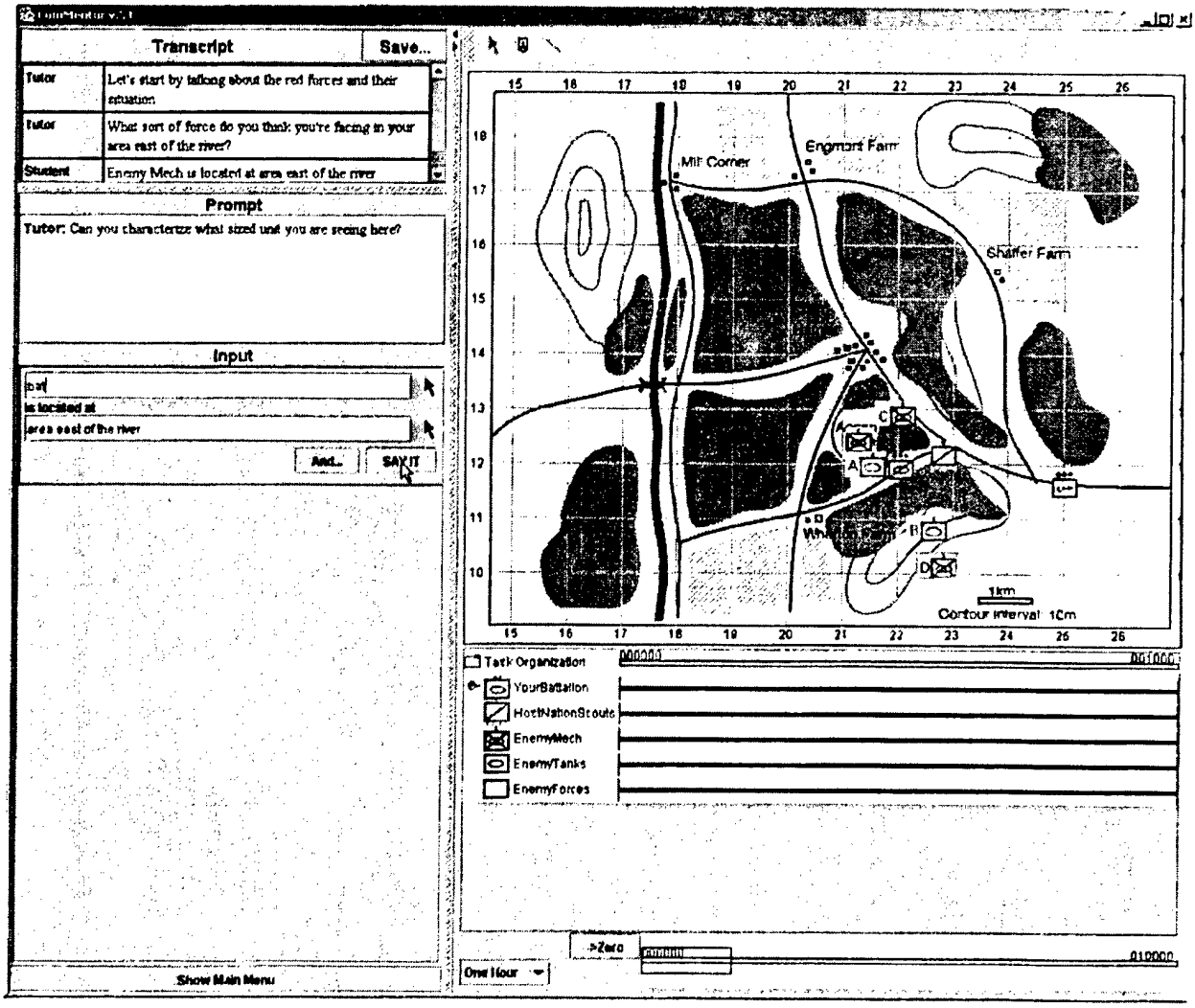


Figure 53. The Red Situation: Student's Answer to Follow-Up on 1st Question.

Student Bat is located at area east of the river.

Tutor Yes. Given that you've seen at least a company of enemy mech in what was supposed to be your assembly area, and given reports of at least 10 T-62s, plus the fact they've been moving over the bridge for at least 2 hours, it seems reasonable to assume that there's something on the order of a red battalion already east of the river.

Tutor What do you think the enemy is up to?

Since the student seems to have reached the appropriate conclusion, Figure 54 shows the tutor summarizing a line of argument without working through it in detail. It then moves on to its next question: "What do you think the enemy is up to?" Here the tutor primes a form that allows the student to succinctly express a task the enemy might be doing, and to whom or where it is doing it.

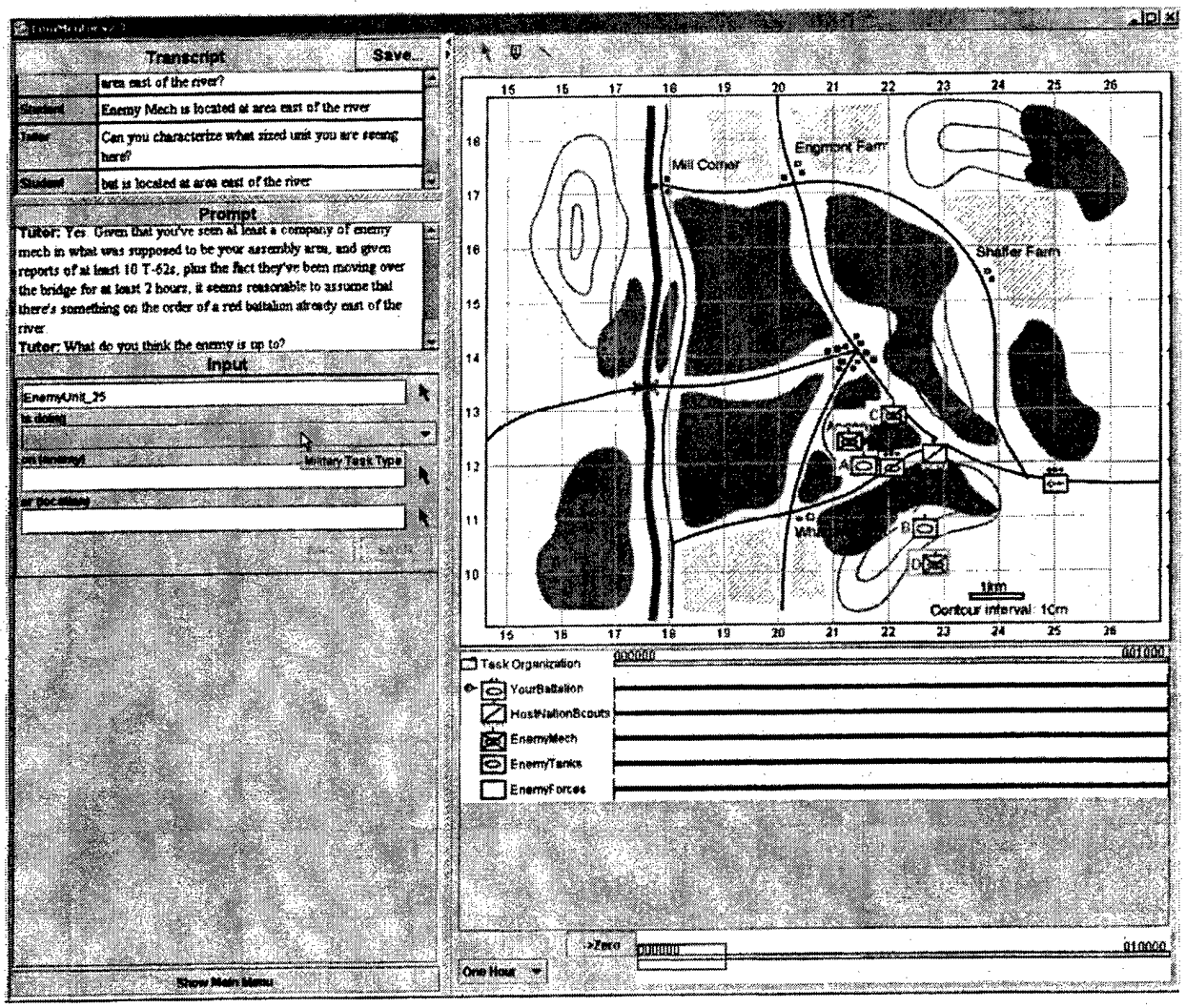


Figure 54. The Red Situation: The Tutor's 2nd Question.

Figure 55 shows the student's initial answer to the question about enemy activity. Again, the student starts out interpreting the question and the situation too narrowly. The student says the enemy is attacking his battalion task force.

Student EnemyUnit_25[21] is doing Attack on (enemy) the task force.

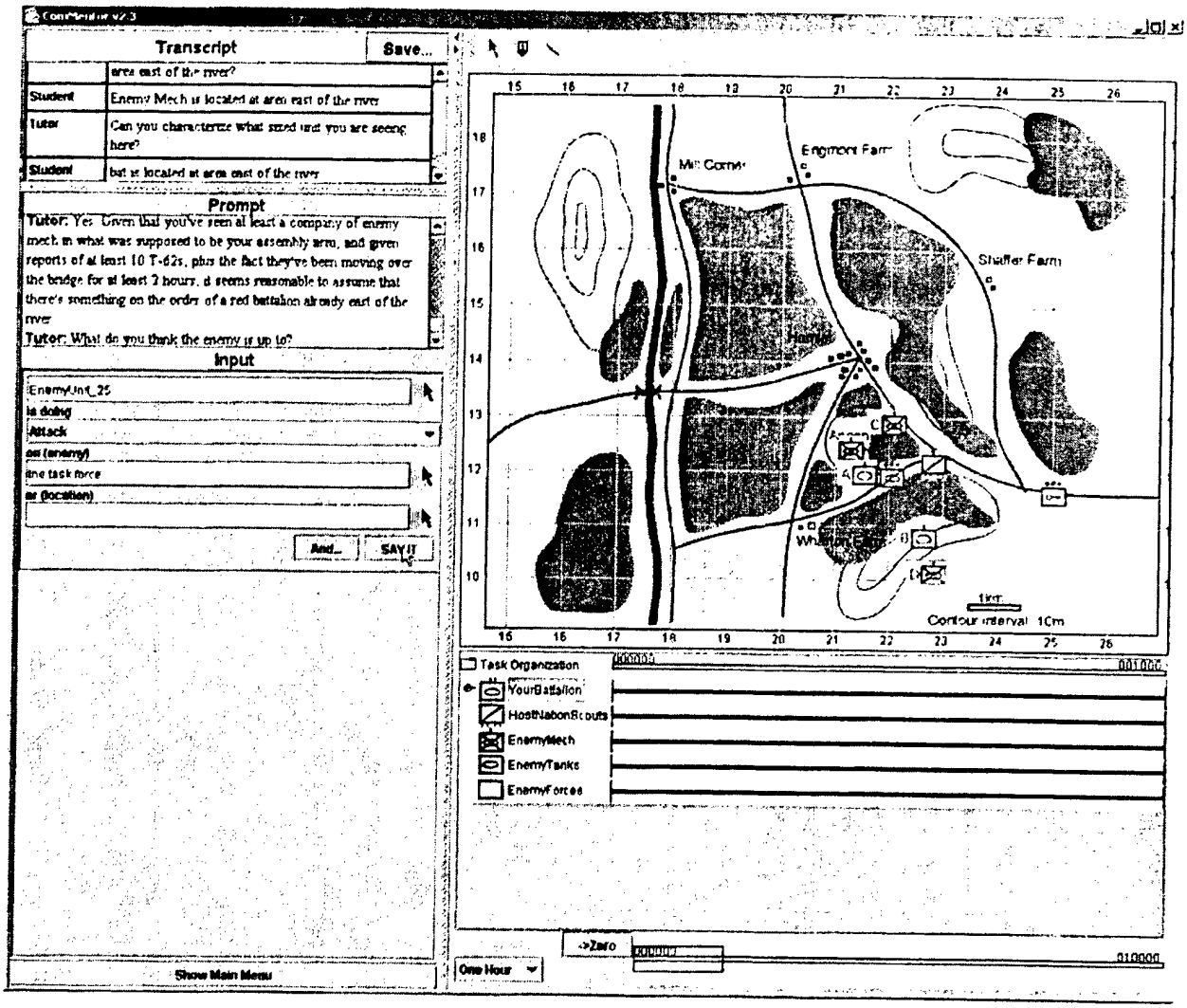


Figure 55. The Red Situation: The Student's 2nd Answer.

Tutor Yes, but where are they doing that Attack?

Student EnemyUnit_26 is doing Attack or (location) the assembly area.

The tutor is actually looking to focus the student on the location of the attack, rather than the target (which is assumed to be Blue and/or host-nation forces in general). The follow-up question, "Yes, but where are they doing that attack?" directs the student to focus on the

[21] Note that throughout the transcript, the appearance of constructs such as "EnemyUnit_25" (with different numbers on each occurance) is simply ComMentors way of referring to some unknown Red forces. A somewhat more graceful way to print those out might be the expression, "some enemy forces."

location. Again, the student answers narrowly, saying the attack is at the assembly area. Experience with students who actually responded this way could prompt additional authoring; currently, however, the tutor is not prepared with any special behavior to deal with a student who focuses so narrowly on his own forces and his own location. Instead, the tutor accepts the answer that the enemy is attacking at the assembly area as an example of the more general point that the enemy is attacking east of the river. It then moves on to its next question that is intended to start getting at possible dispositions of the enemy, east of the river.

Tutor Yes, since you've got most of a battalion on your side of the river, it's not unreasonable to guess that the enemy has beaten us to the punch, launching an offensive to the east, just as we were planning to launch one to the west.

Tutor If a red battalion has just crossed over the river in advance of a major red offensive, what do you think that battalion is now trying to do?

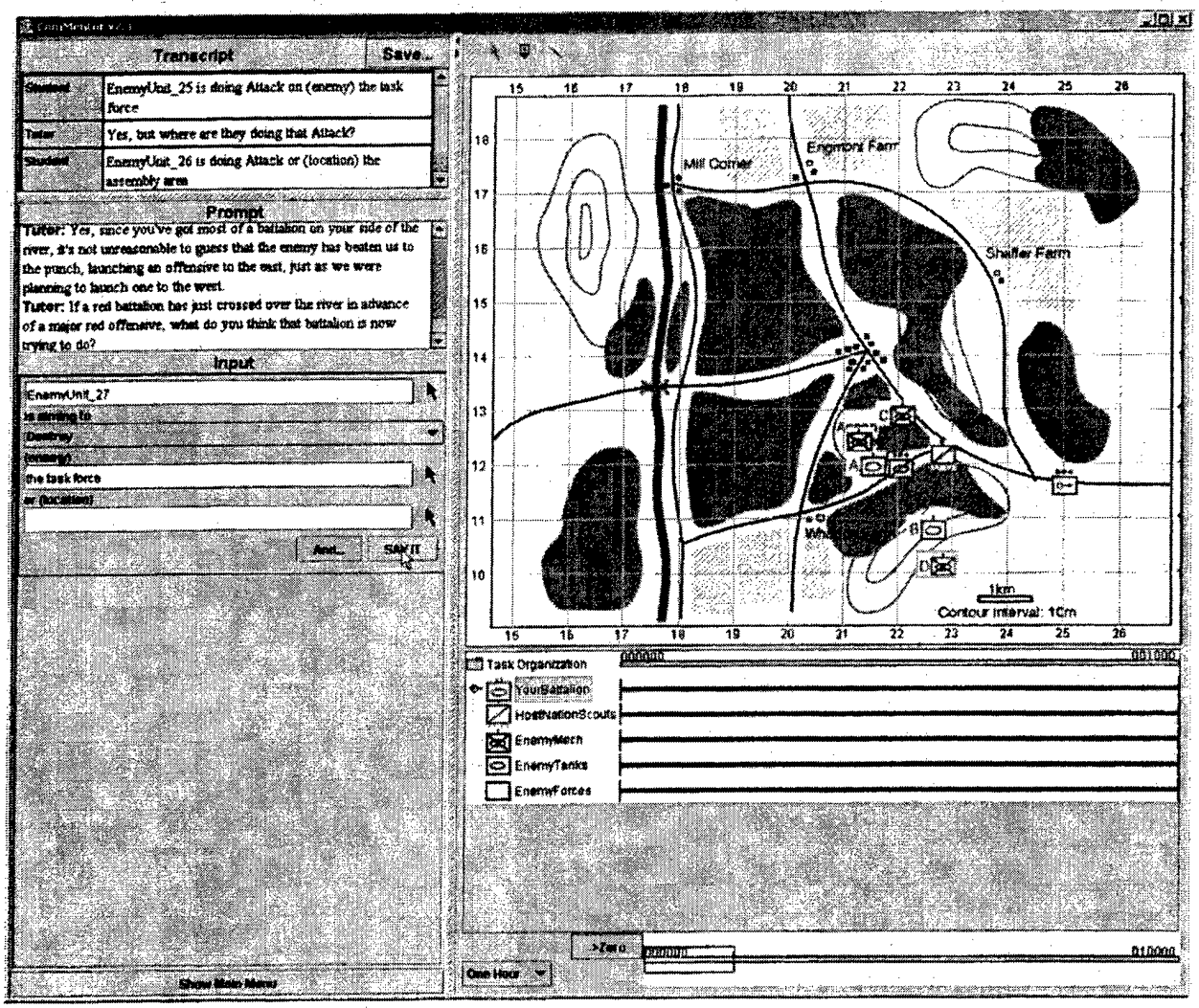


Figure 56. The Red Situation: The Student's 3rd Answer.

Student EnemyUnit_27 is aiming to Destroy (enemy) the task force.

As shown in Figure 56, the student continues to focus narrowly, claiming that the enemy is aiming to destroy his battalion. In return, the tutor re-asks the question, this time with a hint, trying to put the student into the red commander's shoes. This time, the student starts to see what the tutor is getting at, and in Figure 57 responds that the enemy is probably trying to secure the bridge. However, this is still a bit narrower than what the tutor is looking for. So in Figure 58 it asks another follow-up question, this time as a simple yes/no question, intended to get the student thinking about where the enemy might be beyond the bridge itself.

Tutor	What would you do if you had just pushed a battalion into enemy territory across a bridge?
Student	EnemyUnit_28 is aiming to Secure or (location) the bridge.
Tutor	OK. They're looking to secure their hold. Is it fair to say they're trying to secure a bridgehead?
Student	Yes.

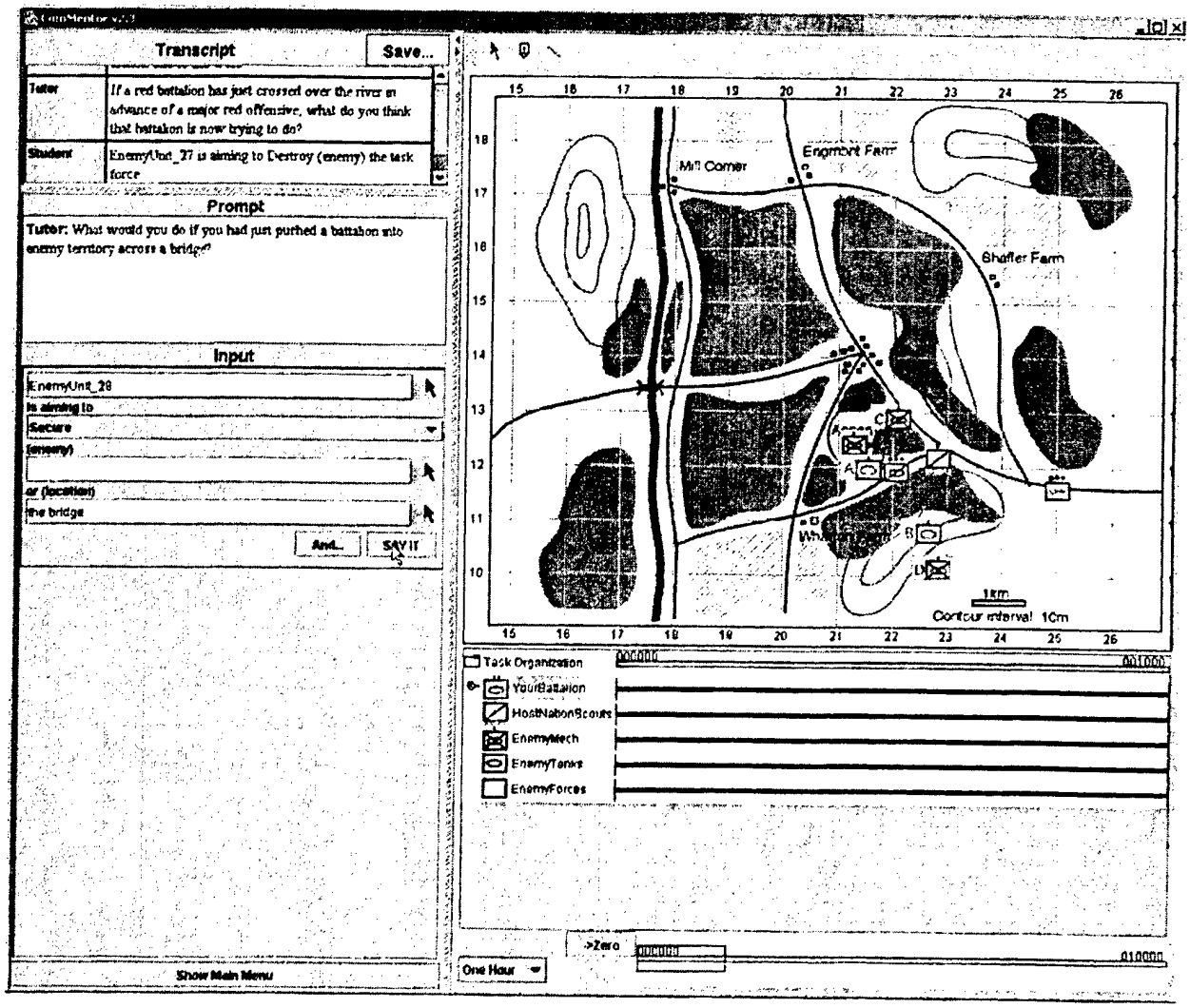


Figure 57. The Red Situation: The Student's Answer to the 3rd Question's 1st Follow-Up.

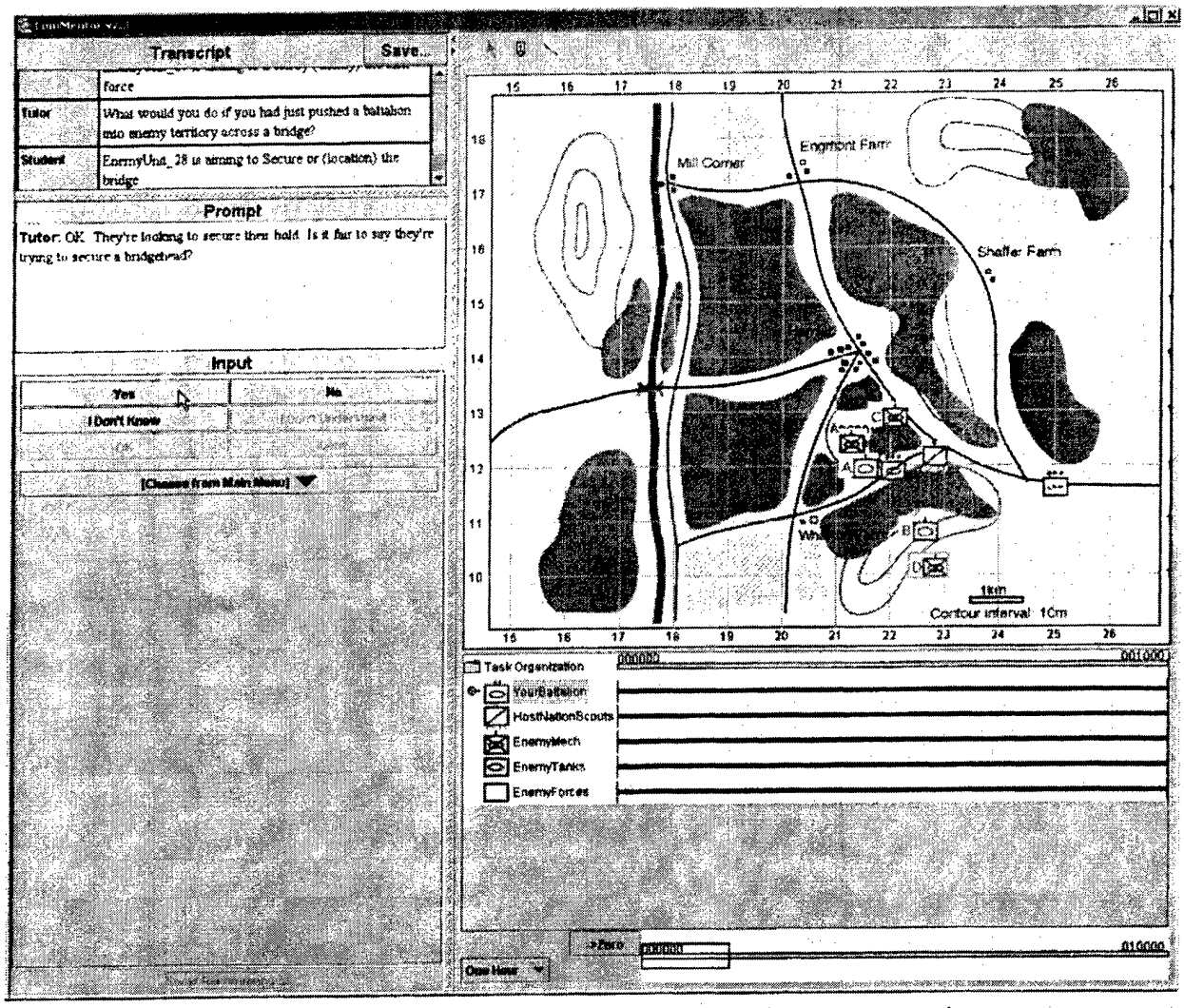


Figure 58. The Red Situation: The Student's Answer to the 3rd Question's 2nd Follow-Up.

The tutor has finally gotten the response it was aiming for, even if it had to try several times, and in the end, lead the student quite heavily. It summarizes its point, and then asks the student to start filling in details of where Red might be found as part of this likely bridgehead. In Figure 59 the student is in the middle of assembling their answer. The "might be located at" list box can hold any number of locations. First the student used the mouse pointer to enter "the bridge." In the figure, they are in the midst of entering "the assembly area" as a second value using the mouse.

Tutor Absolutely. If we had just managed to push a battalion across a river into enemy territory, it's safe to say that we would be thinking about how to secure the bridgehead for follow-on forces.

Tutor So if the red forces you are seeing are trying to create a bridgehead for a red offensive, tell me some of the locations you think Red might be occupying.

Student EnemyUnit_30 might be located at the bridge and the assembly area.

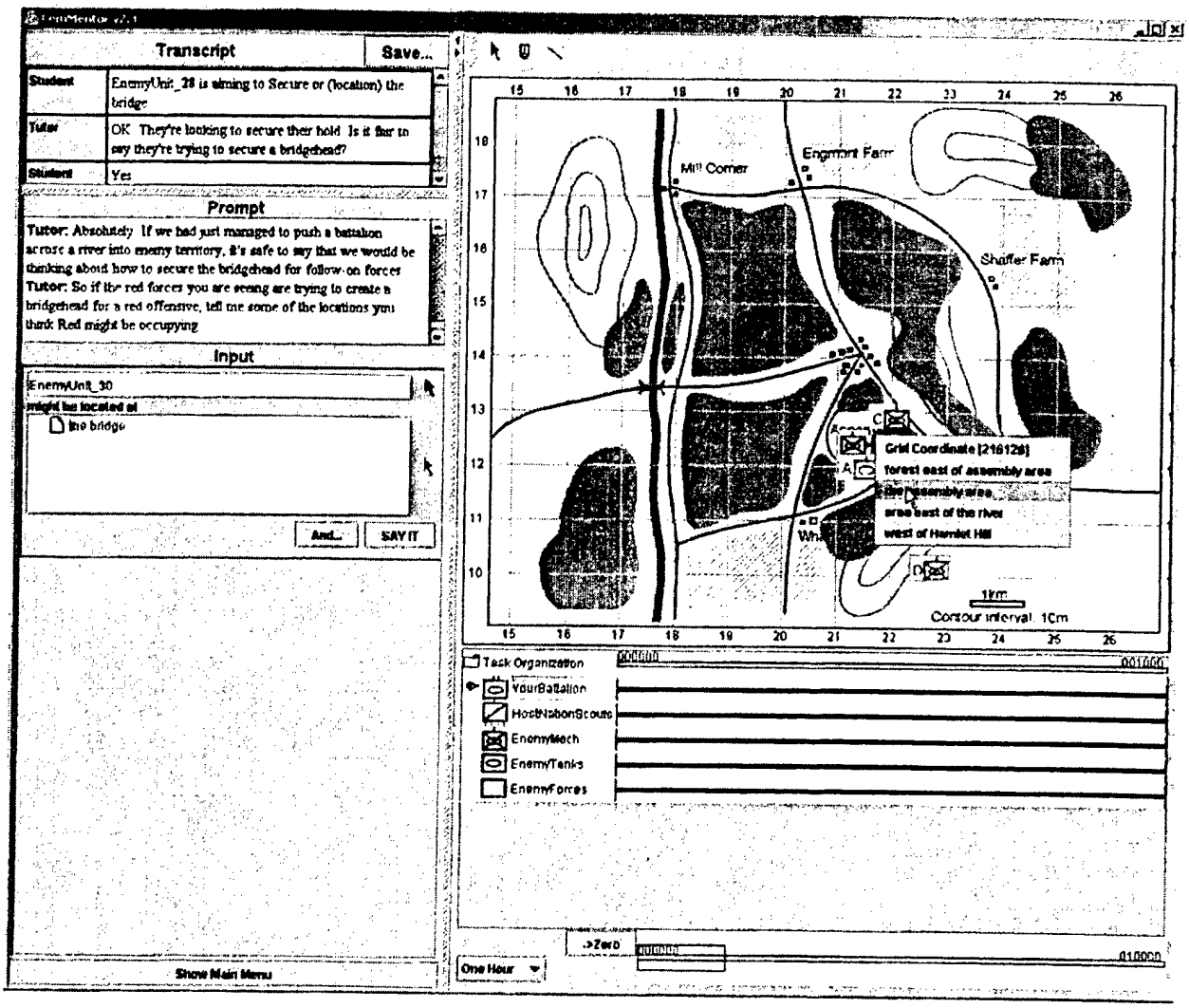


Figure 59. The Red Situation: The Student's Answer to the 4th Question.

The student's answer leaves out many of the likely sites for Red forces. The tutor follows up with two more questions to suggest places the student might consider. Figure 60 shows the student responding to the first of these with "Hamlet Hill" and "Wharton Hill." The transcript segment below also shows the second follow-up and the student's answer.

Tutor	Can you pick out any high-ground that Red might want to grab for his bridgehead?
Student	EnemyUnit_31 might be located at Hamlet hill and Wharton Hill.
Tutor	Can you pick out any junctions that Red might want to grab for his bridgehead?
Student	EnemyUnit_32 might be located at Intersection east of bridge.

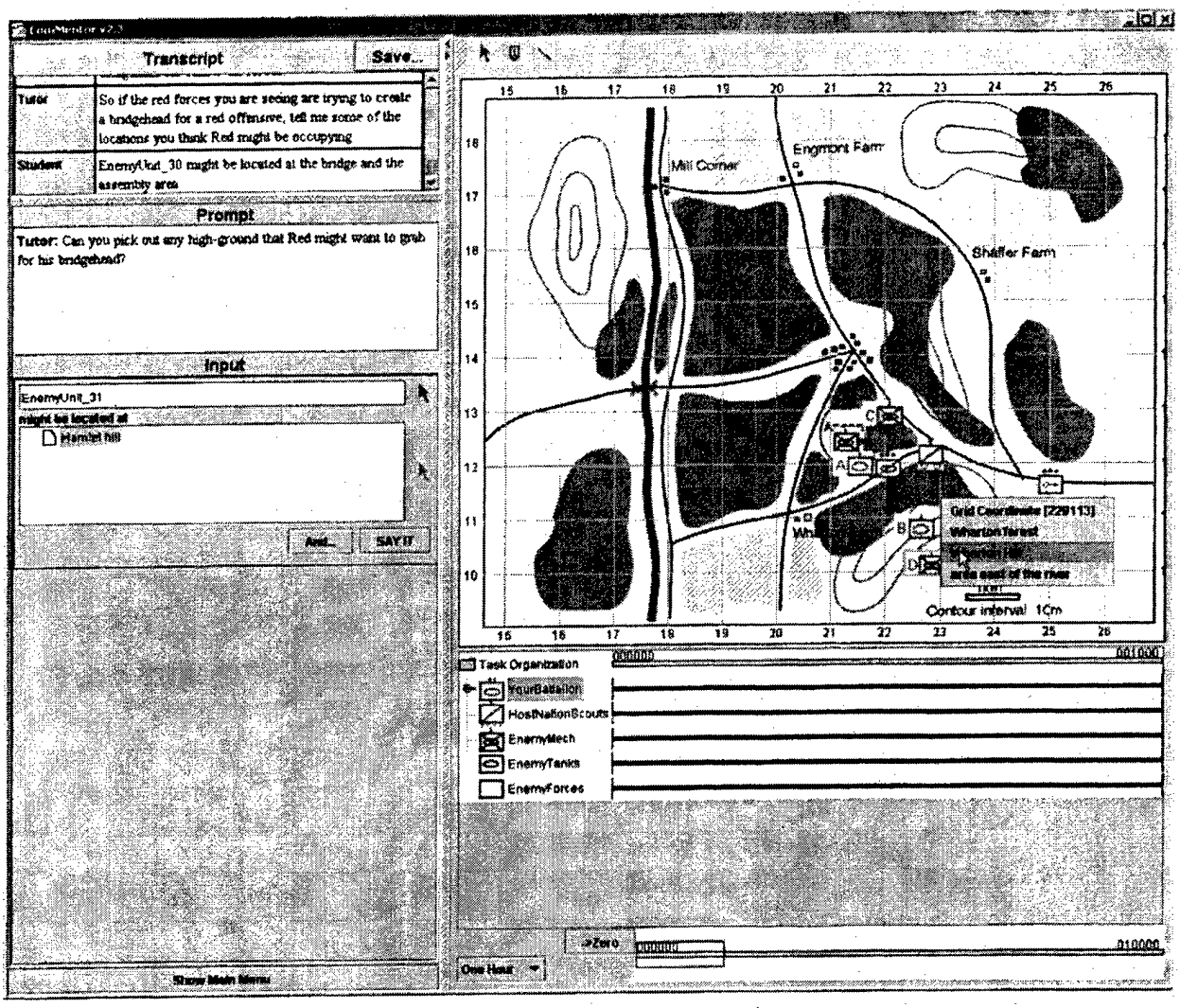


Figure 60. The Red Situation: The Student's Answer to the 4th Question's 1st Follow-Up.

Once the tutor has elicited as many suggestions from the student as possible, it starts in discussing the various nominated places. First it passes quickly over the suggestion that the enemy is on the bridge itself, then it moves on to consider why the enemy might be at the assembly area. Figure 61 shows the student responding to a question about the value of the assembly area. The interesting aspect of this figure is the relative complexity of the menu choices available to describe the tactical significance of a place. This is actually a significantly simplified and re-organized version of such choices, guided by input from one of our SME's

Tutor	Being right on the bridge is certainly reasonable, but of course it's not possible to secure the bridge just by sitting on it.
Tutor	Let's consider why Red might want to have forces in what was supposed to be your assembly area.
Tutor	What is the most obvious use for Red of the assembly area?
Student	The assembly area has tactical significance as AssemblyArea.

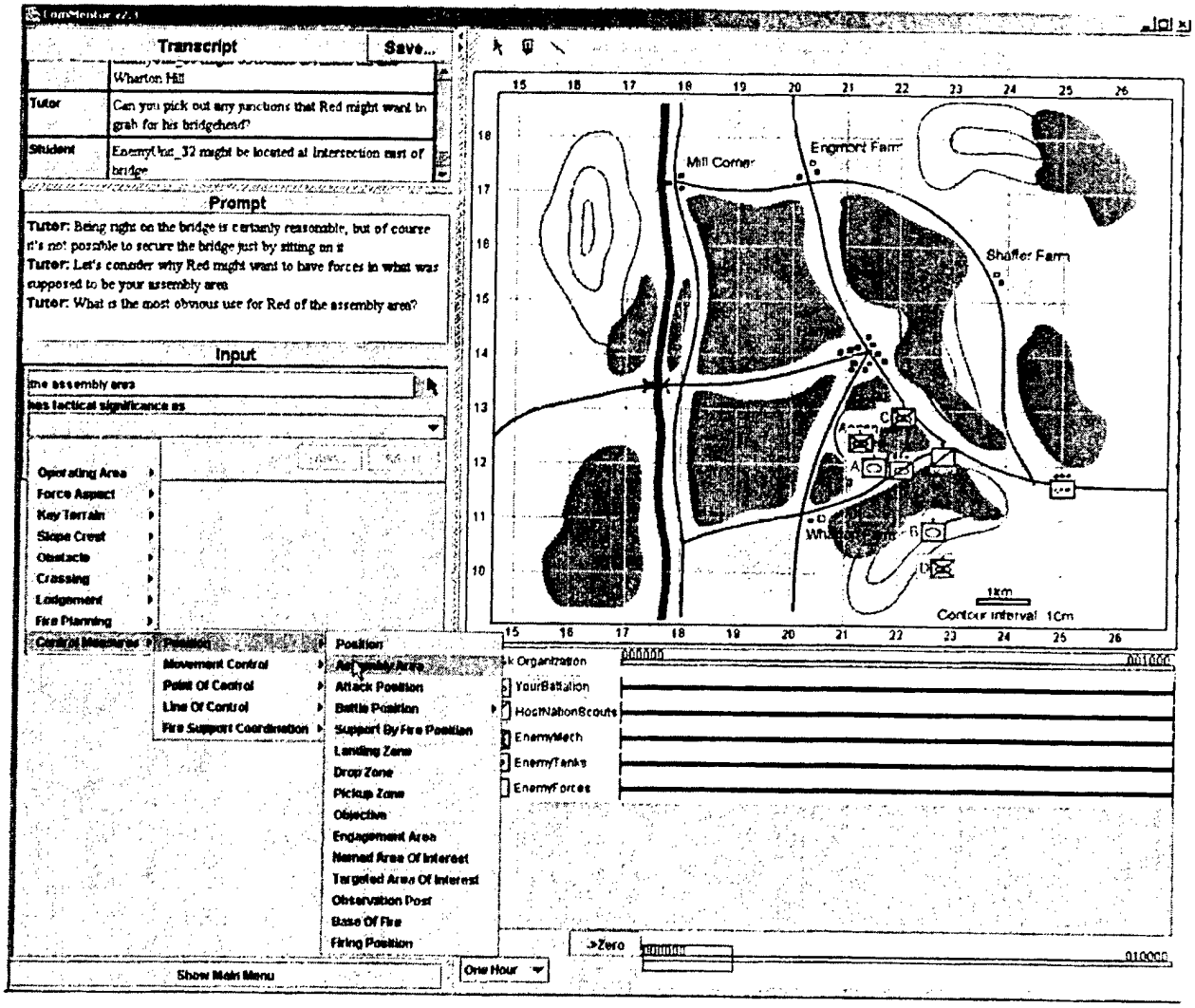


Figure 61. The Red Situation: The Student's Answer to the 5th Question.

The more interesting question is what the assembly area might help Red to do. Figure 62 shows the student making a rather complicated response to a rather simple question. This is an example of where the system should probably include a more tailored input form for answering questions about positions controlling other positions.

Tutor Yes. Red's most obvious use for the assembly area is to use it just as we intended--as an assembly area.

Tutor What areas does Red's presence in the assembly area help them control?

Student EnemyUnit_33 being at the assembly area TacticallyEnables Cover by Enemy Mech or (location) intersection at choke point near assembly area.

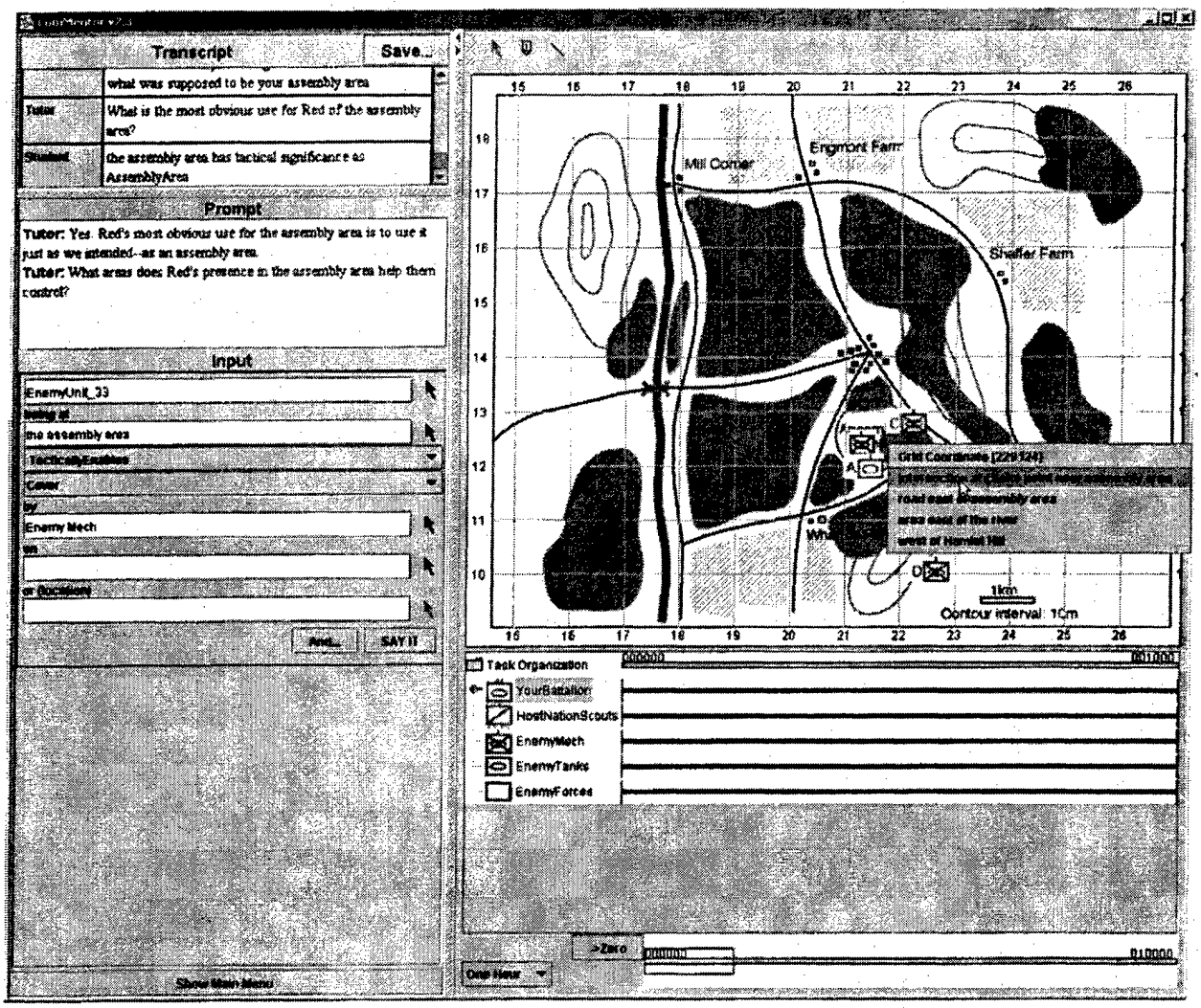


Figure 62. The Red Situation: The Student's Answer to the 6th Question.

The tutor summarizes the points made about the assembly area, then discussion moves on to some of the other sites suggested by the student. Figure 63 shows the student answering a question about Hamlet Hill.

Tutor	Being in the assembly area allows red forces to control movements on the roads to the east and south of the area, and if they work their way into the woods at the southeast, they can even directly affect the junction just after the choke point.
Tutor	From Red's perspective, let's look at the pros and cons of having forces on Hamlet Hill.
Tutor	How would you assess the tactical significance of Hamlet Hill?
Student	Hamlet Hill has tactical significance as KeyTerrain.
Tutor	Yes. Given the fields of observation and fire from Hamlet Hill covering the whole bowl into Hamlet on the west and the larger territory out to the east, I'd say Red would consider it key terrain for their purposes.

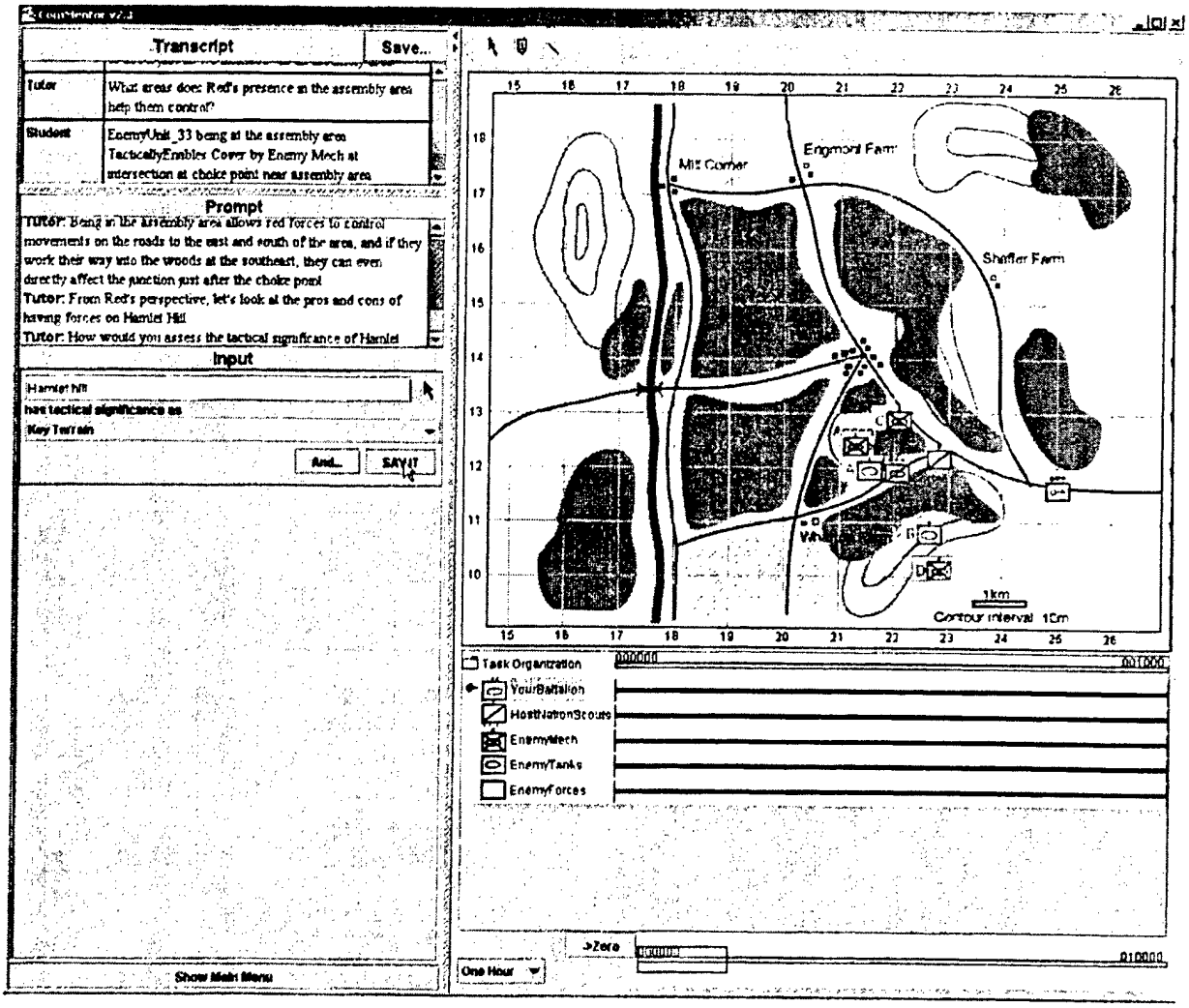


Figure 63. The Red Situation: The Student's Answer to the 7th Question.

A similar exchange disposes of Wharton Hill. In both cases, if the student does not recognize their significance, the tutor is ready to delve into Red's interest in these locations in greater depth. The tutor finishes up with another comment on the intersection at the east end of the bridge before summarizing the entire discussion about the possible layout of a bridgehead.

Tutor For Red, let's consider what's good or bad about having forces on Wharton Hill.

Tutor How would you assess the tactical significance of Wharton Hill?

Student Wharton Hill has tactical significance as KeyTerrain.

Tutor Given the fields of observation and fire from Wharton Hill covering the road just to the north and the territory to the west and south, Red would probably consider it key terrain for their bridgehead.

Tutor They are certainly going to be at the intersection at the east end of the bridge. They'll be flowing all their forces through there, and that junction gives them access to three alternate LOCs, so it's a key point for them.

Figure 64 shows the tutor's bridgehead layout summary. This presentation includes a map manipulation: the tutor flashes a red highlight over the entire bowl centered on Hamlet to reinforce the idea that Red will likely be looking to solidify its hold on a sizable piece of terrain.

Tutor Overall, you would expect the enemy to try to secure a region something like that highlighted on the map.

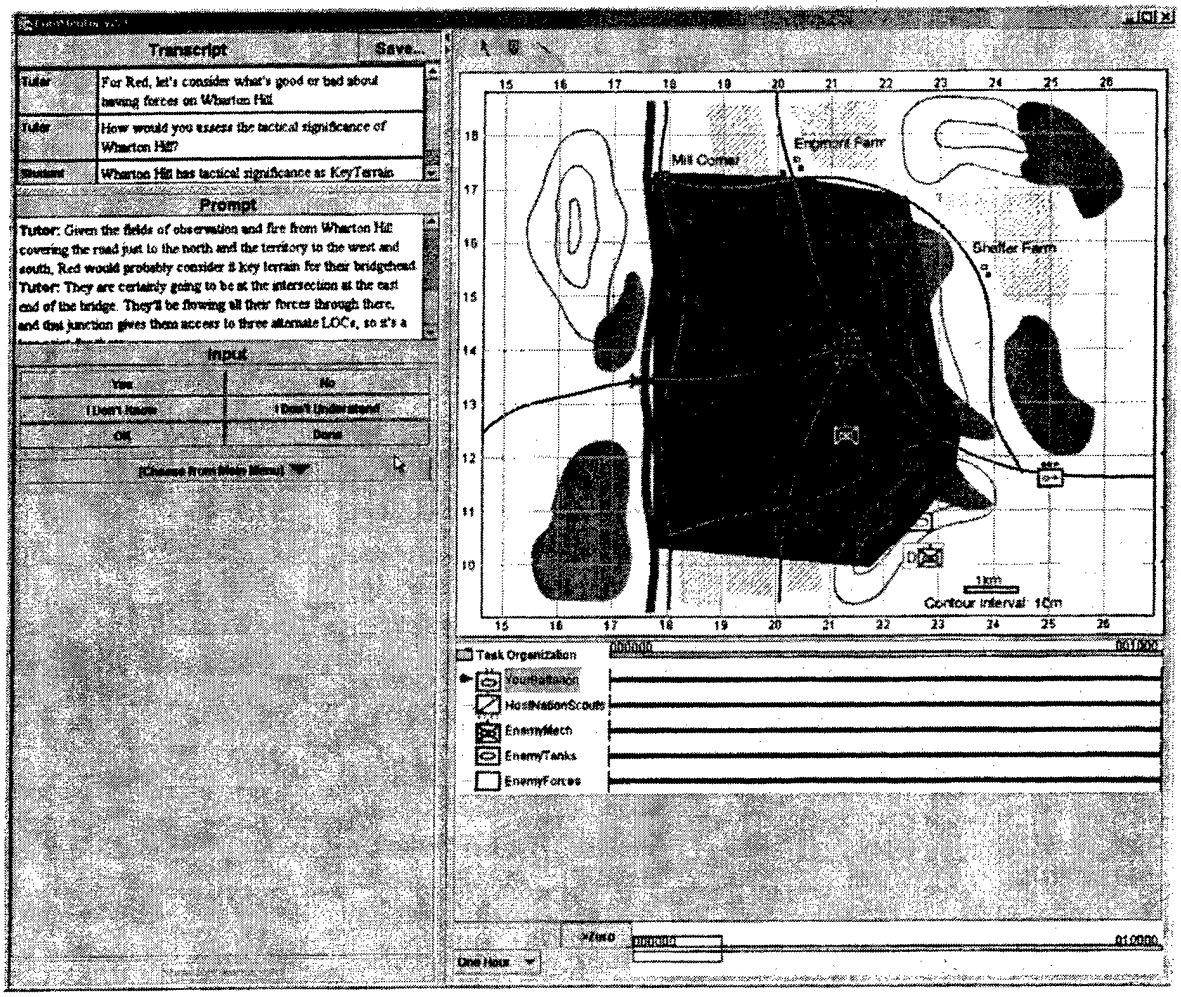


Figure 64. The Red Situation: Tutor Highlights Red Bridgehead Area.

The tutor has now concluded its discussion of the Red situation. Its next question is addressed more specifically at the student's proposed course of action: the attack on the assembly area. The tutor asks about risk factors, primes the Input pane with a form that can answer such a question. Figure 65 shows the student choosing their answer—"high uncertainty"—from among ComMentor's currently defined set of risk factors. While this is not a bad answer, it is not the one the tutor is looking for. The tutor asks a follow-up question, but we assume that the student cannot figure out what the tutor is getting at, and responds with "I Don't Know"—an option that is generally available along with the Main Menu (which the student can access by clicking "Show Main Menu" at the bottom of the Input pane).

Tutor	What sort of risks do you see in your move against the assembly area?
Student	The major risk factor for this operation is HighUncertainty affecting the task force.
Tutor	Can you characterize any risks posed by your use of these particular forces to move against the assembly area?
Student	I Don't Know.

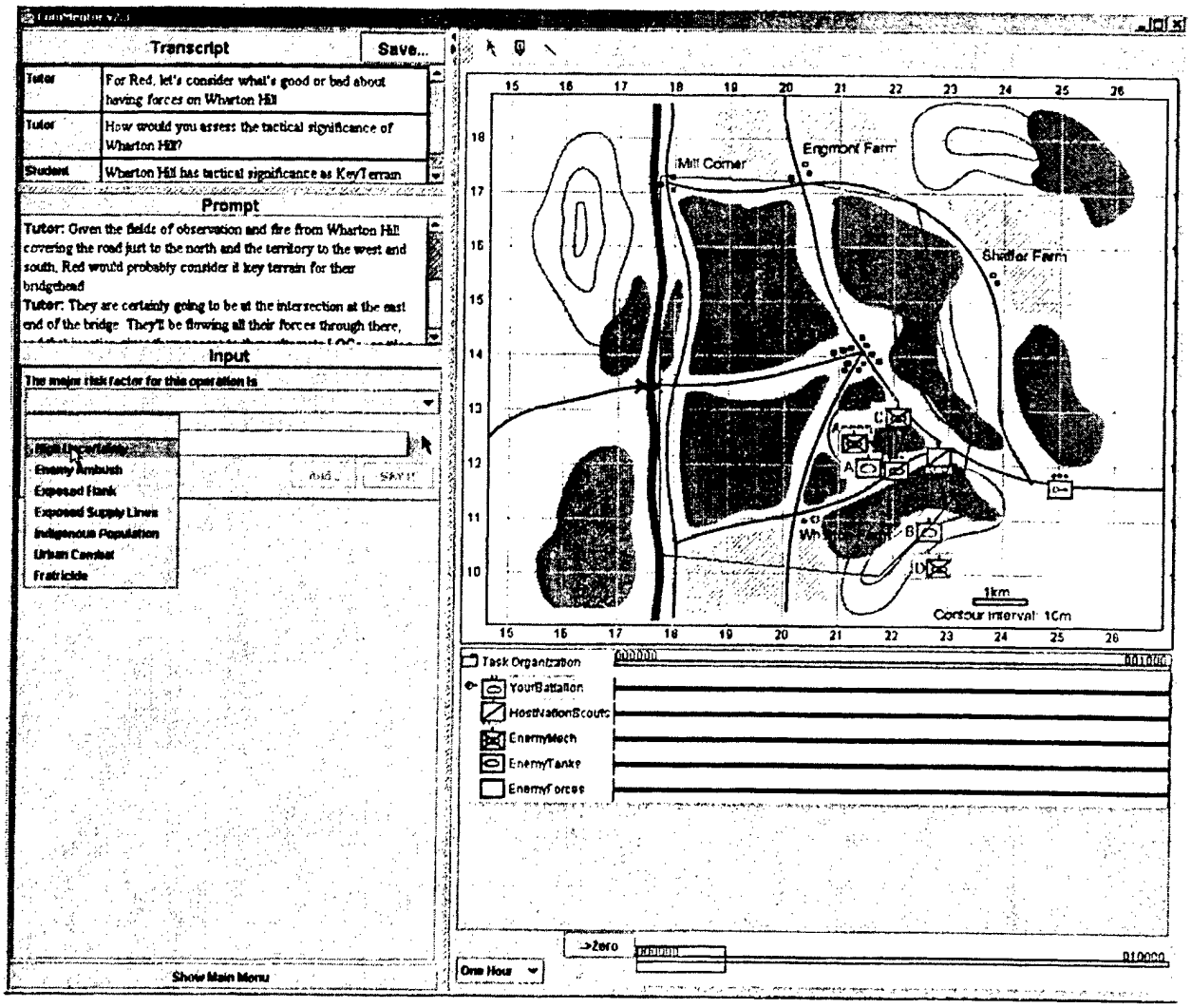


Figure 65. Attack on AA: The Student's Answer to Main Question.

In response to the student's "I Don't Know" answer the tutor starts to drill down into an argument for the particular risk it has in mind—the risk of fratricide. Figure 66 shows the 1st drill-down question and the complex nested form the system recommends. This is a form that allows description of terrain conditions including ground, atmospheric, and illumination conditions. The student might understandably be overwhelmed or unsure of what to say, and so here, we show the student again saying "I Don't Know."

Tutor What are the prevailing conditions as this scenario opens?
Student I Don't Know.

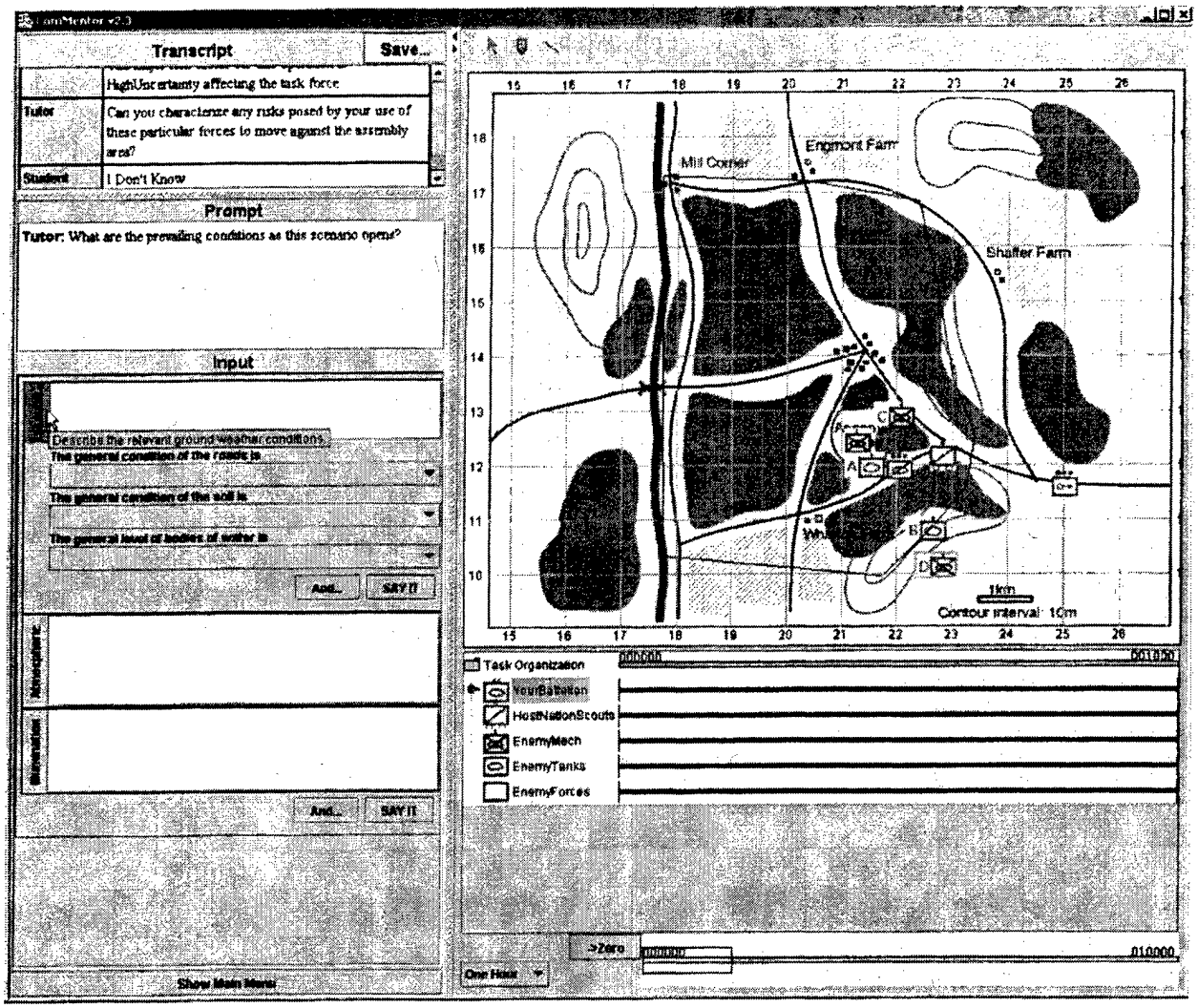


Figure 66. Attack on AA: The Tutor's 1st Drill-Down Question.

When the student clicks "I Don't Know" in response to the general drill-down question, the tutor comes back with a more focused follow-up question (and primes a more specific form—the Illumination Conditions form that originally appeared nested inside the general Terrain Conditions form). Figure 67 shows the student providing an answer to that more specific question. This is a place where the current input facilities are likely to prove quite unnatural, as a student would probably want to say something like the current illumination is "low" or "dim" or any of a number of other non-numeric descriptions.

Tutor	What are the illumination conditions?
Student	The percent illumination is 10.0.
Tutor	Right. It's important to remember is that this whole encounter is happening around 0100, so it's pretty dark out there.

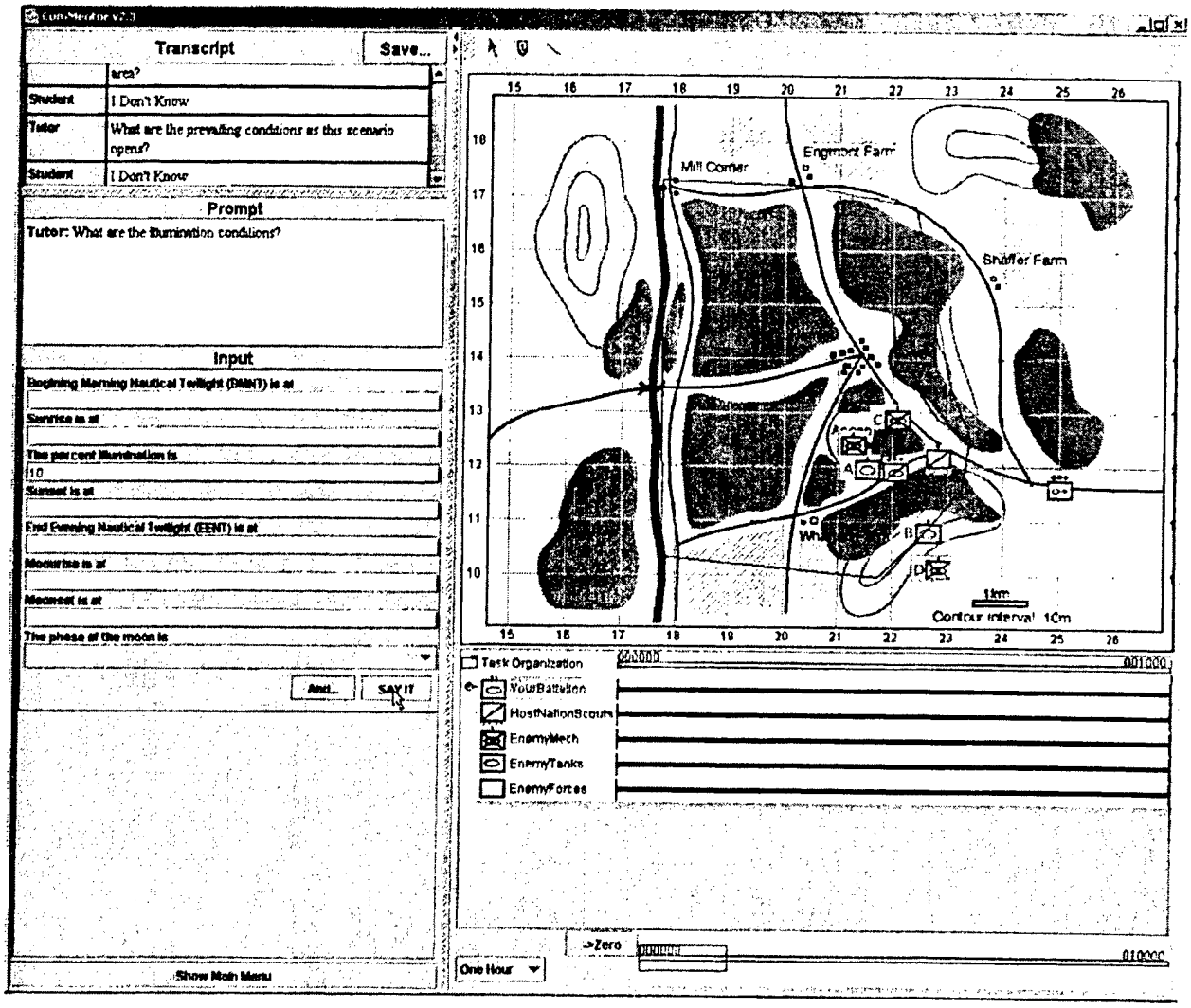


Figure 67. Attack on AA: The Student Response to 1st Drill-Down Question Follow-Up.

Figure 68 shows the student in the middle of answering the 2nd drill-down question in the tutor's argument for a risk of fratricide—a question about the size of the assembly area.

Tutor About how big is the assembly area?
Student The size of the assembly area is 1.5 SquareKilometer.
Tutor The whole assembly area is probably just about one square Km.

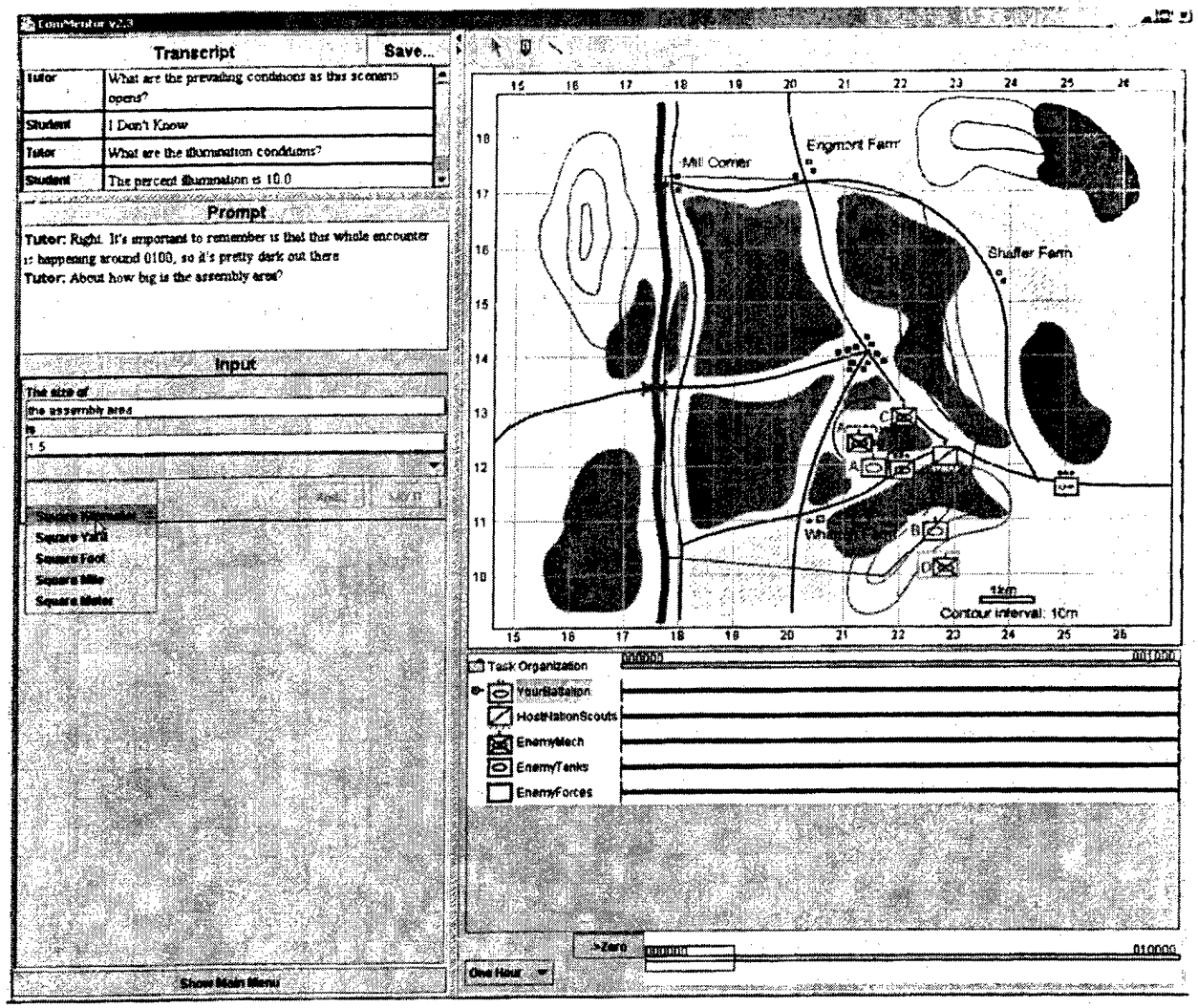


Figure 68. Attack on AA: The Student Responding to 2nd Drill-Down Question.

Figure 69 shows the tutor's response after answering the 3rd and final drill-down question—a question about the size of the assembly area (with an eye towards relating the size of the target area to the range of the student's tanks' main guns. The tutor takes the set of points just established and builds them into a final yes/no question that attempts to get closure on the risk of fratricide. At this point, the most likely response is for the student to click "Yes." In that case, they get the summary shown in the Prompt pane in Figure 70.

Tutor	What is a typical range for the main gun of a modern tank?
Student	A M1A1 has a range of 3000.0 Meter.
Tutor	Right. Your tanks can easily shoot far enough that when they open up on the assembly area, there are likely to be shots flying right through and out the other side.
Tutor	Does a night fight in close quarters give you any qualms about the possibility of fratricide?
Student	Yes.

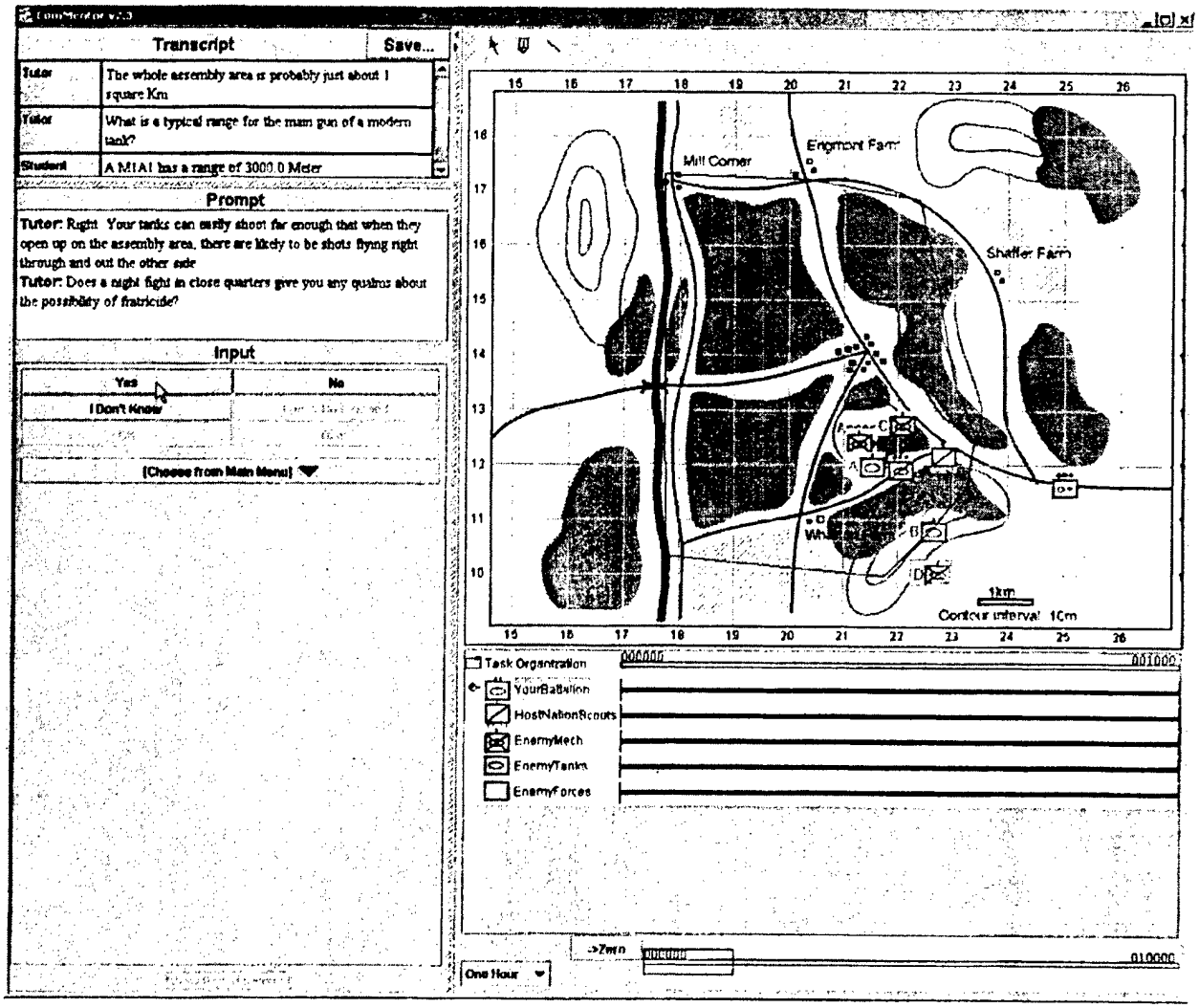


Figure 69. Attack on AA: The Tutor's Query to Close out Drill-Down on Main Question.

Tutor Good call. The fact that you're going to be fighting in the dark, in an area that's small compared to the range of your tanks' main guns means that you need to give extra consideration to the danger of fratricide if you're going to be maneuvering forces in from more than one side.[22]

Having raised the point about the possibility of fratricide between the two companies attacking the assembly area, the tutor now shifts to a related topic: the possibility of fratricide between the oncoming combat unit and the scouts. Figure 70 shows the student responding to the tutor's query about the location of the scout platoon.

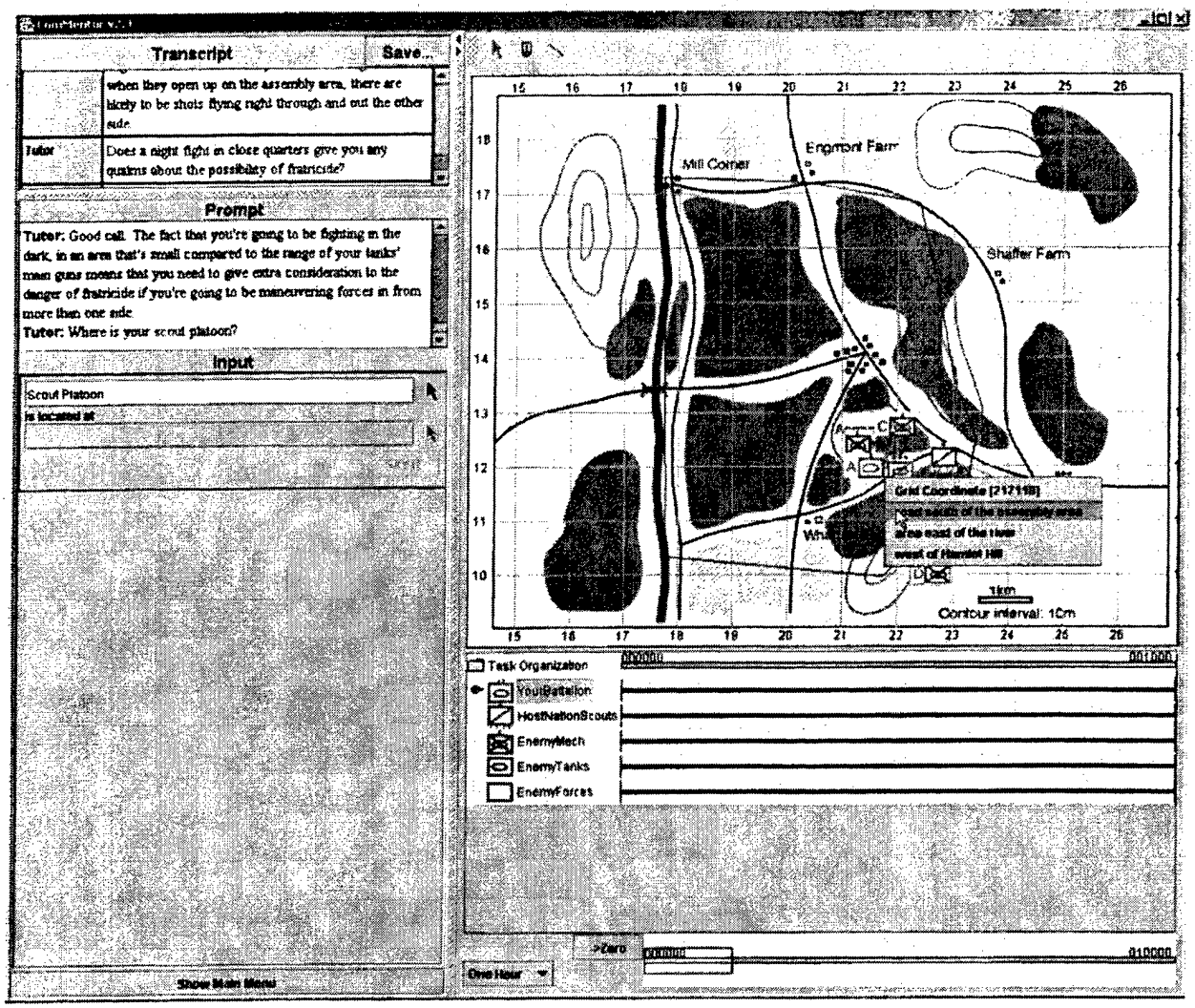


Figure 70. Scouts Tasking: The Student Responding to 1st Question.

[22] Note that the tutoring for this scenario, as currently authored, is not as sensitive to the geometry of the situation as it ought to be. Further, in live tutoring sessions we observed, when this line of discussion came up (or a similar one bearing on the positioning of the scouts, as below), the tutors often went on to talk about the general issue of coordinating multiple units working together in the same area, when a higher commander should get involved in such coordination, and specific ideas about possible control measures and sequencing to avoid problems in this particular case.

Tutor Where is your scout platoon?

Student Scout Platoon is located at road south of the assembly area.

Tutor Right. Your scouts are sitting somewhere on the southern edge of the assembly area, along the road to Wharton Farm.

This line of discussion continues with the exchange below. Figure 71 shows the student responding to the next question on the way to highlighting the problems that may follow from leaving the scouts where they were at the start of the scenario.

Tutor What are your scouts doing right now?

Student Scout Platoon is Observe on Enemy unit vicinity of The Assembly Area.

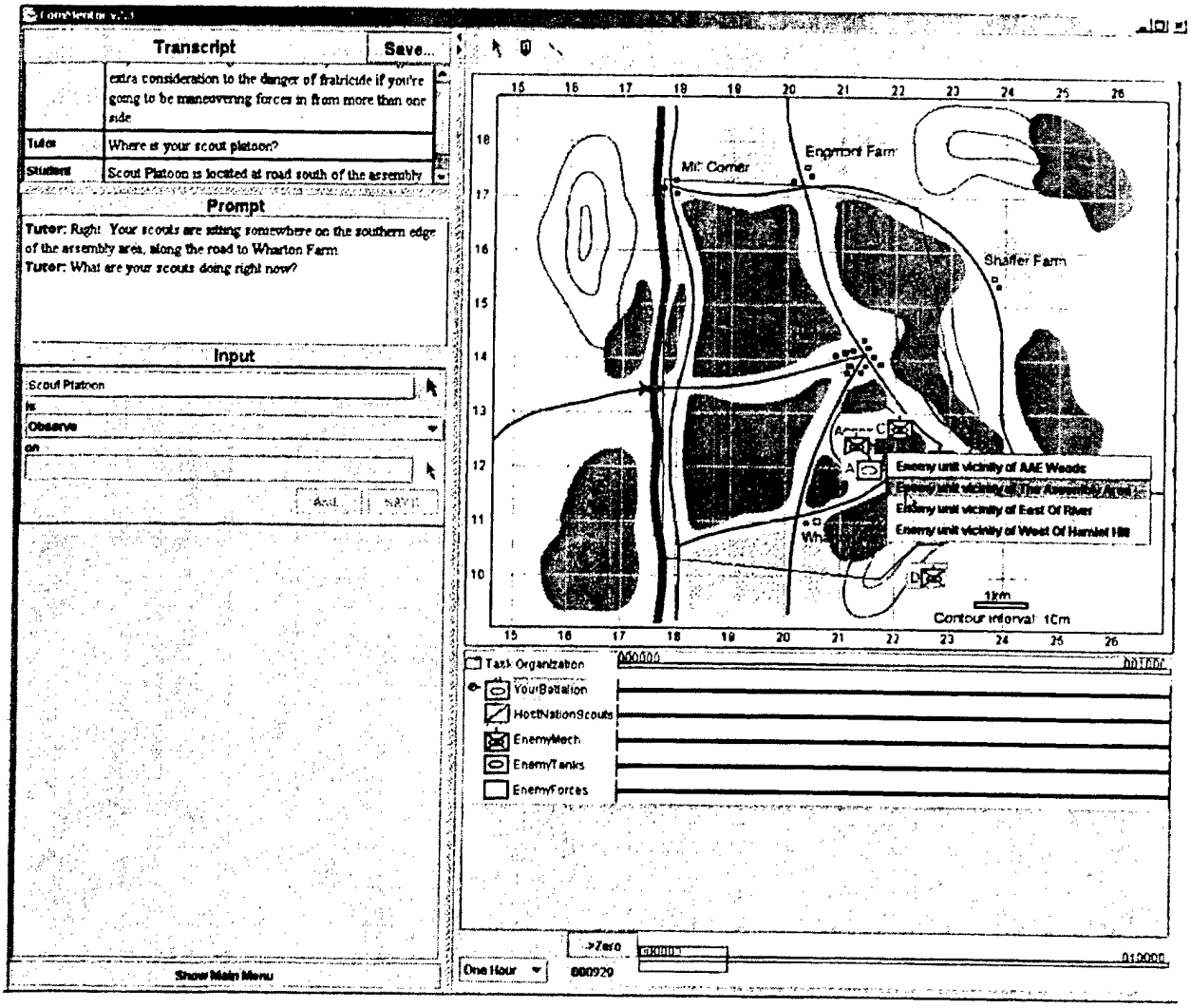


Figure 71. Scouts Tasking: The Student Responding to 2nd Question.

The student's answer is correct, but not the one the tutor was looking for. The important point, from the tutor's perspective is that the scout platoon is engaged with the enemy. The tutor

tries a follow-up question to elicit that observation. Figure 72 shows the student in the process of assembling that answer.

Tutor	What's going on between the scouts and the enemy forces in the assembly area?
Student	Scout Platoon is Engage on Enemy Mech.
Tutor	Right. Your scouts are engaged with the enemy forces in the assembly area.

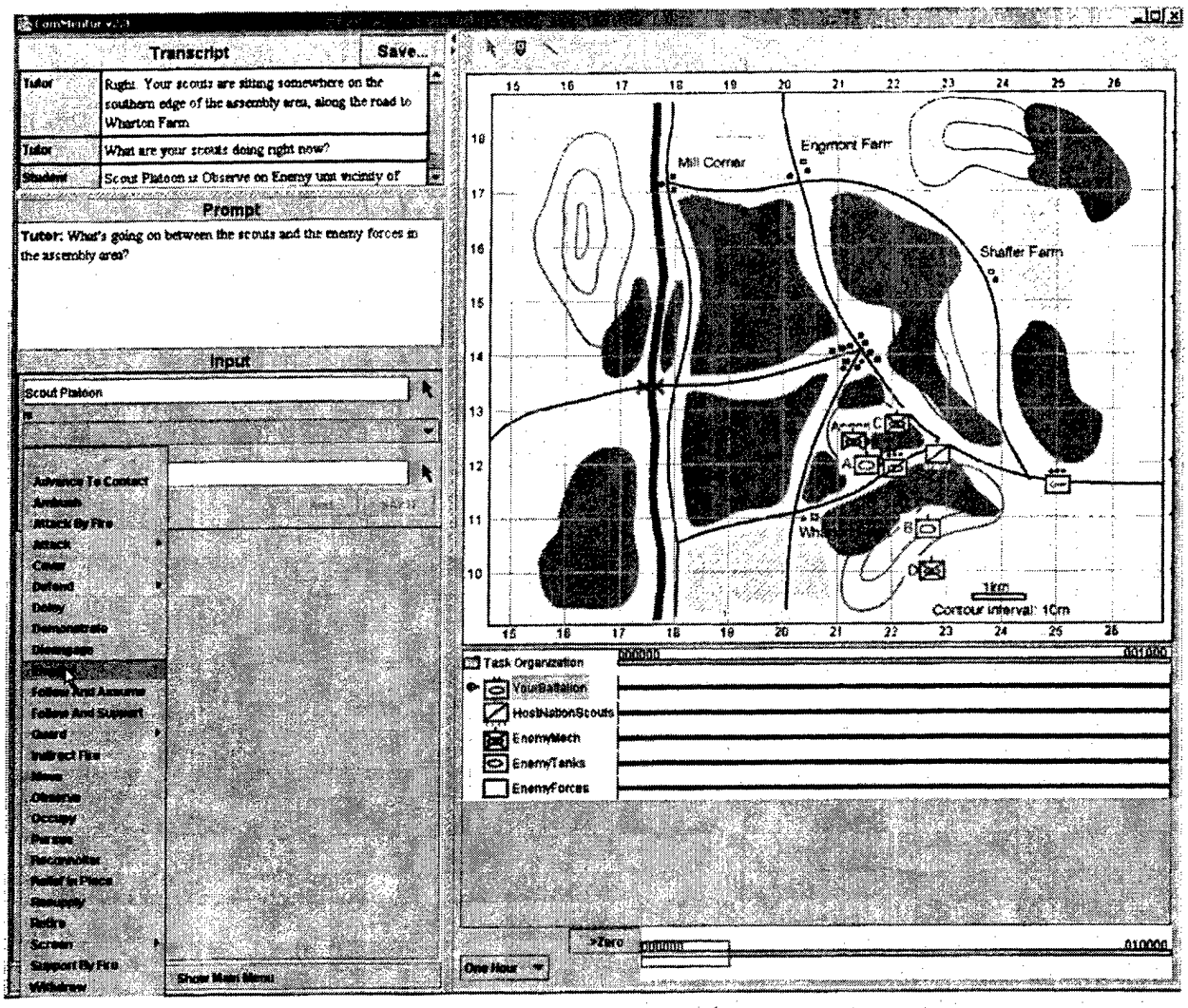


Figure 72. Scouts Tasking: The Student Responding to 2nd Question Follow-Up.

Figure 72 shows the student picking from ComMentor's list of military tasks. This list comes up frequently, especially in the context of issuing taskings. It is primarily a flat list with nested choices in only four places: Attack, Defend, Guard, and Screen. The task list was reviewed (and significantly simplified) by one of our SMEs, so we are reasonably confident the menu covers the vast majority of activities in these scenarios, and yet remains manageable so that it can quickly be mastered by new students.

Now the tutor comes to the crux of the current discussion topic: what is going to happen to the scouts when the assault on the assembly area starts (when Alpha-Company comes blazing down the road to get to the assembly area)? As shown in Figure 73 the student realizes that the scouts are likely to suffer losses from friendly fire.

Tutor What's going to happen to the Scouts when your assault on the AA starts?
Student Alpha Company is Attrit to Scout Platoon.

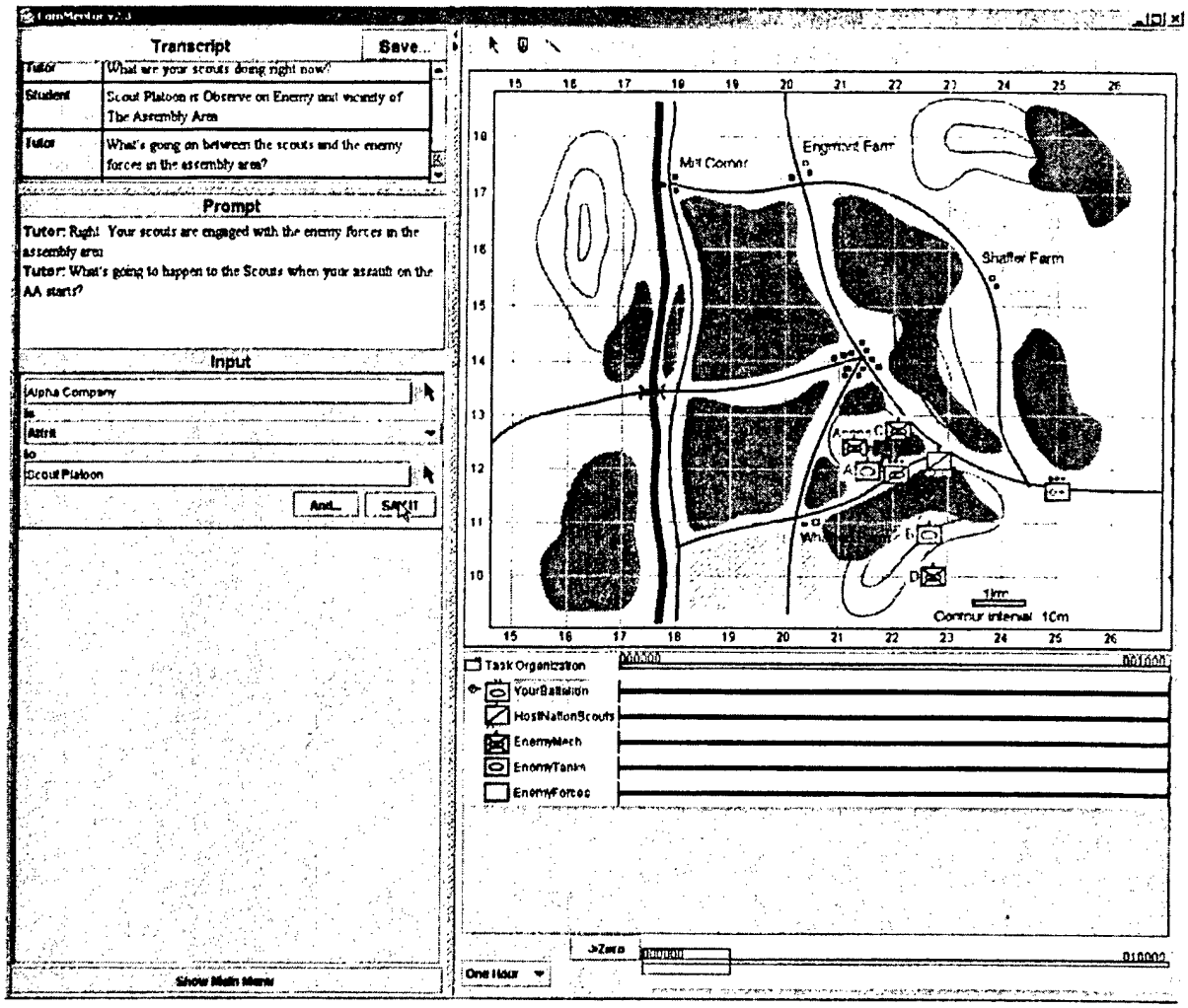


Figure 73. Scouts Tasking: The Student's Response to 3rd Question.

One of the interesting things about this discussion is that it culminates in a request for the student to issue a new tasking. In live tutoring sessions this was a common occurrence. One of the points of discussing likely consequences of the student's plan is to get the student to see the situation in a new way, and to consider how that enhanced view of the situation might lead to better courses of action. Figure 74 shows the student assembling the new tasking for the scouts. We note that in the live sessions, the tutor typically also encouraged the student to issue a tasking to the host-nation forces, generally one that had them linking up with the scouts and moving together.

Tutor	Yes. Given their current location, your scouts are likely to get shot up by your own assault force.
Tutor	Why don't you give the Scouts some orders about what they should be doing--especially where they should go.
Student	Who: Scout Platoon Where: Wharton Hill Route: road south of the assembly area, Wharton forest, and Wharton Hill

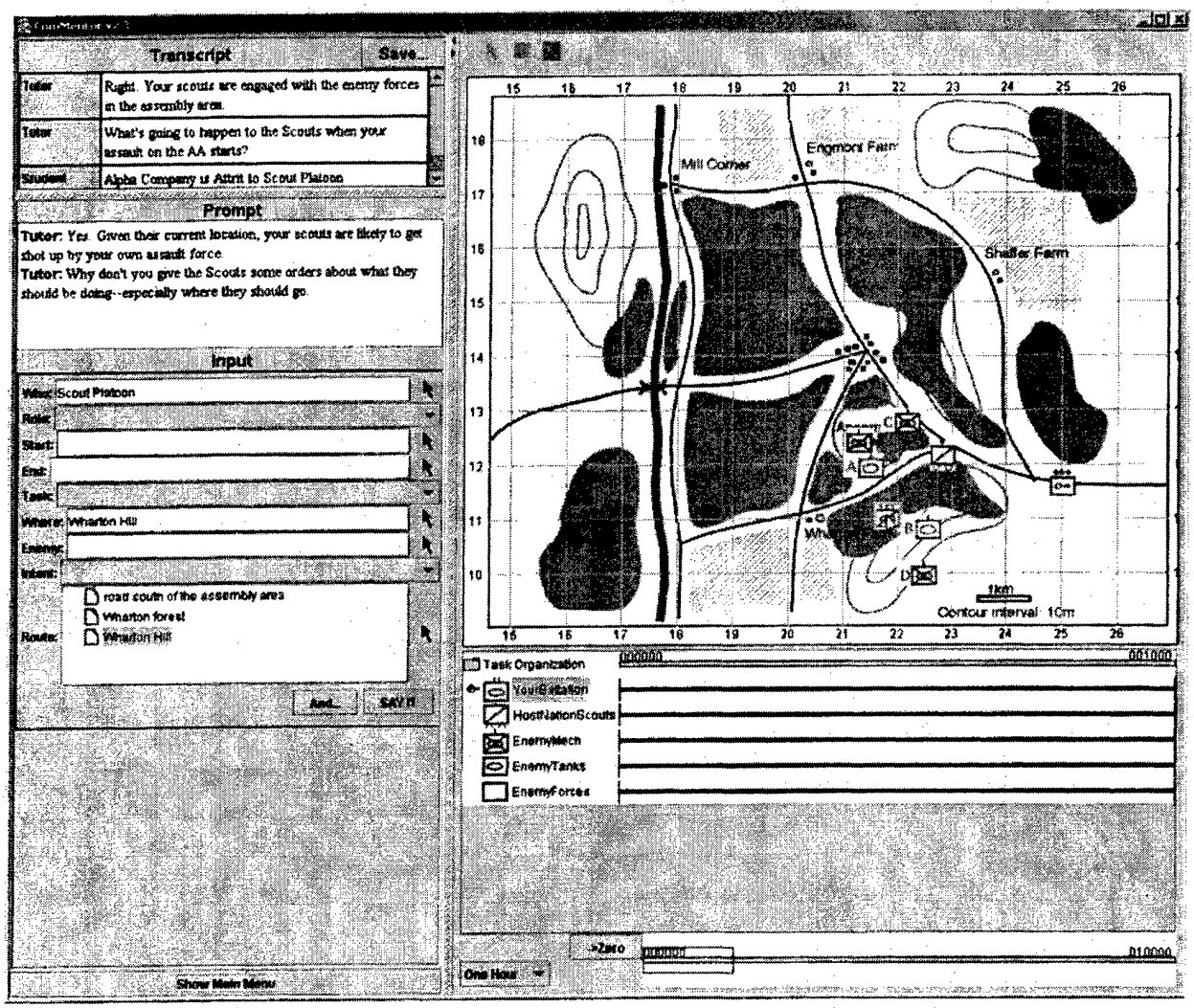


Figure 74. Scouts Tasking: The Student Moves the Scouts.

The new tasking has the scouts moving to the south. This decision too becomes the basis for additional discussion. The questions are: what is actually accomplished by moving the scouts south into Wharton Hill forest, and what else might have been done with them? Figure 75 shows the student describing why they moved the scouts south (following on the earlier discussion of what might happen if they stayed in place).

Tutor What is the primary value of moving your scouts to the south?

Student Scout Platoon being at Wharton Hill TacticallyDisables Engage by Alpha Company on Scout Platoon.

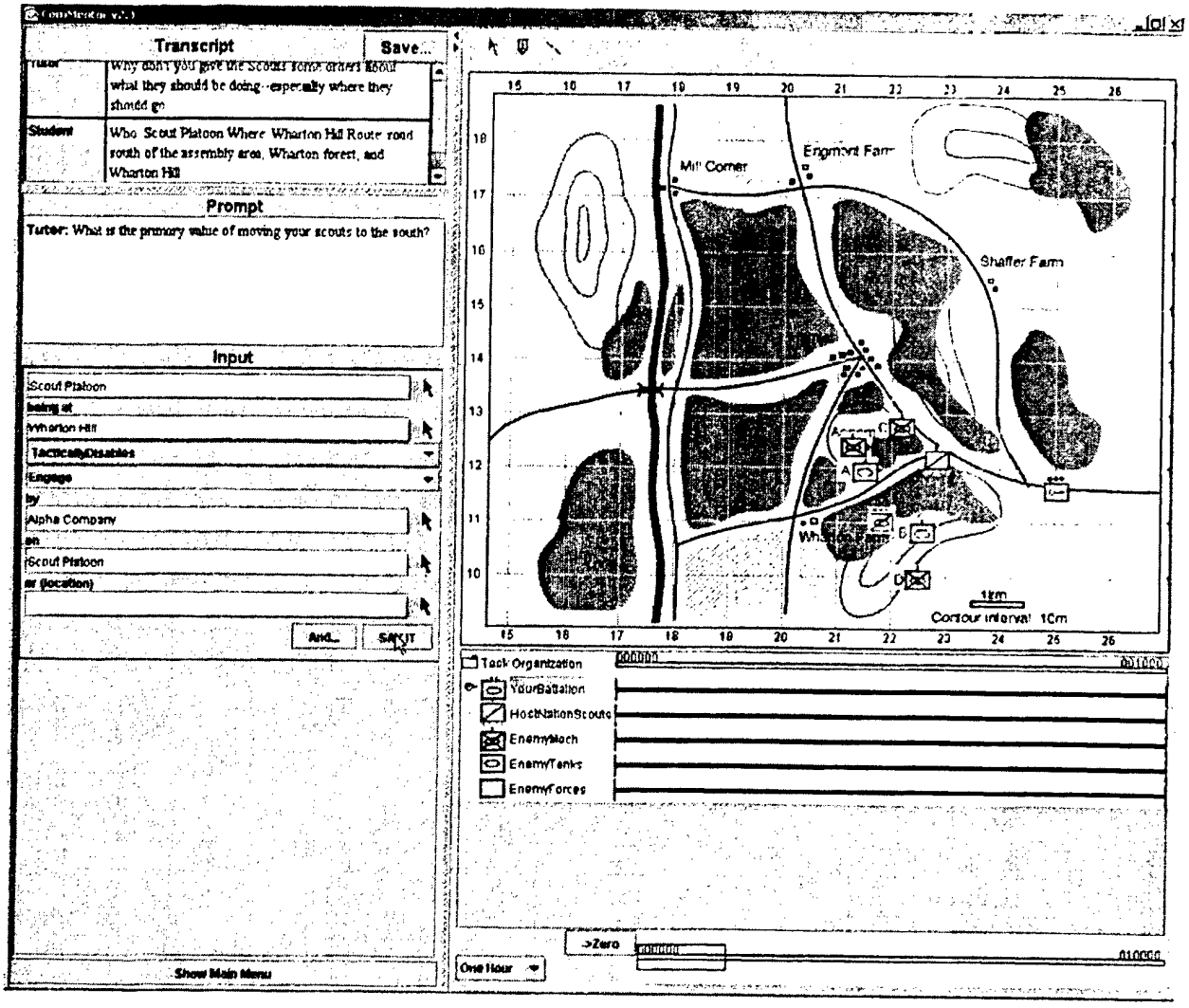


Figure 75. Scouts Moving to South: The Student's Answer to the 1st Question.

The tutor grants that the proposed move achieves the original goal, but follows with a question that poses another possible goal: where might the scouts be positioned to gain additional useful intelligence? The student does not immediately see a better alternative, and so in Figure 76 we see the initial response is "I Don't Know."

Tutor	Yes. Moving your scouts to the south has the nice effect of getting them out of the way of your oncoming forces.
Tutor	Where might the scouts go to develop more useful intelligence for you?
Student	I Don't Know.

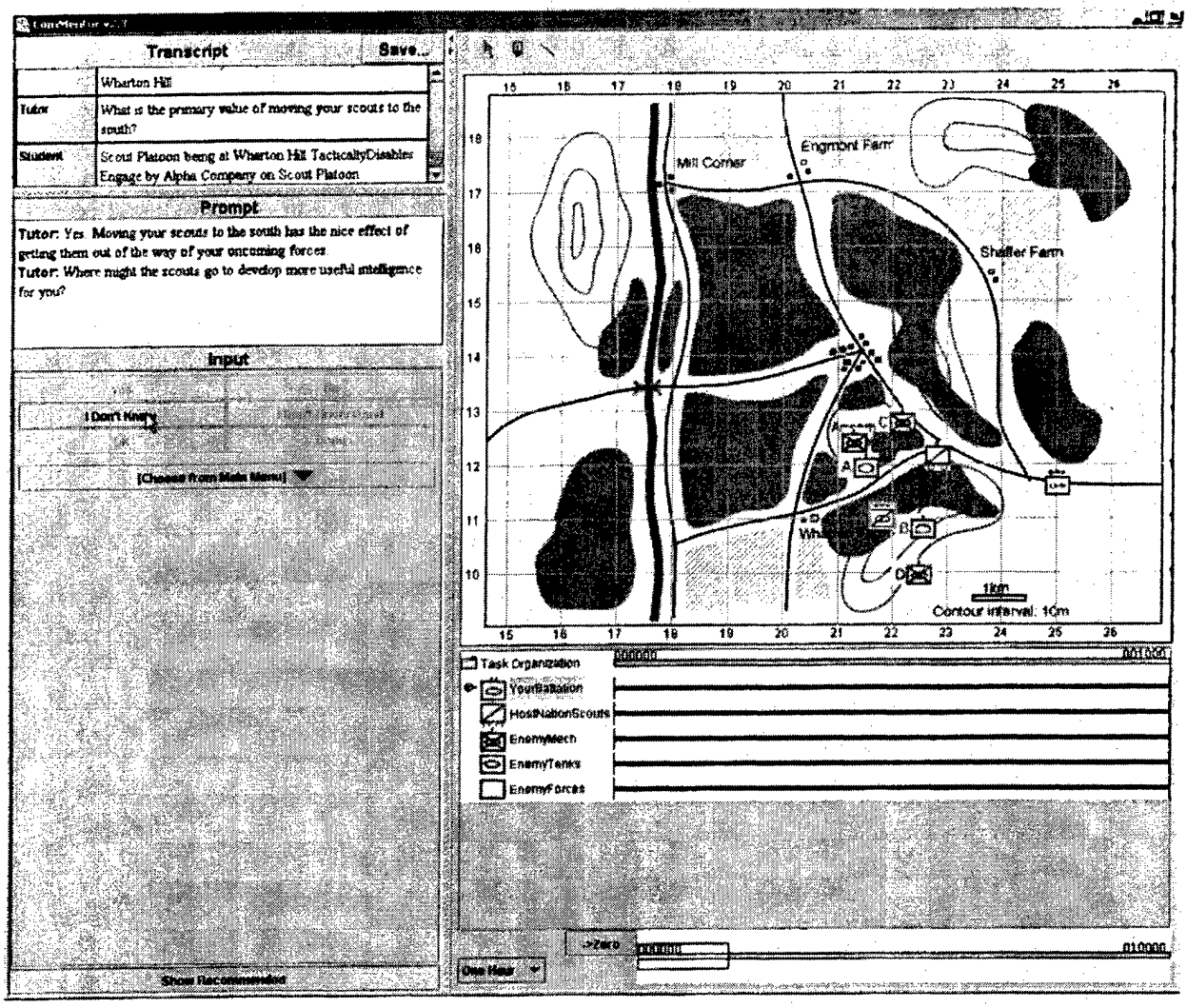


Figure 76. Scouts Moving to South: The Student's Answer to the 2nd Question.

The tutor has a follow-up question ready to suggest how the student might think about alternate positions for the scouts. Figure 77 shows the student assembling an answer to this follow-up question.

Tutor	You are coming from the east, and the enemy is certainly to the north. Where might the scouts develop new information?
Student	Scout Platoon being at Wharton intersection TacticallyEnables Observe by Scout Platoon or (location) intersection south of bridge.
Tutor	Yes. Moving your scouts to the west would also get them out of immediate harm's way and might help develop more situation awareness for you.

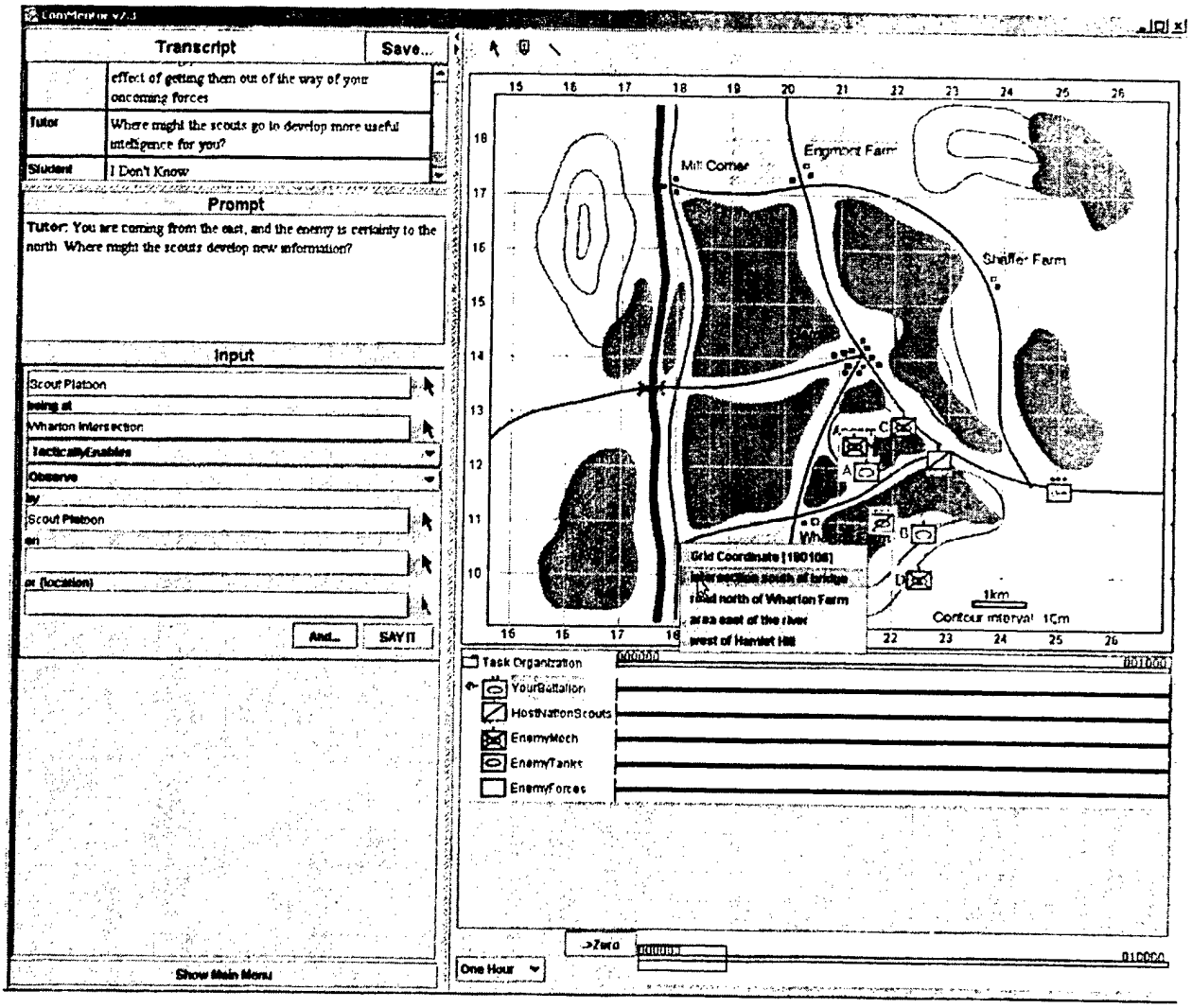


Figure 77. Scouts Moving to South: The Student Responds to the 2nd Question Follow-Up.

This time, the tutor does not solicit another update to the student's orders. Instead, it moves on to open a new discussion topic centered on one of the most important questions in the scenario: what about the bridge? Figure 78 shows the first exchange on this new topic. This is a case where the point is so important that simply having the student say "the bridge is key" is not sufficient. Instead of ending the discussion, that only starts the discussion.

Tutor What do you think about the tactical significance of the bridge?
Student The bridge has tactical significance as KeyTerrain.

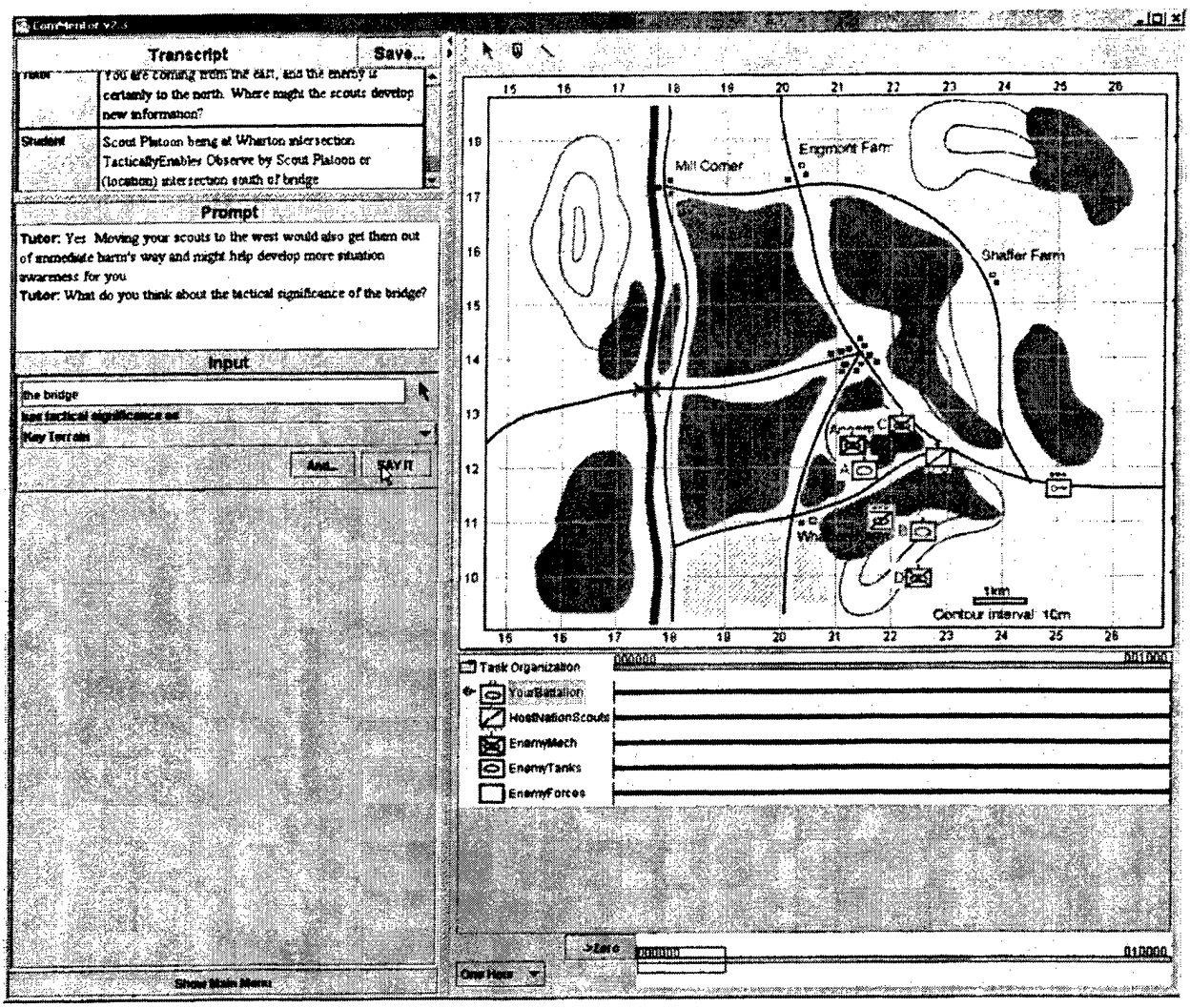


Figure 78. Ignoring the Bridge: The Student's Answer to the 1st Question.

The bridge is important both to Red and to Blue. The tutor starts by asking about why the bridge matters for Red. The student correctly answers that holding the bridge enables Red to stage its attack to the east.

Tutor	Why does the bridge matter for Red?
Student	EnemyUnit_53 being at the bridge TacticallyEnables Attack by the enemy forces or (location) area east of the river
Tutor	Right. Given the river is unfordable, and there don't appear to be any other bridges near here, Red needs the bridge if they are going to launch an offensive into our territory.

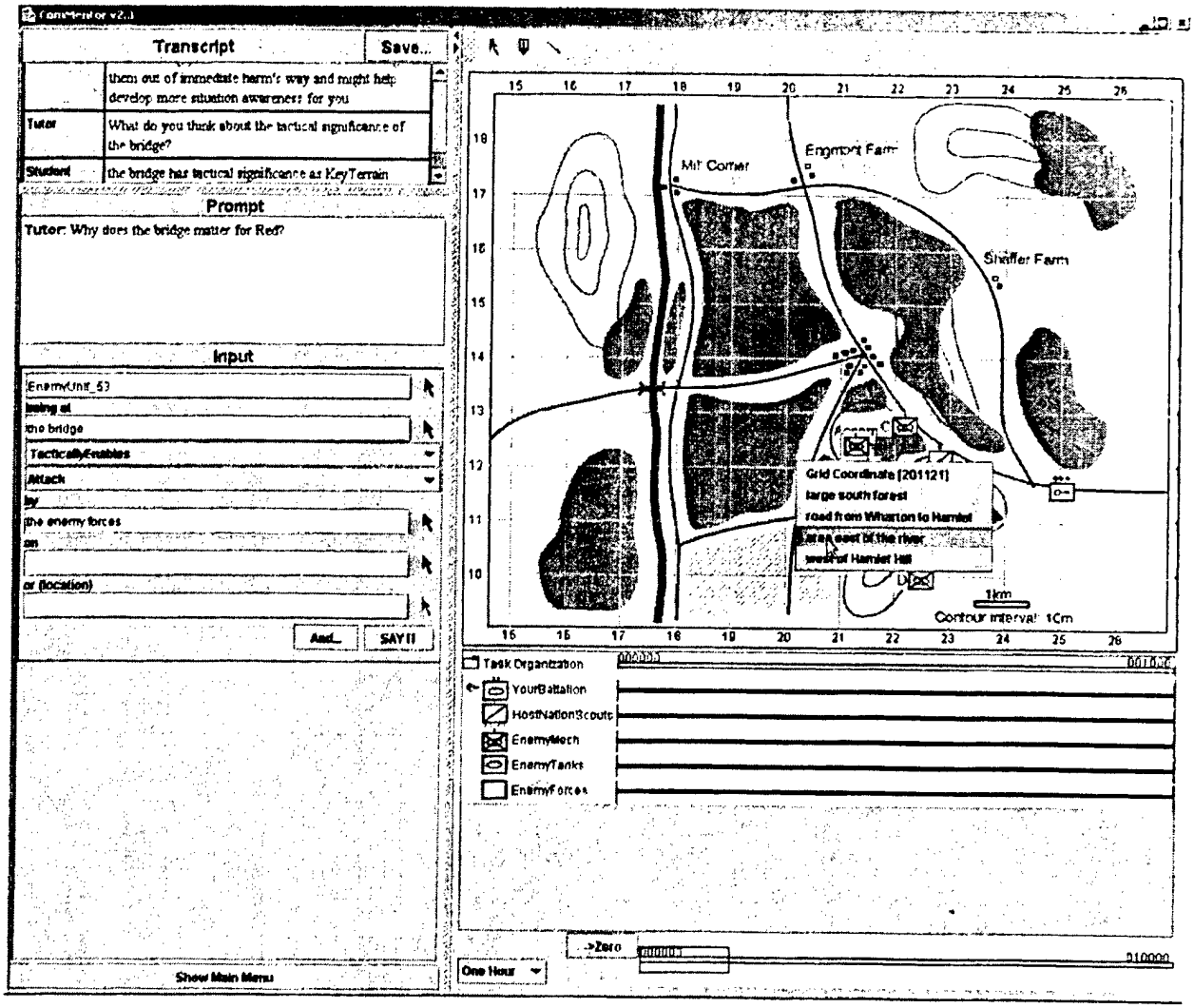


Figure 79. Ignoring the Bridge: The Student Responds to the 2nd Question.

The next question is intended to get at the most essential fact of the entire scenario: so long as the bridge remains in Red's hands, the student will have extreme difficulty winning the fight because more Red forces will keep flowing into the area. At first the student does not understand what the tutor is getting at. In fact, the student gives the same answer to this question that he did to the previous one.

Tutor So long as Red holds the bridge, what is happening?

Student EnemyUnit_54 being at the bridge TacticallyEnables Attack by the enemy forces or (location) area east of the river.

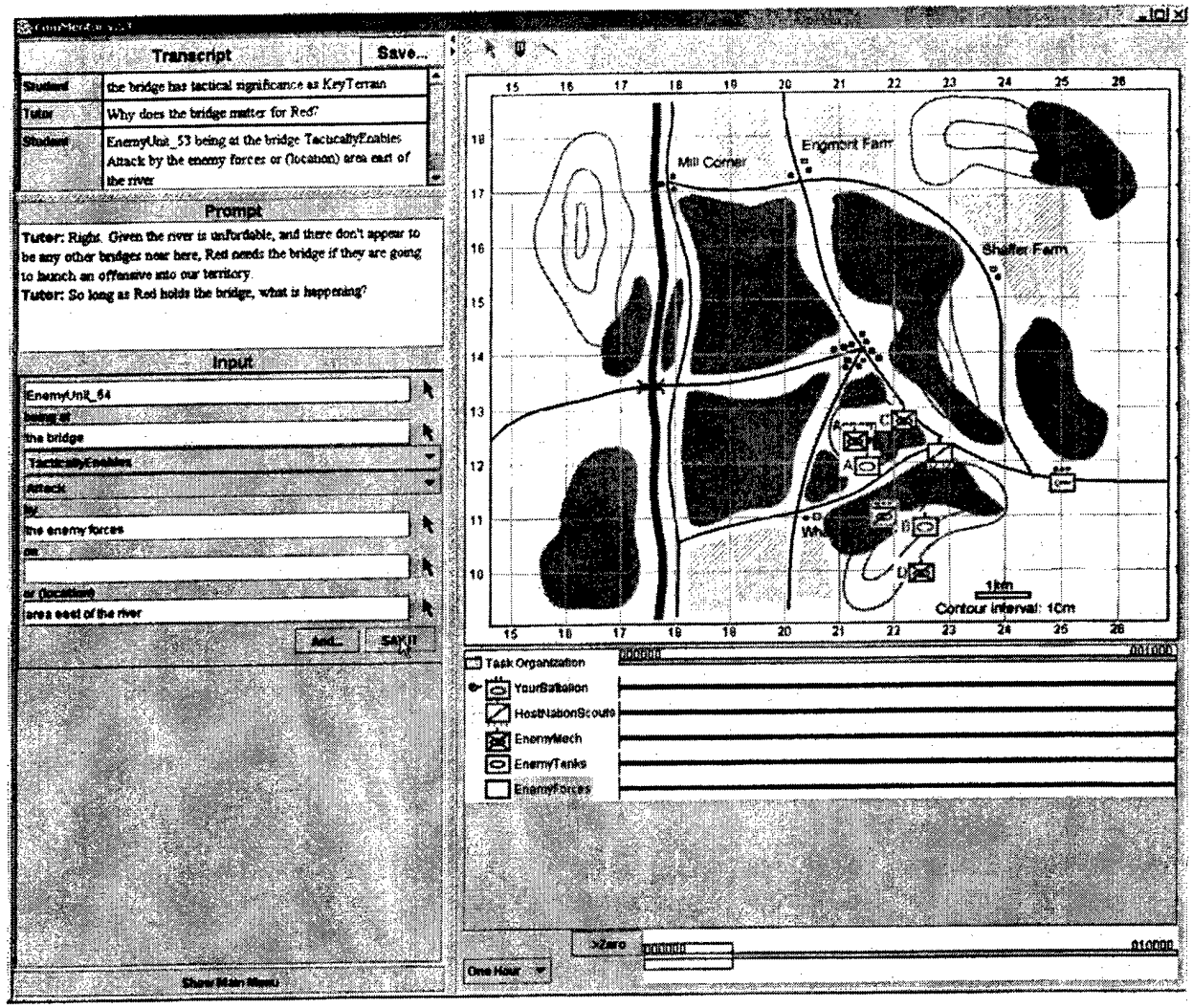


Figure 80. Ignoring the Bridge: The Student's Answer to the 3rd Question.

The tutor must follow-up with a much more leading question to get the student to see the point it is driving at. First the tutor acknowledges the accuracy of what the student has said. But then it explicitly directs the student to pay attention to the ongoing flow of Red forces into the contested area. Figure 81 shows the student acknowledging the validity of the tutor's point. The tutor summarizes and amplifies on the importance of the observation.

Tutor	Yeah, its what the bridge lets Red do in our territory east of the river that ultimately matters. But right now, would you agree that the most important thing it's letting them do is flow in more forces?
Student	Yes.
Tutor	Of course. So long as the bridge stays in Red's hands, it doesn't matter much how many red forces you kill, more are going to keep flowing into your area east of the river, and you'll just have to keep fighting them.

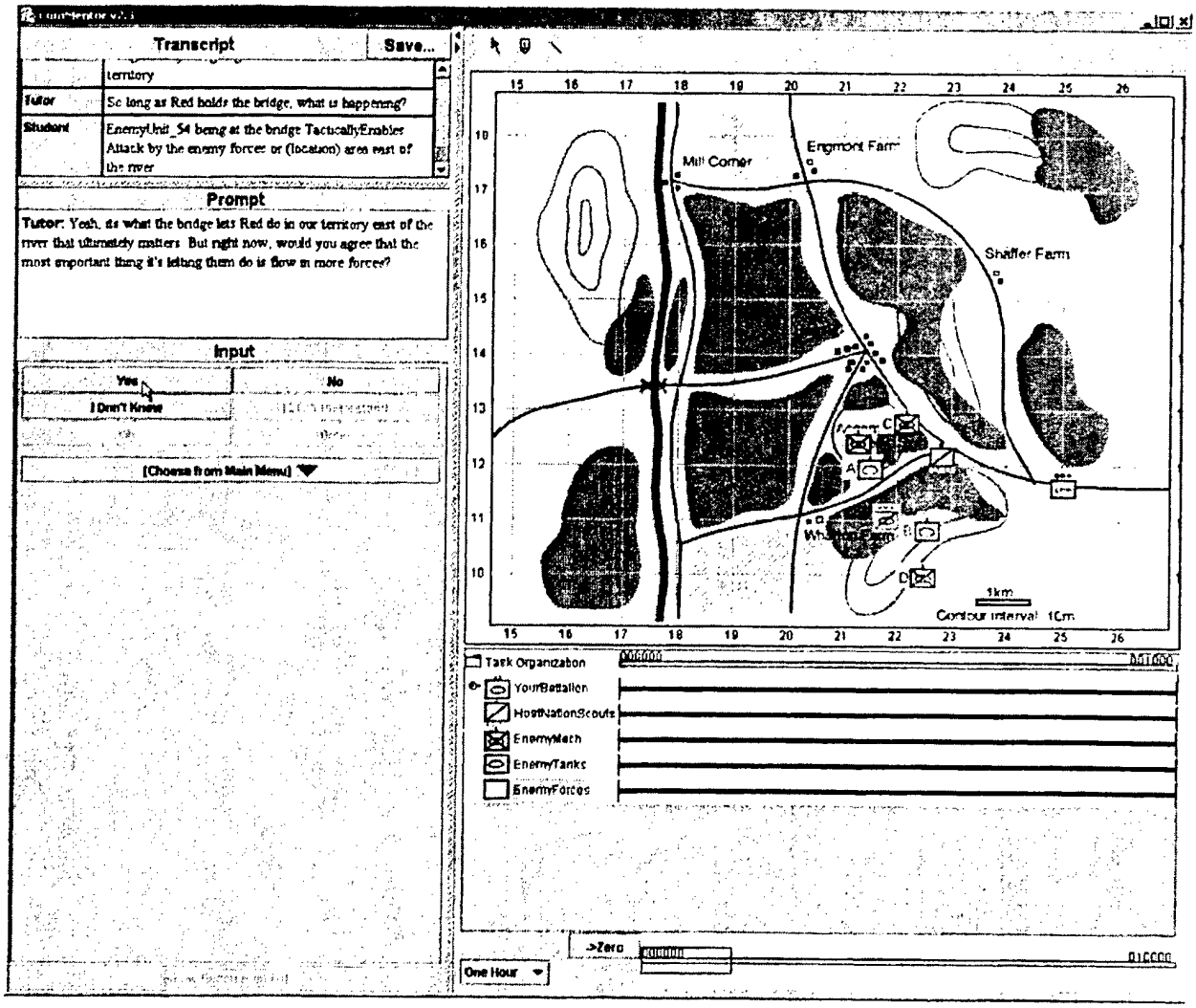


Figure 81. Ignoring the Bridge: The Student's Response to 3rd Question Follow-Up.

The tutor moves to the final point in its argument about the importance of the bridge: its relevance to the student's stated mission. Figure 82 shows the student correctly stating that the bridge is needed to enable the attack that Blue is supposed to stage to the west.

Tutor	In the longer term, why else does the bridge matter for us?
Student	FriendlyUnit_1 being at the bridge TacticallyEnables Attack by the task force or (location) area west of the river.
Tutor	Correct. Just as Red needs the bridge to launch their offensive, so we need the bridge too to be in a position to carry out the intent of our order: to launch the brigade offensive to the west.

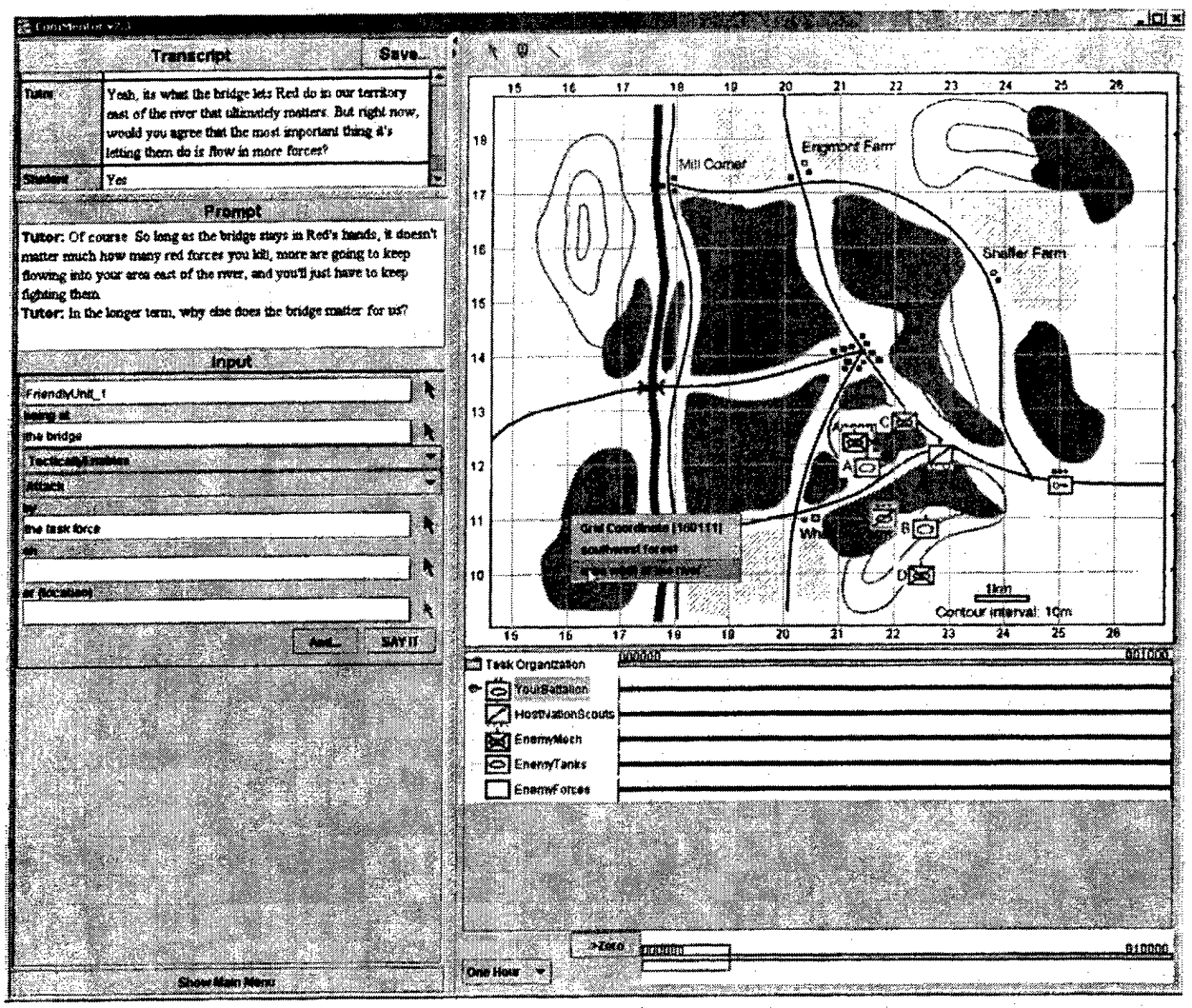


Figure 82. Ignoring the Bridge: The Student's Fourth Answer.

As with the argument that the scouts should be moved, the argument that the bridge should be taken culminates in the tutor prompting the student to enter a new tasking that addresses the newly identified goal. In Figure 83, the student tasks Bravo-Company to move down south of Wharton Hill, to the river, and up north to the bridge.

Tutor If the bridge is critical in this situation, can you think of a way to get some of your forces addressing that issue?

Student Who: Bravo Company Task: Occupy Where: the bridge Intent: Control Route: south of Wharton Hill, Wharton Farm, road through Wharton Farm, intersection south of bridge, river road south of bridge, Intersection east of bridge, and the bridge.

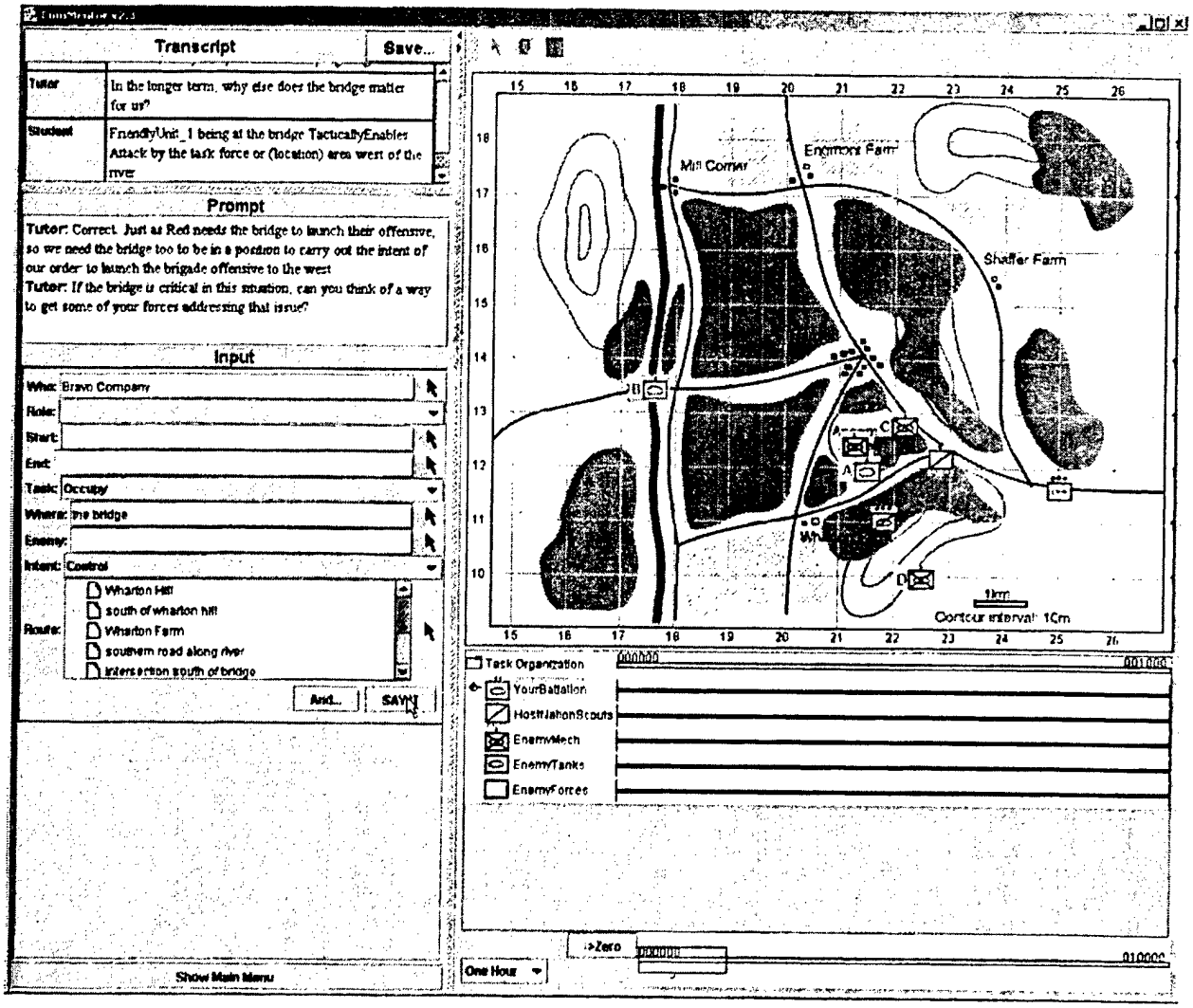


Figure 83. Ignoring the Bridge No Longer: A New Tasking.

At this point, the tutor moves on to a discussion of the chosen approach to the bridge. In typical fashion, the tutor asks the student to give the reasons they chose the route to the south of Wharton Hill. Figure 84 shows the student's answer.

Tutor What are the advantages of moving on the bridge by a route that takes you south of Wharton Hill?

Student South of Wharton Hill has IsScreenedFrom to the assembly area.

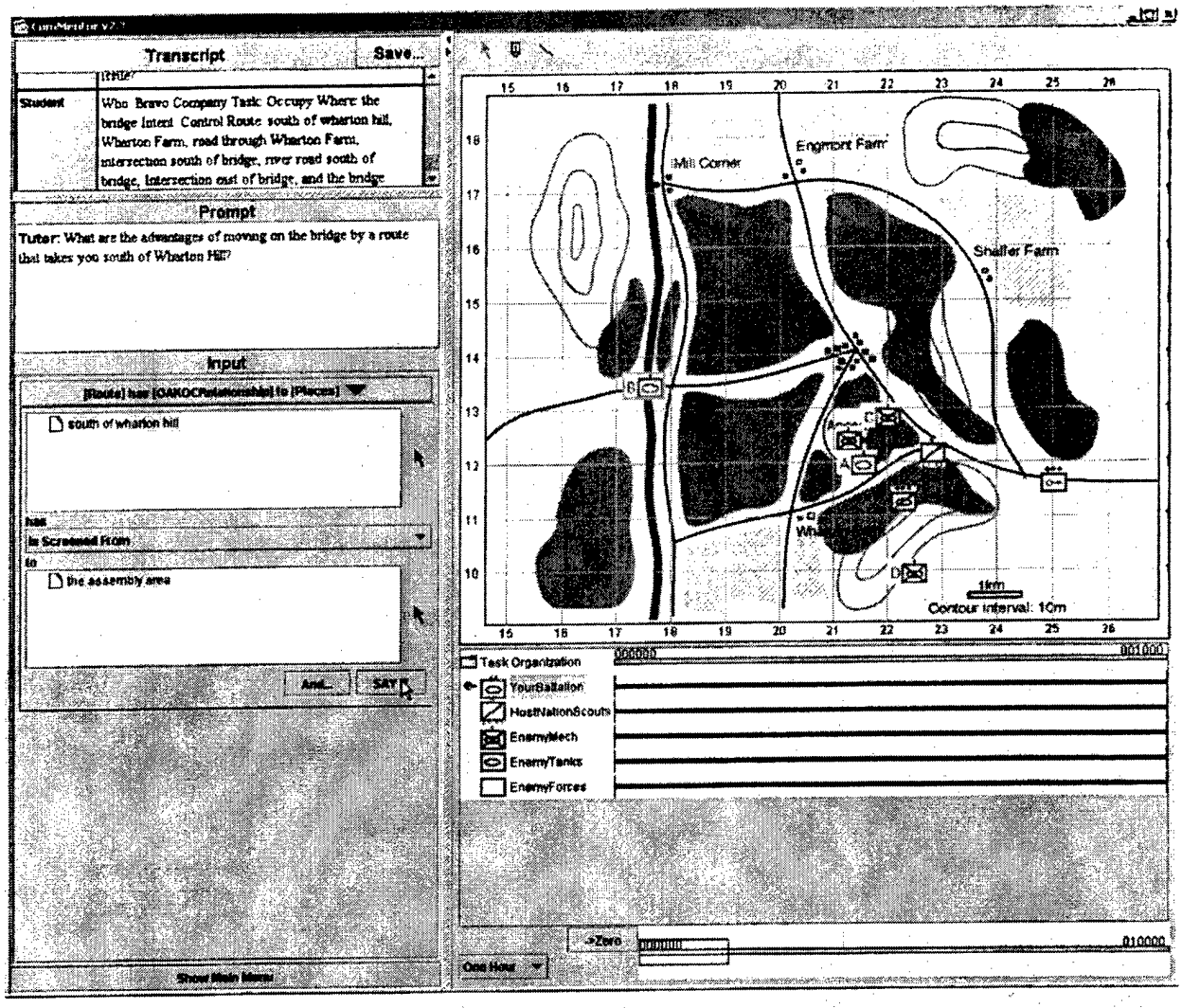


Figure 84. To The Bridge: Student's Answer to 1st Question.

The tutor has a couple of other reasons in mind why this might be a good route. It probes for a second reason, at first without success, but with an extra hint, the student provides the rational the tutor was looking for. Figure 85 shows the student providing that answer.

Tutor	In addition to offering cover and concealment from the known enemy position at the assembly area, what else does that route screen you from?
Student	I Don't Know.
Tutor	What piece of terrain might the enemy try to use for observation or fire that this route screens you from?
Student	South of Wharton Hill has IsScreenedFrom to Hamlet hill.
Tutor	Yes. Even if the enemy does get forces up on Hamlet Hill they still won't be able to see what you're doing if you run your forces south of Wharton Hill.

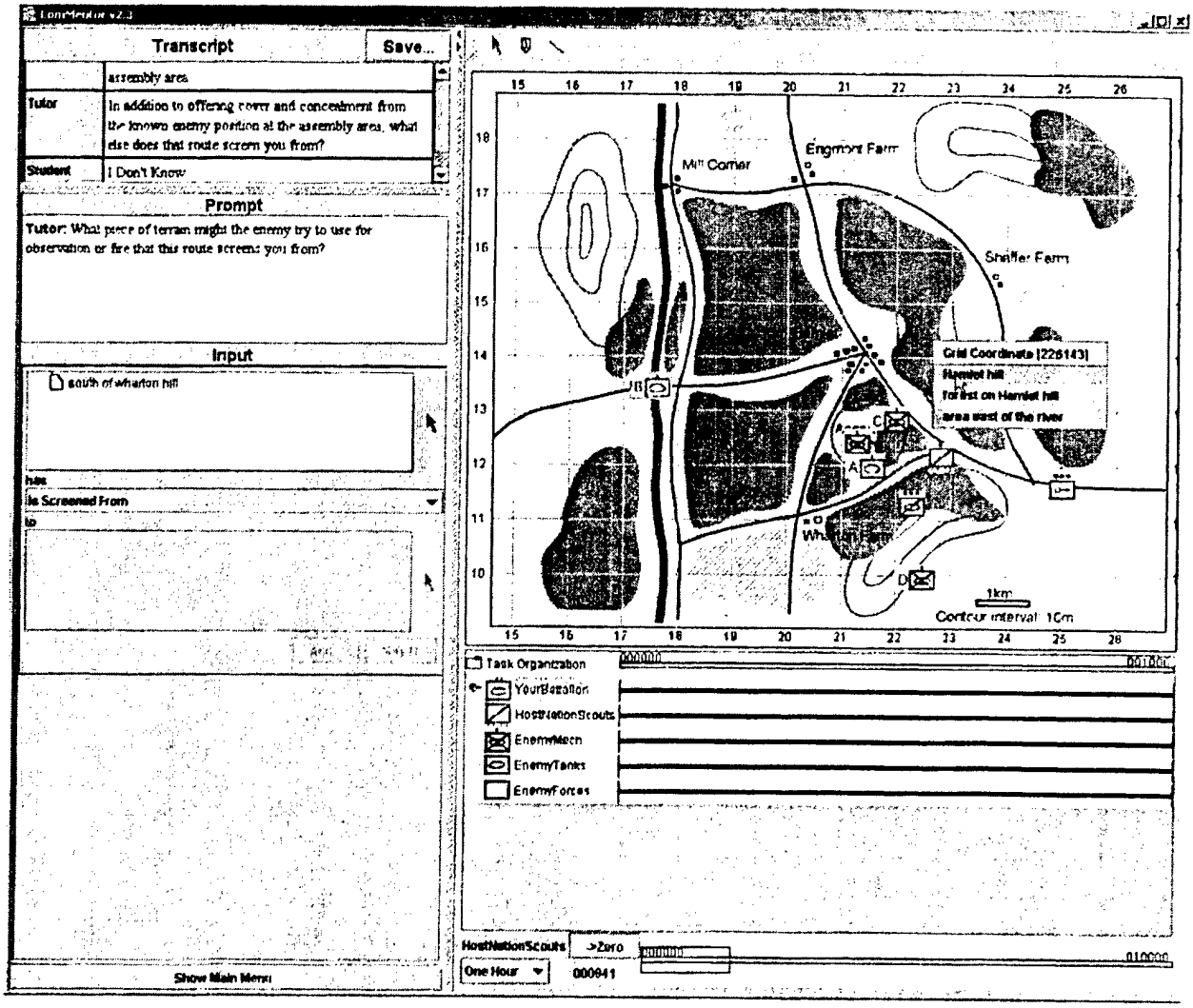


Figure 85. To The Bridge: Student's Answer to 2nd Question.

Again, the tutor comes back with a question intended to elicit another possible rationale for the southern route. This point is a more subtle inference about the situation, based on what

has happened earlier in the scenario (e.g., the successful passage of the host-nation unit). Figure 86 shows the student constructing this answer, and gives a look at the menu of choices for characterizing the state of a place.

Tutor	Given what you know about the situation so far what else is nice about using Wharton Hill as a screen?
Student	Wharton Hill is Unoccupied.
Tutor	Right. Given the host nation scouts got through all right, it might be reasonable to assume the enemy had not yet gotten established up on Wharton Hill. Wharton Hill may even be available to you to use for your own overwatch as you move to the river.
Tutor	Going south of Wharton Hill makes sense because you avoid running along the road by Wharton Farm with its forests to the north that could be hiding enemy dismounts. And while you still have the hill over you, there's reason to believe the enemy isn't up there yet, and so that hill, rather than being a threat, may provide good cover and concealment. You can even use the hill for overwatch positions. Finally, going to the south is probably a shorter route to the intersection near the river.

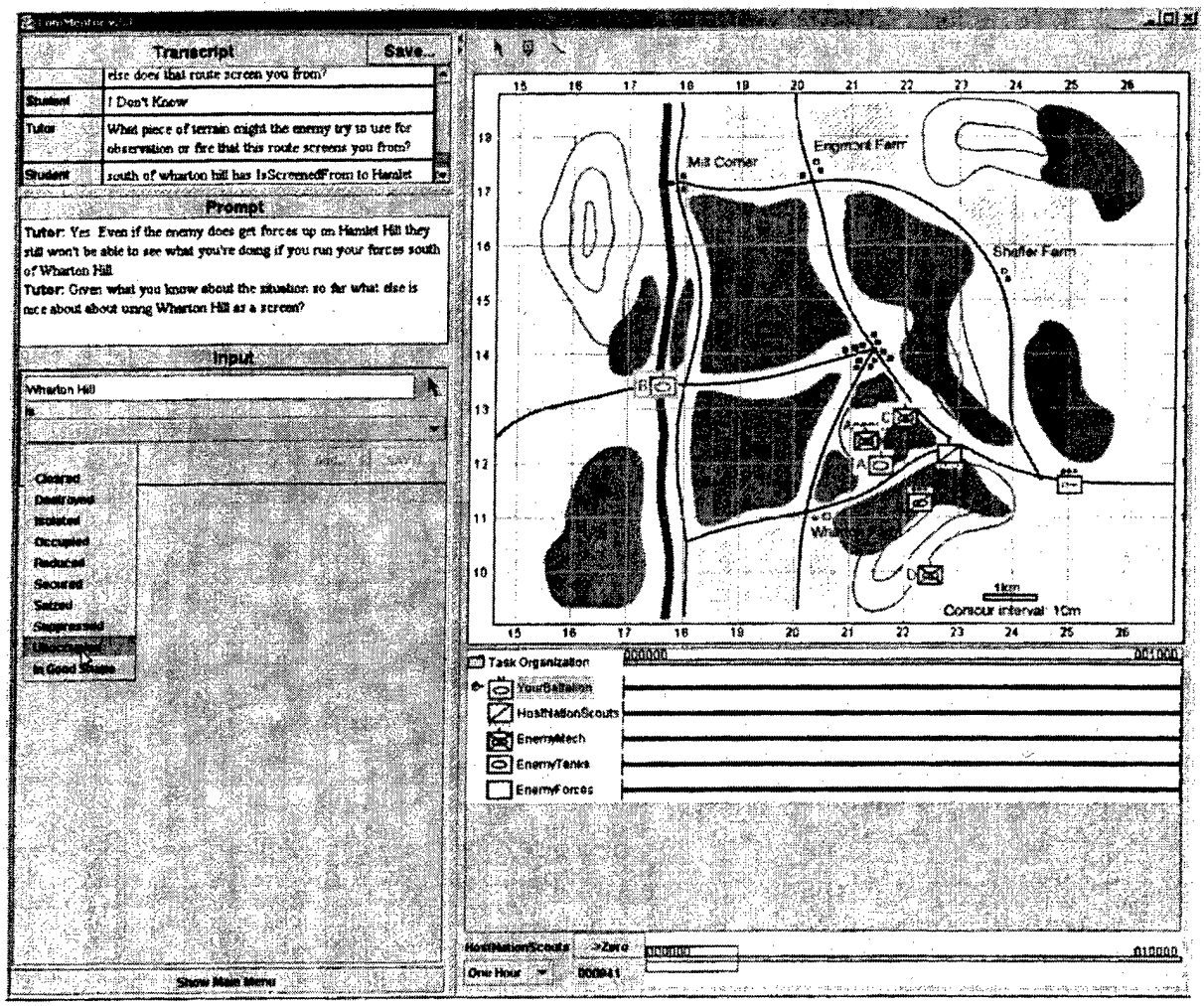


Figure 86. To The Bridge: Student's Response to 3rd Question.

The tutor now switches gears again, and tries to get the student to think about the pieces of the evolving plan as part of a unified maneuver. Figure 87 shows the student constructing a response to the first question in this new line of discussion. It also shows a major chunk of ComMentor's menu of maneuver types.

Tutor	So let's say you've got an attack going on at the assembly area, and you're also moving forces on the bridge. Let's talk about that.
Tutor	What kind of operation are you doing here?
Student	The task force is doing SingleEnvelopment on (enemy) the enemy forces.
Tutor	Right. By fighting him in the assembly area and then swinging around to hit him back at the bridge, you are doing a classic envelopment.

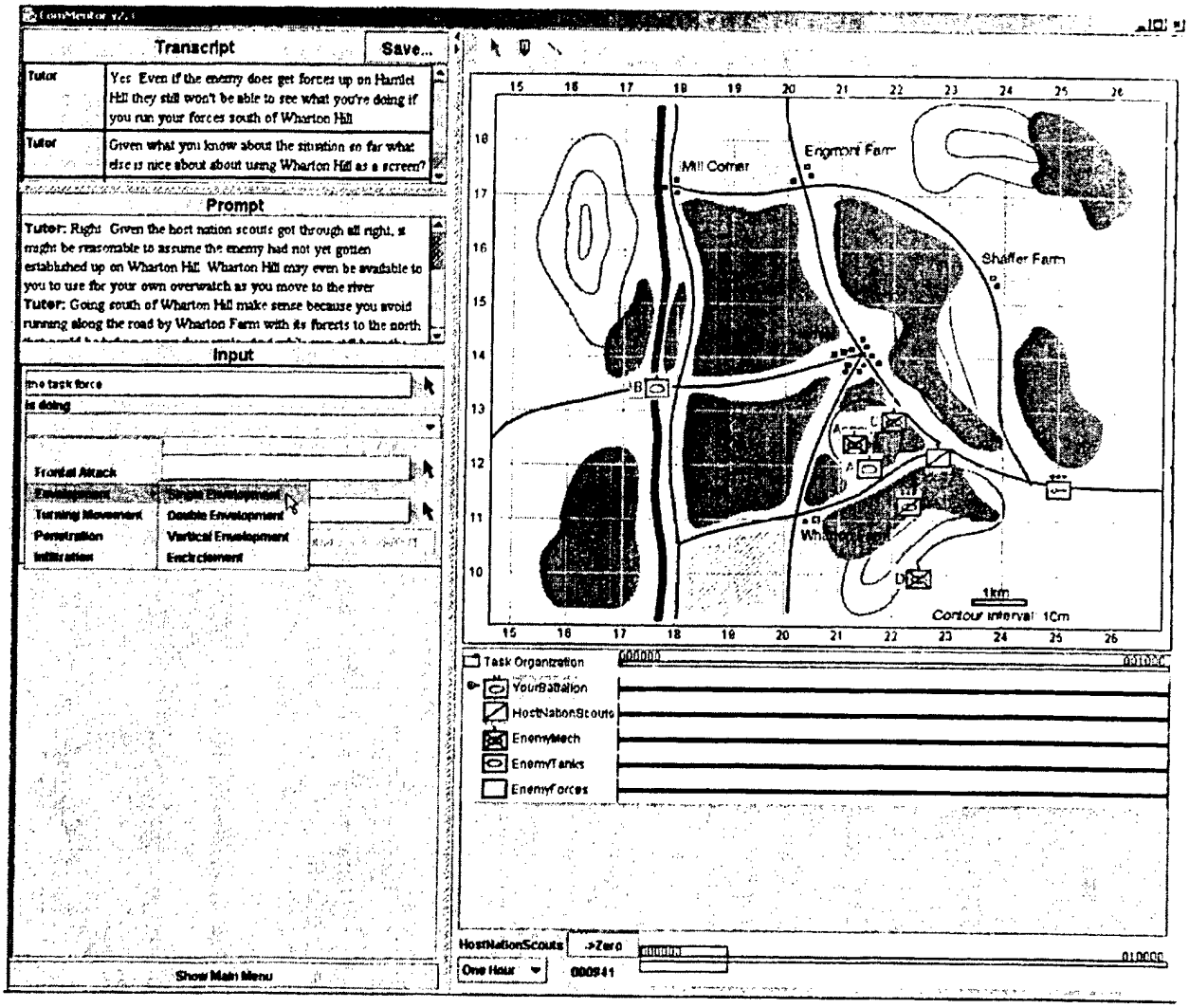


Figure 87. Envelopment: Student's Response to 1st Overview Question.

The next several turns are taken up with a sort of theoretical review of envelopments.

Tutor Let's talk about the basic roles in a classic envelopment.

Tutor	The base of any envelopment is the 'fixing' force. You can think of that as establishing the pivot point for your whole operation.
Tutor	What is the basic purpose of the fixing force?
Student	Using a unit as a FixingForce can help to Fix.
Tutor	That was kind of a freebie... The basic purpose of the fixing force is to fix the enemy in place.
Tutor	While the fixing force is keeping the enemy pinned down, what is the basic purpose of the enveloping force?
Student	Using a unit as a EnvelopingForce can help to Defeat.
Tutor	Yes. The enveloping force's purpose is to defeat or destroy the enemy by getting into their soft underbelly.
Tutor	The final key piece of almost any envelopment operation is the reserve.
Tutor	What can a reserve force do for you in the context of an envelopment?
Student	Using a unit as a ReserveForce can help to Support.
Tutor	A reserve force is useful in this context for pretty much all the standard reasons: they can reinforce your enveloping force if they succeed at tearing into the enemy, they can buttress the fixing force if they start to get into trouble, or they can attack on a third front if an enemy weakness is created by their response to the other forces.

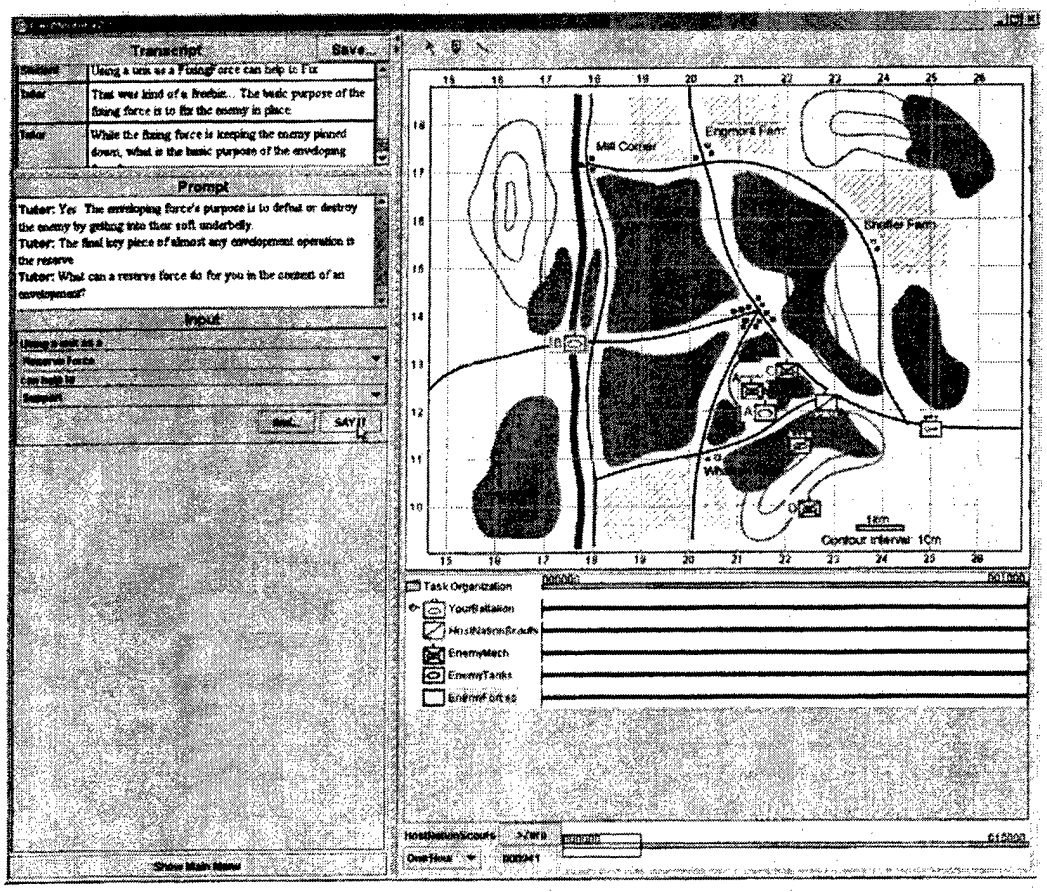


Figure 88. Envelopment: Student's Response to final Definitional Drill-Down Question.

The tutor now goes deeper in its discussion of one key part of an envelopment: the establishment of a pivot point. Figure 89 shows the student constructing an answer to the tutor's question about why you want a pivot. The battlefield-geometry menu used here is one of the more complex in the system, and the student settles for an obvious answer, which happens not be what the tutor was looking for.

Tutor	With your fixing force attacking the assembly area, and your enveloping force heading toward the bridge, you've got a kind of one-two punch going on. Just like when you're throwing combination with a hook, you need to set a pivot point to serve as a strong base. Let's talk a little about the concept of a pivot point in an engagement like this.
Tutor	Your pivot point plays an important role as a source of strength and stability as your throw your punches.
Tutor	You can think of your pivot point as your base of operations. What roles is that base playing in terms of supporting and sustaining your fight?
Student	BaseOfOperations_1 has tactical significance as BaseOfOperations.

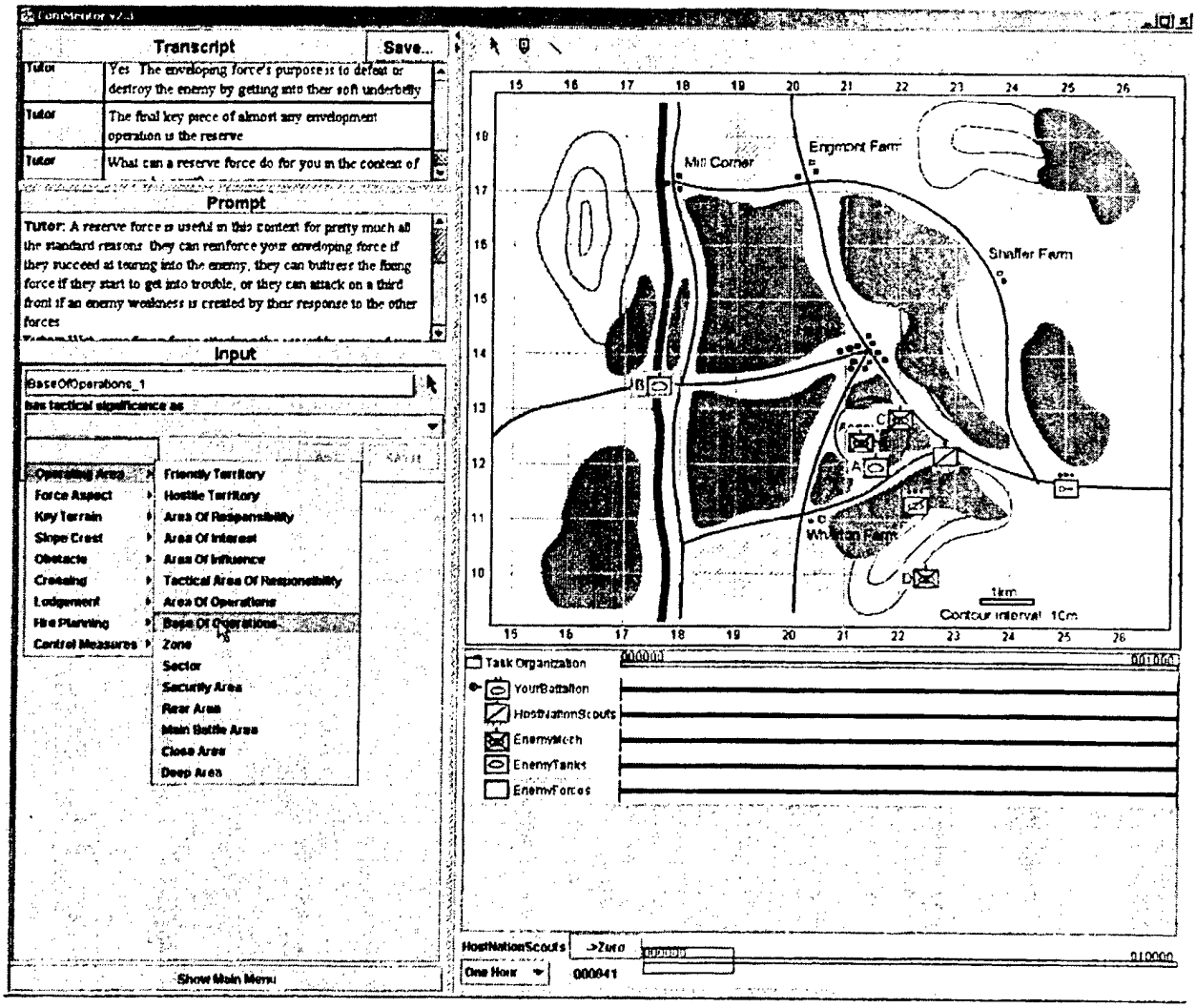


Figure 89. Envelopment: Student's Response to 1st Overview Question.

The tutor follows up with a heavily leading yes/no question to make the point it was aiming for.

Tutor	The pivot point should help you limit the enemy's movements while facilitating your own. So would it be reasonable to characterize one useful function of the pivot as being a kind of blocking position?
Student	Yes.
Tutor	A good pivot point helps you put strength behind your punch, and helps you sustain or recover from the punch.

This discussion is finished out with a request for the student to try to apply the theoretical discussion to the particular situation. Figure 90 shows the student picking out a reasonable location for their pivot point in the proposed envelopment.

Tutor	In this case, where might you want to put your pivot point?
Student	Road to choke point has tactical significance as BaseOfOperations.
Tutor	Good. In this case, you've got a plausible position right on an MSR--so you can get supplies and reinforcements in, or get casualties out--and that position can also put a cork in the bottle to limit what the enemy can do.

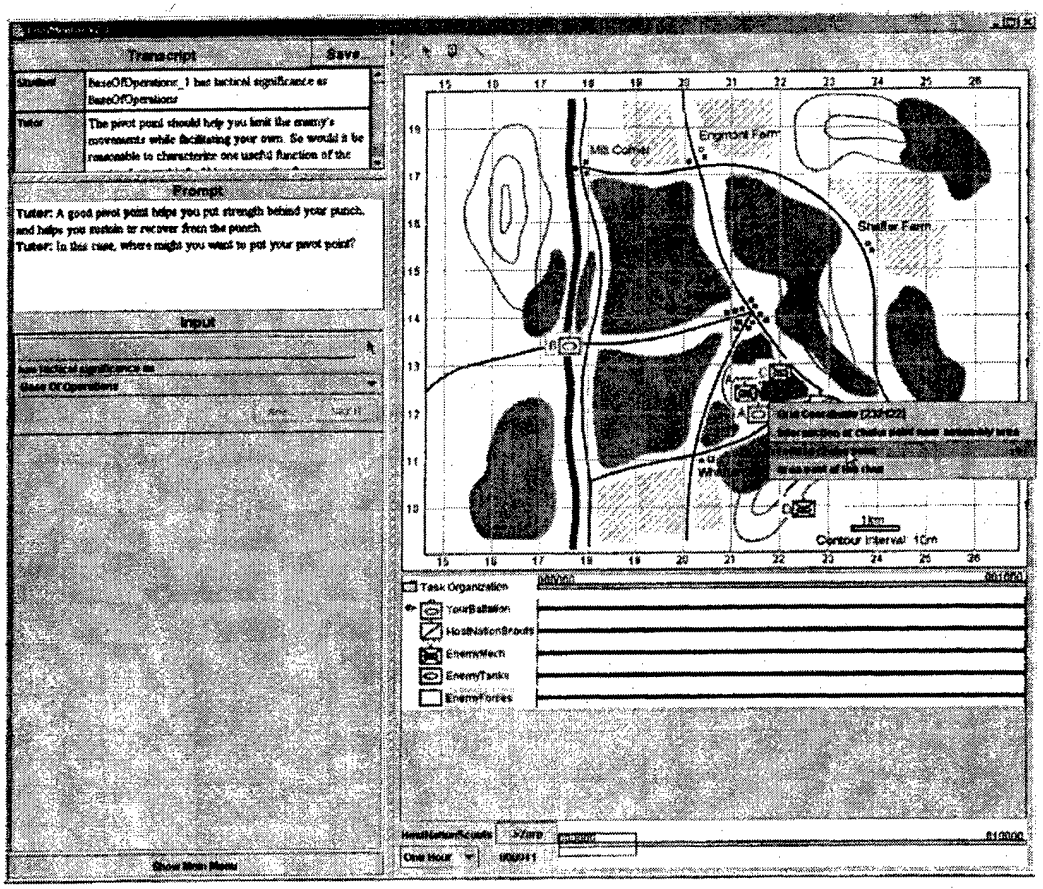


Figure 90. Envelopment: Student's Proposed Pivot Point.

The tutor moves on to consider the uses of a reserve force in the student's plan for this situation. The initial question asks what the student is intending to achieve in this situation. He correctly answers: secure a bridgehead. This is a relatively rare case where the student has to type the place description from scratch (there is no menu, and no pre-defined place on the map called a bridgehead). Figure 91 shows the student's response.

Tutor	It's good to have designated a reserve. But let's talk in some more detail about how you imagine using the reserve, and the consequences of thinking ahead in that way.
Tutor	So one good use of a reserve is to capitalize on and reinforce success.
Tutor	In this context, what would constitute 'success?'
Student	FriendlyUnit_4 is aiming to Secure or (location) bridgehead.
Tutor	Right. We would have succeeded here if we managed to get back control of the bridge, which would shut down the flow of enemy troops, allow us to clean up the mess east of the river, and prepare for our offensive west.

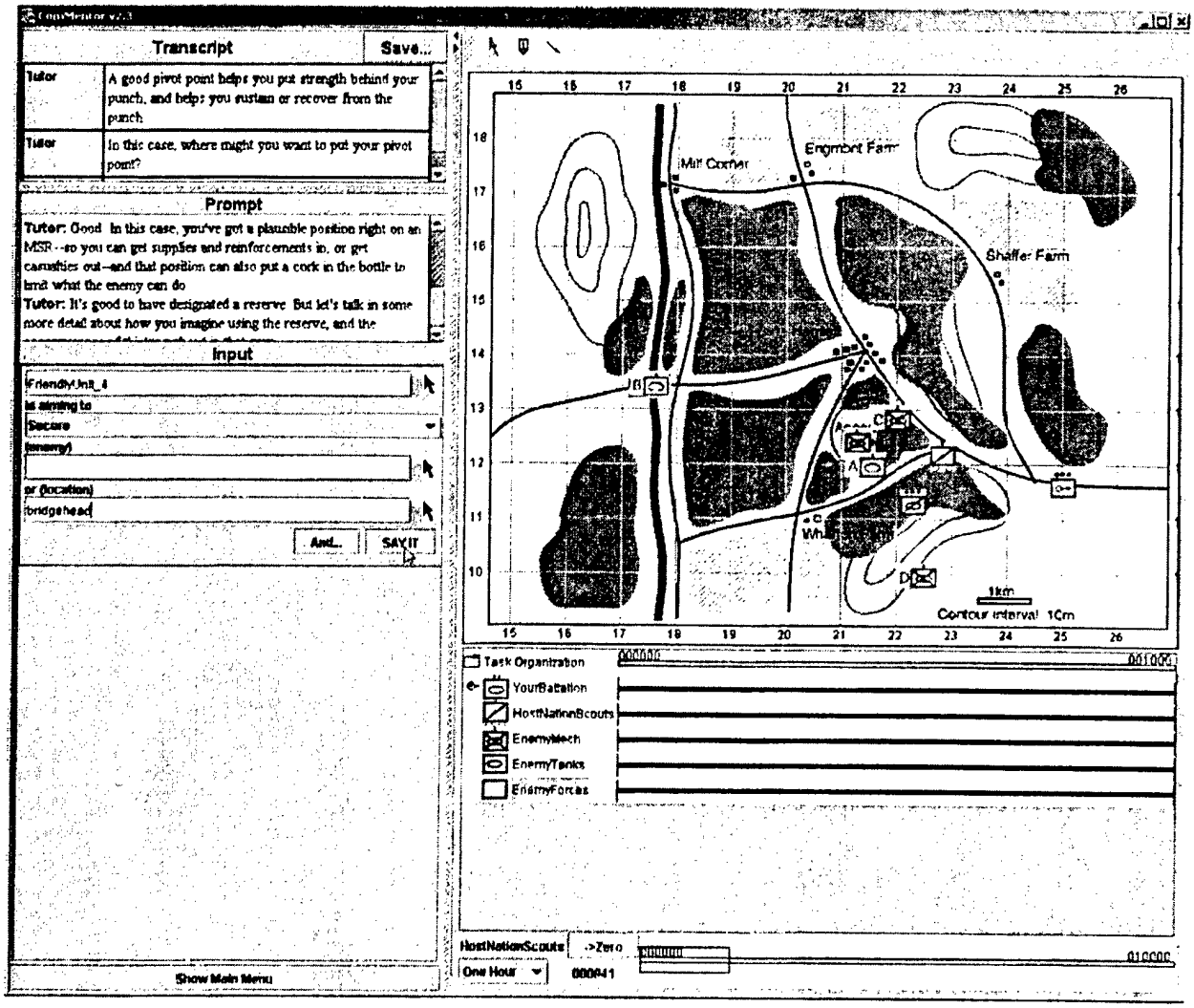


Figure 91. Reserve: What Constitutes Success.

A set of follow-up questions push the student to pick out places where they might want their reserve force to end up in the case where the reserve is reinforcing success. Figure 92 shows the student answering the first such probe.

Tutor If you were going to send your reserves to help reinforce that kind of success, where might you want to send them?

Student Delta Company is located at hill west of river.

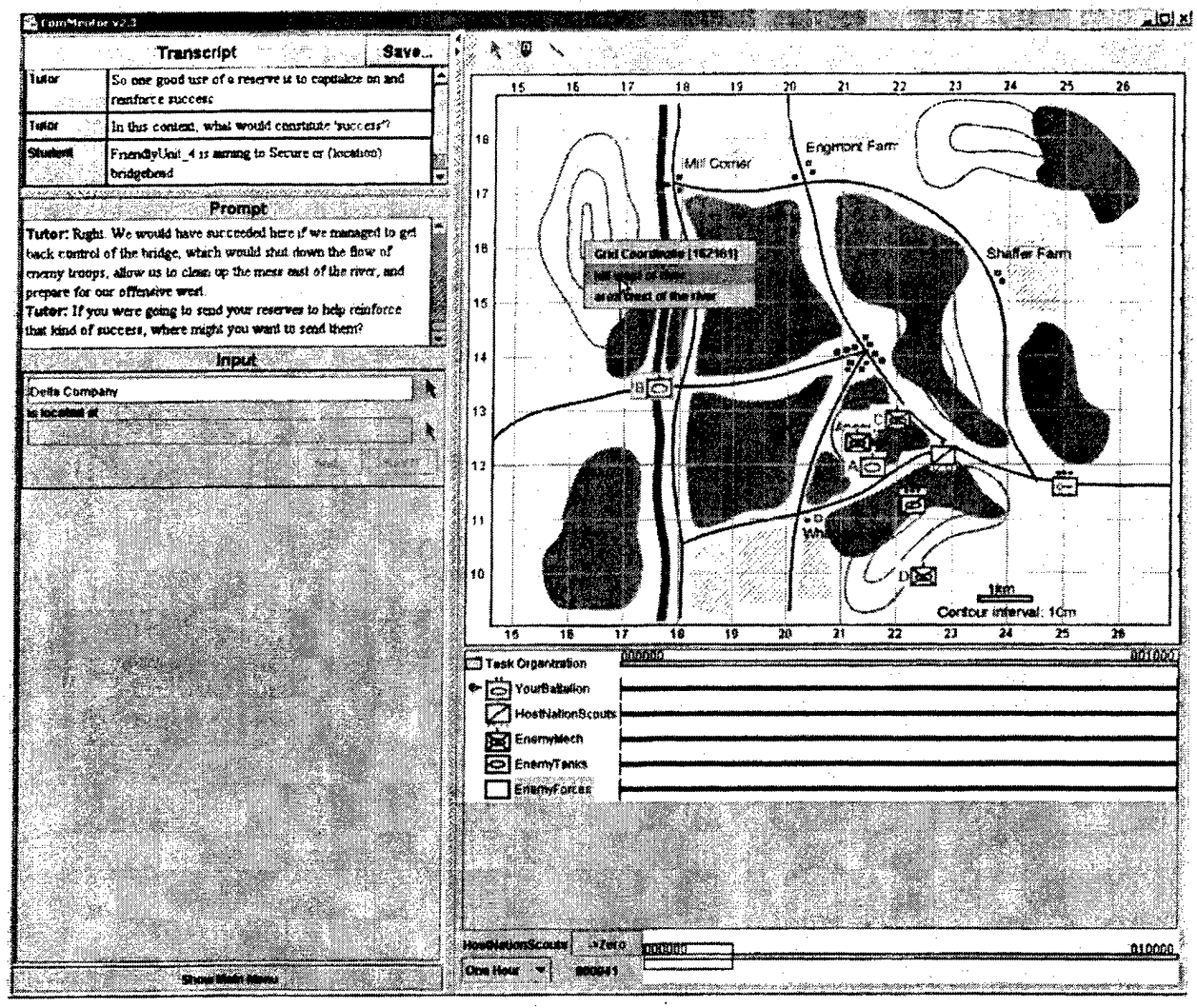


Figure 92. Reserve: Reserve Forces End-State to Reinforce Success.

The tutor follows up with another request for a location the student might want the reserves to end up.

Tutor Remember, to hold the bridge, you have to do more than just sit on it. You probably want to get out and secure some area in front of it. So where might you want your reserves to end up?

Student Delta Company is located at southwest forest.

Tutor You're looking to build a bridgehead, so I would think you'd want to get some forces up on the hill west of the river, or maybe in the treelines of the woods west of the river.

Having established some positions a reserve might ultimately aim to occupy, the tutor asks what kind of reserve force the student thinks is most appropriate in this situation. This is a question that could only be answered once a purpose had been identified for the reserves. Figure 93 shows the student's answer. This particular answer required the student to type in the phrase "tank company" (though clicking on a particular tank company, such as Delta would have worked as well).

Tutor If you were going to put a force up on the hill west of the river, what kind of unit would you like it to be?

Student Tank company is doing Occupy or (location) hill west of river.

Tutor I agree. It's not a bad idea to put some big guns up on commanding terrain like that, so when it comes to this possible mission, I would probably lean towards wanting a tank unit as my reserve.

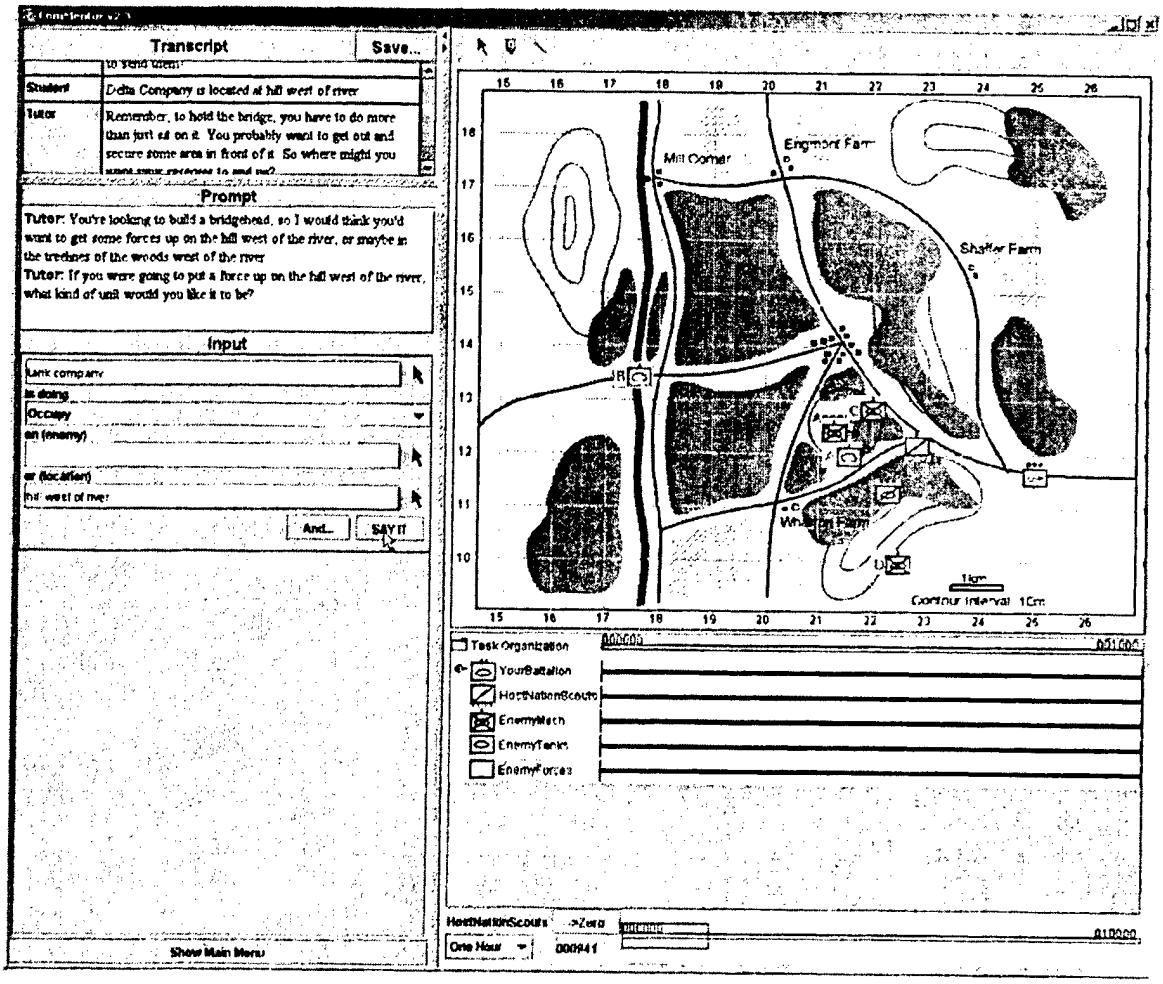


Figure 93. Reserve: Choice of Reserve Forces to Reinforce Success.

The discussion of reinforcing success is only one consideration in the detailed planning for a reserve force. Next the tutor turns to the perhaps more commonly considered issue of preventing failure. Figure 94 shows the student responding to the tutor's query about where the enemy might go when confronted with a tank battalion to the south.

Tutor	Reinforcing success is one of the best uses for a reserve, but probably a more common way people think about a reserve force is to try to prevent failure.
Tutor	While you are busy punching the enemy in the nose and bottling him up at the assembly area, where might he try to go?
Student	EnemyUnit_56 might be located at road north from Hamlet.
Tutor	When a force is blocked on one path, it immediately starts to maneuver and tries to wiggle out some other way. In this situation, Red could try to go either to the north or the south.

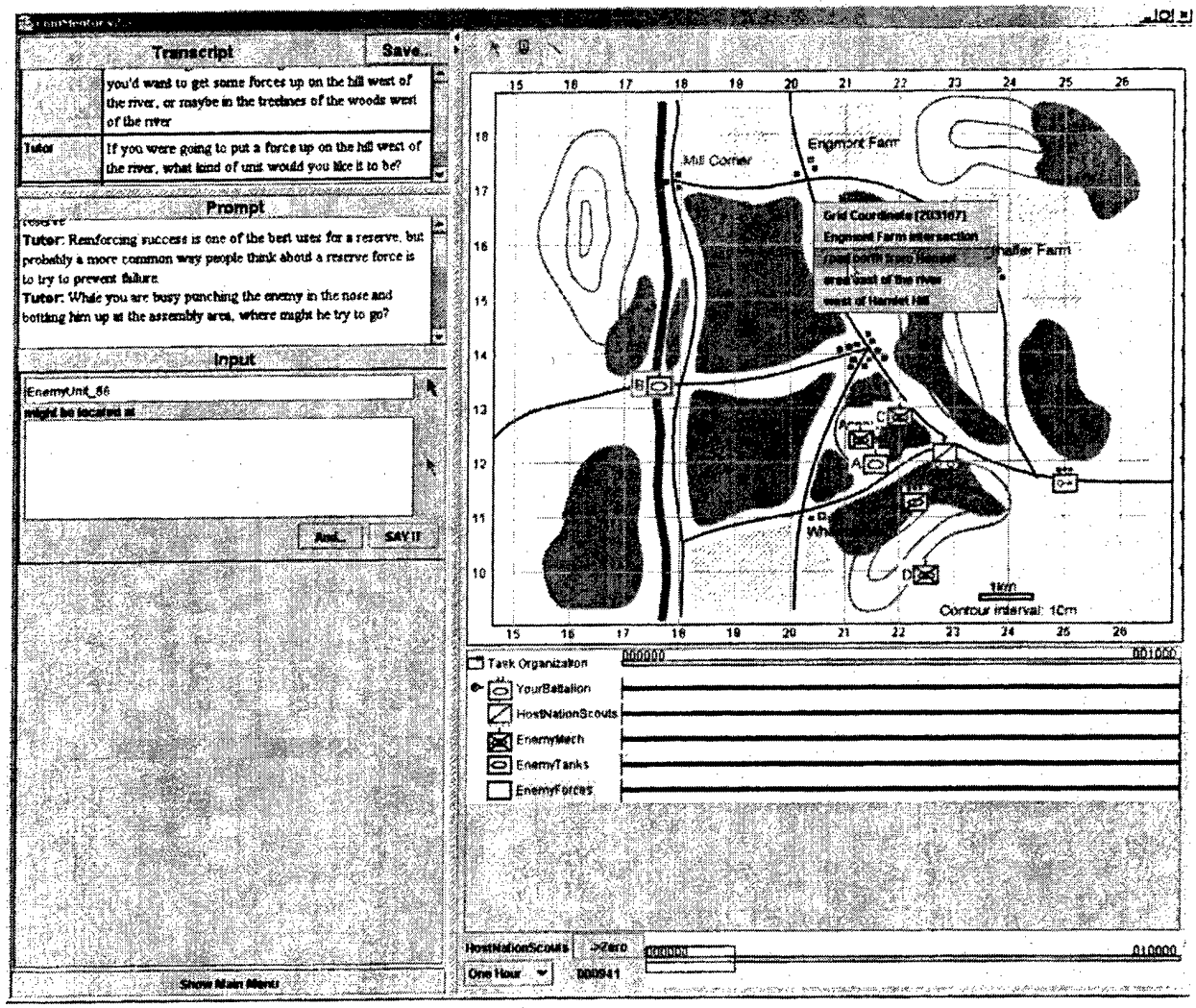


Figure 94. Reserve: Where an Enemy Might Go.

The tutor follows up on the student's suggestion that the enemy might run to the north by asking where a unit might be posted to watch for that eventuality. Figure 95 shows the student nominating Shaffer Hill as a good location to watch from.

Tutor	So if you're worried about the enemy leaking out, and you've already got forces in the south, where should you be watching?
Student	Delta Company might be located at Shaffer Hill
Tutor	Yes. The hill north of Shaffer Farm is a pretty choice location if you can grab it.

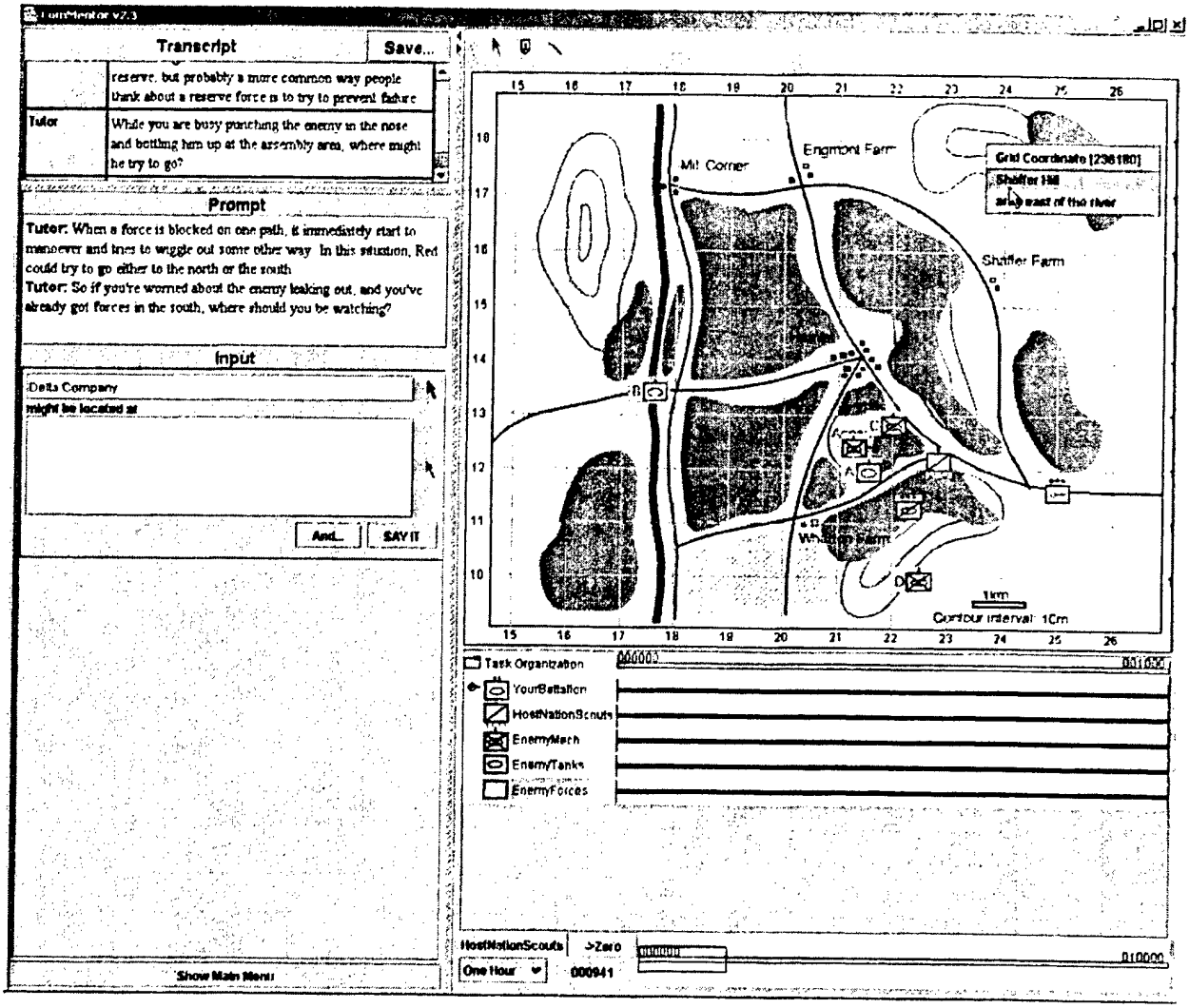


Figure 95. Reserve: Where a Reserve Might Watch for an Enemy Move to the North.

The tutor probes for yet another job the reserve might be able to help out with. Harking back to the earlier discussion of the pivot point, Figure 96 shows the student nominating the road to the choke point as a location worth watching over.

Tutor In addition to keeping their eyes on things, what part of your battlespace could the reserves take responsibility for?

Student Delta Company is aiming to Secure or (location) road to choke point.

Tutor I agree. While you're thinking about possible reserve positions with respect to what they can do to keep the basic envelopment on track, you should also think about what they can do to help you maintain a strong pivot point--securing the base of your whole operation.

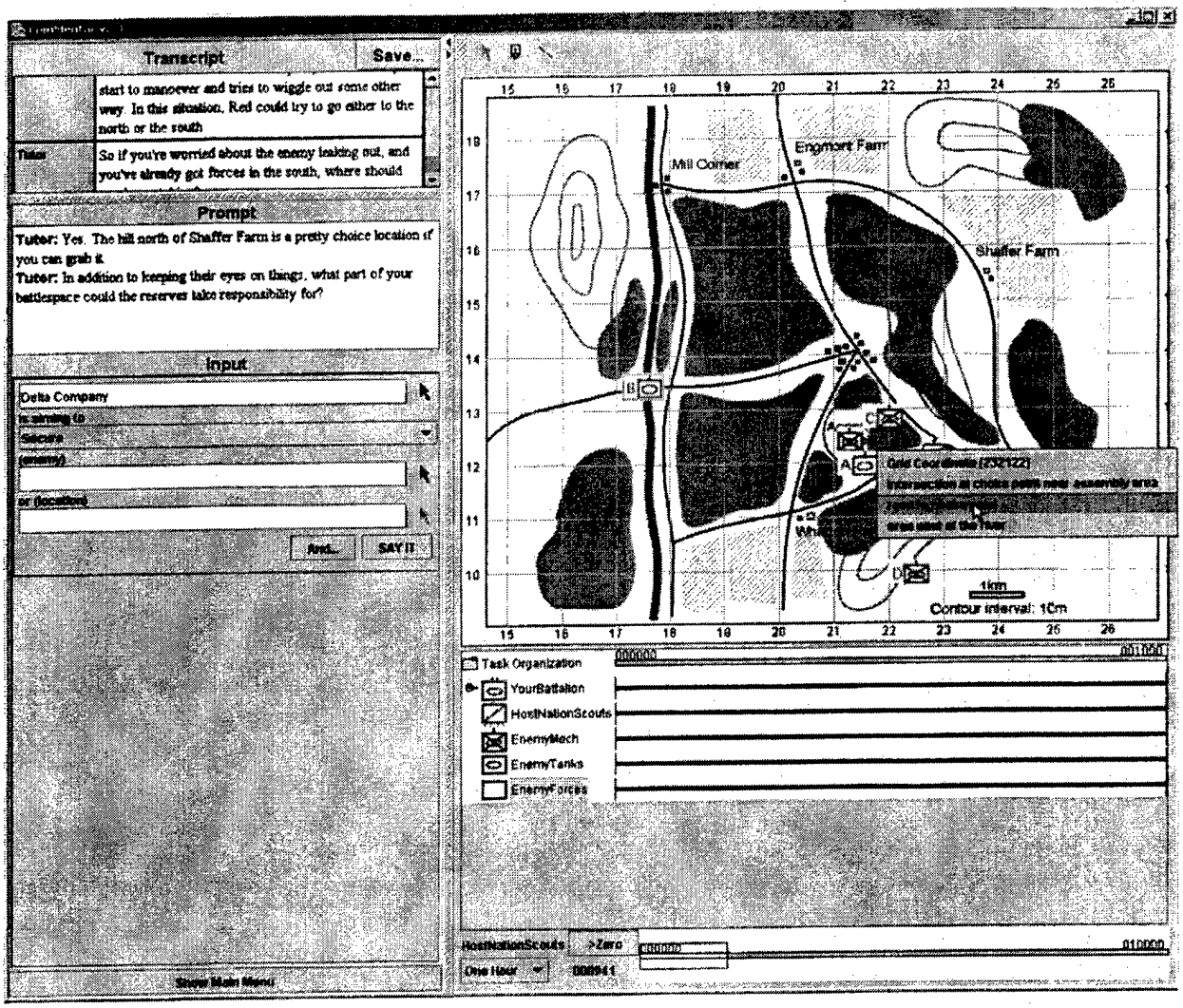


Figure 96. Reserve: Where Else a Reserve Might Help Out.

The discussion of the reserve force wraps up with consideration of various alternatives the student might use to get to the bridge. The input forms currently available somewhat limit this interaction, but the system is able to use it map manipulation abilities on the output side to highlight a range of alternate routes the student might consider. The interaction wraps up with a query about where the student would position their reserve given the various issues discussed along the way. Figure 97 shows the student making his choice.

Tutor	Another good use for your reserves would be to provide an alternate way to achieve success. Here that means some alternate route for getting decisive forces to the bridge.
Tutor	Suggest some routes to the bridge by selecting key places you might go through for the different approaches.
Student	FriendlyUnit_5 might be located at Engmont Farm intersection.
Tutor	You could go along the road past the assembly area and Wharton Farm, or you could swing south of Wharton Hill. You could also swing way north past Shaffer Farm. And of course there's always the run up the middle through Hamlet itself.
Tutor	Where might you want to put your reserve so it would be ready to attempt an alternate approach to the bridge?
Student	Delta Company is located at Shaffer Hill.

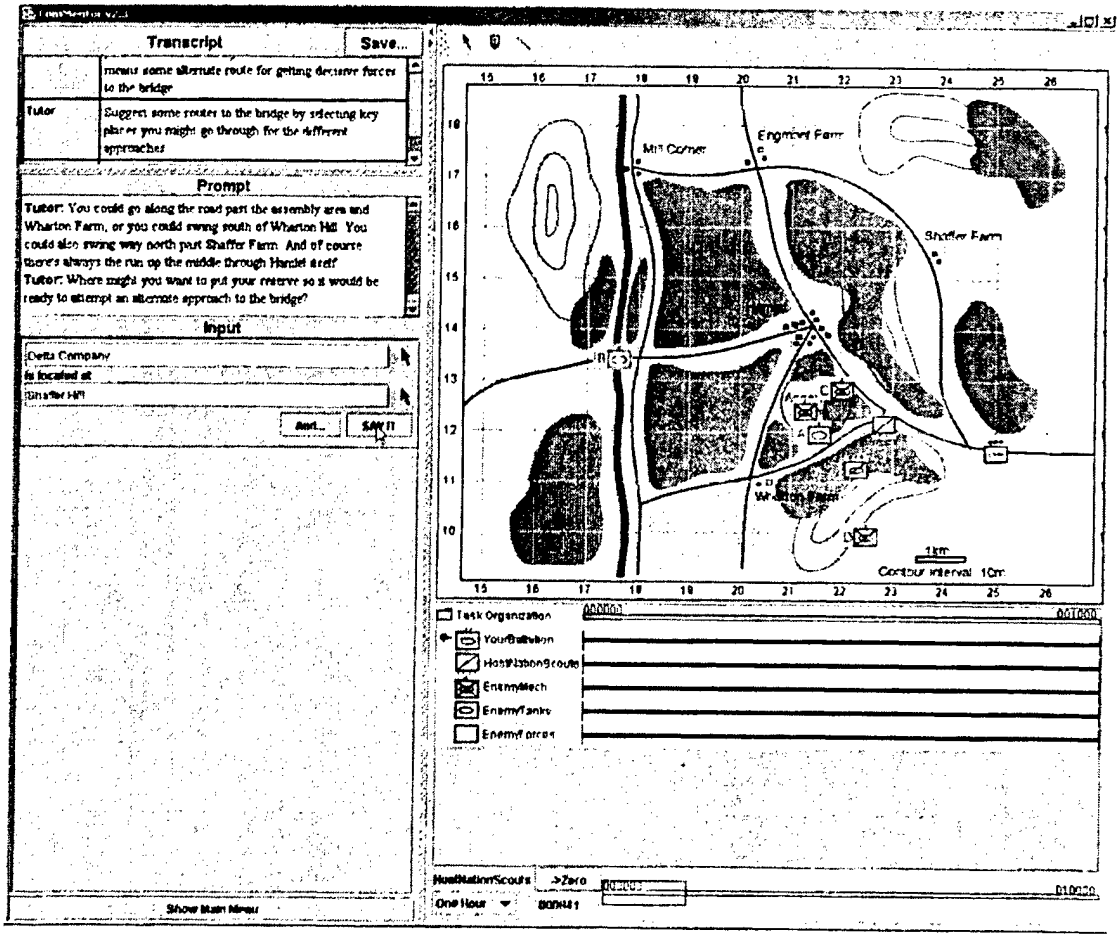


Figure 97. Reserve: The Student's Final Choice of a Location for the Reserve.

Following a brief summary about positioning of the reserve, the tutor concludes the session with a short fill-in-the-blank questionnaire. This is the current stand-in for the reflective session that our SMEs used during live sessions. We have added an extra question about the performance of the ComMentor system. By analyzing session transcripts and student responses to these questions, the quality of ComMentor's general and scenario-specific behavior can be improved over time.

Tutor Between wanting to be able to follow on your main assault on the bridge, securing your operational base, and preparing to strike at the bridge by an alternate route, you now have a lot of constraints on where you might put your reserve. But this is the kind of analysis you want to be doing so you can make the right tradeoffs.

Tutor I don't have anything else to discuss about this scenario, but I recommend and request you take some time to fill out the questionnaire to the right. Writing up your answers should help you crystalize what you learned during this session, and it will also be helpful to the system developers. I hope you enjoyed this session with ComMentor. Thank you for your time.

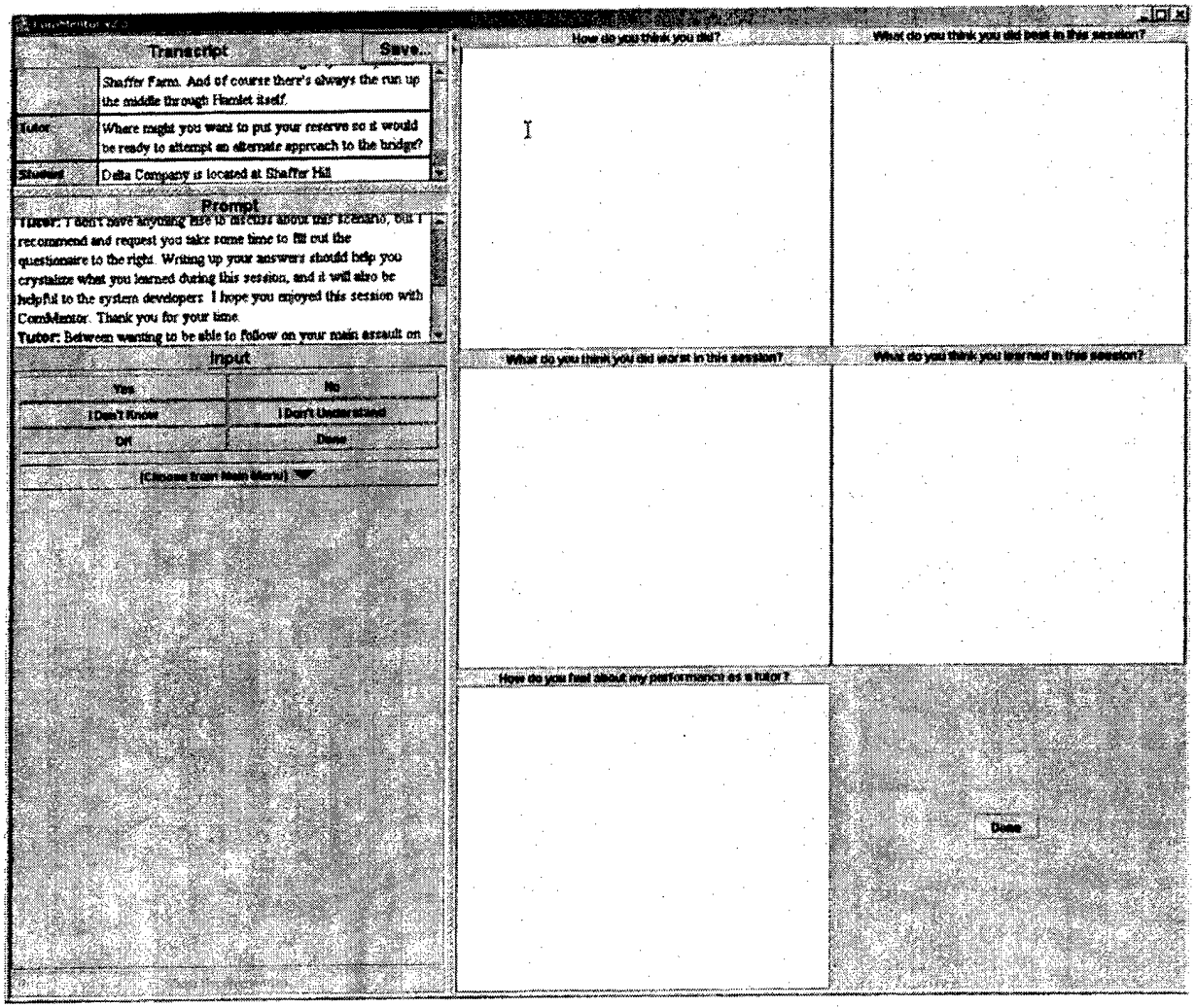


Figure 98. Finale: The Reflective Questionnaire.