Developing an Adaptive ADL Solution for Training Medical Teams

Sowmya Ramachandran, Michael Cramer
Stottler Henke Associates, Inc.
San Mateo, CA
Sowmya@stottlerhenke.com
cramer@stottlerhenke.com

Donald L. Harville, Alan Ashworth
AFRL/HEAI
Brooks City Base, TX
donald.harville@brooks.af.mil
Alan.Ashworth@brooks.af.mil

ABSTRACT

Training military medical personnel to maintain readiness for medical emergencies and combat-related operations is a critical problem. Distance learning solutions are required for providing effective training while minimizing time away from the important peacetime duty of providing quality medical care to military personnel. Since medical emergencies are unexpected, it is important to dynamically generate customized courses to address the particular emergency. Since time is at a premium in such situations, it is important to address the precise learning needs of the medical team being trained.

We are developing an Intelligent Tutoring System (ITS), called ADAPT-MD, for military medical teams in combat and emergency procedures. Using a scenario-based approach, this system provides adaptive instruction that is customized to individual teams and their members. Team training poses challenges beyond individual training and few ITSs address this problem. Issues like student modeling, team performance evaluation, tailoring the challenge level of scenarios to student expertise, etc. take on added complexity. Adapt-MD addresses these issues by using a compositional approach to scenario generation and student model representation. A student model of a team comprises of a model of the team as a whole and models of each of the individual members. In addition to representing each members own state of expertise, the student model also represents his knowledge of the other team members' tasks and abilities. ADAPT-MD has facilities for creating scenarios that are adapted to individual team members' expertise levels. Simulations can include simulated intelligent entities to take the place of team members. The ADAPT-MD framework includes an authoring tool for specifying presentation content, domain knowledge, training scenarios, and instructional strategies. This framework is currently being applied to create an ITS for training hyperbaric treatment teams. It is, however, domain-independent and can be used to create ITSs for other medical domains.

ABOUT THE AUTHORS

Dr. Sowmya Ramachandran is a research scientist at Stottler Henke Associates, Inc. Dr. Ramachandran received her Ph.D. is Computer Science from the University of Texas at Austin. She has a strong background in a wide variety of Artificial Intelligence techniques, including Intelligent Tutoring Systems and Machine Learning. Her research interests include application of Artificial Intelligence techniques to Education Technology with a focus on addressing motivational, affective, and meta-cognitive issues. Dr. Ramachandran has headed several ITS development efforts, including one for adult literacy enhancement and one to teach Algebra to at-risk high-school students. She is currently developing a general-purpose authoring framework for rapid development of ITSs.

Michael Cramer is a software engineer at Stottler Henke Associates, Inc. He holds a Master's Degree in Computing in the area of Intelligent Systems from the Georgia Institute of Technology. He is interested in computational perception and robotics and his research efforts include the Tactical Mobile Robots project funded by DARPA where he and his colleagues at the Georgia Tech Mobile Robotics Lab addressed behavior design and evaluation for urban warfare scenarios. Mr. Cramer also participated in the Personal Pet project funded by Honda Robotics where he designed an architecture for emotional behaviors for pet robots. In industry, Mr. Cramer has developed position sensing software for racecars for broadcast enhancement and has developed software models for multiple camera systems. Currently, Mr. Cramer is developing intelligent tutoring systems for Stottler Henke Associates.

Donald L. Harville is a Personnel Research Psychologist for the Air Force Research Laboratory at Brooks City-Base, TX. He received both his BA in Biology with Highest Honors and his Ph.D. in Industrial-Organizational Psychology from the University of Tennessee. His MBA in E-Commerce Management was from Our Lady of the Lake University. His research interests include training and selection utility in dollars and productivity improvements.

Dr. Alan Ashworth is the Director of Advanced Distributed Learning (ADL), US Air Force Research Laboratory, which reports to the Undersecretary of Defense, Personnel & Readiness. Dr. Ashworth received a B.S. in psychology from the University of Texas 1990, an M.S. in experimental psychology from Yale University in 1992, an M.Phil. in epistemology from Yale University in 1993, and a Ph.D. in cognitive neuroscience from Yale University in 1995. He has published and presented research in the areas of mnemonic representation, categorization, human learning, and imagery chronometrics. He is an adjunct professor at the University of Texas at San Antonio.

Developing an Adaptive ADL Solution for Training Medical Teams

Sowmya Ramachandran, Michael Cramer
Stottler Henke Associates, Inc.
San Mateo, CA
Sowmya@stottlerhenke.com
cramer@stottlerhenke.com

Donald L. Harville, Alan Ashworth
AFRL/HEAI
Brooks City Base, TX
donald.harville@brooks.af.mil
Alan.Ashworth@brooks.af.mil

Medical Training in the Military: Issues and Challenges

The military has an extensive system of medical care, both during peace and war. In addition to active duty medical professionals, the military also calls upon thousands of reservists to provide medical support during combat or OOTW (Other Operations than War) missions. The military medical system is charged with two very important missions [Hosek and Cecchine 2001]:

The Benefits Mission: To provide quality medical care to military personnel, their dependents, and other qualified people.
The Readiness Mission: To provide, and to be ready to provide medical support during military operations.

The benefits mission conflicts with the readiness mission as the latter requires periodical training that takes medical personnel away from the job of providing day-to-day medical care [Hosek and Cecchine 2001]. Often this training is done with combat units, and involves travel away from the home base. The benefits mission not only provides quality medical care to military personnel, but also helps medical professionals maintain readiness with respect to non-combat related health issues. Maintaining a good balance in the trade-off between training through the benefits mission and training through the readiness mission is an important issue for the military.

Availability of high-quality computer-based, distributed training programs will help resolve this issue significantly. Medical personnel can refresh their skills with respect to combat and other military operations with this type of training. Distributed, computer based training will allow personnel to train from their home bases while performing benefits duties at least part-time. The burden of deploying reserves to substitute for the personnel assigned to training is significantly reduced.

While computer-based training cannot replace live training, it has been shown to be a highly effective complement, especially when it provides scenario-based instruction with realistic simulators [Schank 1995]. However, the effectiveness of such simulation-based training hinges crucially on the availability of instructional support from an instructor. Even then, the traditional classroom scenario of one instructor to many students cannot provide the sort of one-on-one instruction that is needed to achieve across the board learning gains. Different people have different training requirements. Depending on their backgrounds, their years of experience, and their learning speed, some trainees may need less hours of training than others to reach the same level of proficiency at a given task. Moreover, different students need training in different areas, even within the same tasks. Traditional, instructor-led training is inefficient because it cannot accommodate such differences. In a class of tens or hundreds of students, it is impossible to present a course that is customized to individual differences, with the result that some trainees have to stay the required number of hours even when they have mastered the skills, while others do not acquire enough proficiency in the given amount of time. This holds true for team training as well. The only good way to achieve such customized instruction is through one-on-one instruction. The two-sigma problem [Bloom, 1984] describes the fact that students receiving one-on-one instruction perform two standard deviations better than students receiving conventional instruction; the problem is to realize these benefits without requiring an equal number of teachers and students.

The Internet also provides unprecedented opportunities for distributed development and re-use of course content. However, much of the today's mainstream computer-based training is designed to be self-contained and usually addresses one particular subset of a domain. Creating overlapping courses requires duplication of course content. Moreover, the content of any course is pre-determined during creation and is fixed thereafter. This inflexible approach presents difficulties when the training needs of an organization are not exactly predictable. The missions that military personnel are sent on can be varied, unpredictable, and sudden. For example, health personnel on a peace mission might have to deal with an unexpected outbreak of cholera in a refugee camp. Or they can be called in to deal with a crisis involving bio-terrorism. A more flexible training framework where a course can be created that is automatically and dynamically tailored to the situation and learner would be beneficial. Such a course would select from a vast collection of content modules residing on the Internet. Course creators will no longer have to anticipate all the training needs. Instead, multiple authors can create separate modules at different times. Appropriate courses will be automatically assembled from this repository as needed in a timely manner. This fits in with the military's Advanced Distance Learning (ADL) initiative (www.adlnet.org) whose objective is to develop an open architecture for online learning. The SCORM standard, developed under this initiative, facilitates the creation of a repository of course objects that can be dynamically assembled to create online training material. The proposed effort offers the opportunity to advance the technology in the direction of sharable content.

Last but not least, computer-based training has mostly focused on the problem of training an individual. In many organizations, including the military, a person rarely works in isolation. Most jobs involve working in teams with shared goals and mission. While the performance of an individual within a team is critical, the performance of a team as a whole is equally critical. Participating in a team requires special skills and knowledge beyond an individual's proficiency in his area of expertise. In order to attain maximum performance, it is not only important to train individuals in specific areas, but also to train entire teams to work together effectively.

In summary, there is a real need for developing effective, simulation-based training systems to train military medical personnel for the readiness mission. The training systems have to be server-based and distributed in order to enable distance learning. In addition to developing an individual's skills, such systems should also develop team performance skills. Finally, designing mechanisms for creating dynamic, customized courses will be significantly beneficial.

WHY INTELLIGENT TUTORING SYSTEMS (ITS) OFFER A GOOD SOLUTION

Intelligent Tutoring Systems (ITSs) offer a cost-effective way of providing one-on-one instruction while maintaining the current ratio of human instructors to students. ITSs are computer-based training systems that mimic human instructors in providing one-on-one instruction. Much like human instructors, ITSs dynamically assess and diagnose a student's knowledge and skill levels and provide training that is customized to the student's learning needs. To truly tailor instruction, ITSs create, develop, and maintain a **model of the student** which is used as a basis for automatic selection of instruction method and content, automatic diagnosis, remedial course formulation, re-testing, progress monitoring, and reporting. Many ITSs use simulations as the core of their instructional strategy. Thus, students are trained as they apply their skills and knowledge to realistic scenarios. This learning-by-doing approach has been shown to yield significant learning gains [Schank, 1995]. ITSs offer the potential to capture the pedagogical and domain expertise of the best teachers in a virtual tutor that can be replicated and disseminated in large volumes. Studies have shown that ITSs are highly effective. [Anderson et. Al., 1985] showed that students working with a LISP ITS learned knowledge in about one-third to two-thirds the time that it took a control group of students to acquire the same amount of expertise (as measured by assessment tests). [Nichols et. Al, 1992] performed an evaluation of an ITS designed to teach avionics troubleshooting skills. They found that students could acquire, in 20 hours, skills comparable to those possessed by technicians with 4 years of experience.

This paper describes an Intelligent Tutoring System environment being developed, called Adapt-MD, to train medical teams in time-critical procedures. Adapt-MD is a general ITS framework that can be applied to various medical training tasks. It is currently being applied to create an ITS for training teams of hyperbaric treatment providers.

AN INTELLIGENT TUTORING SYSTEM FOR TRAINING MILITARY MEDICAL PERSONNEL

Figure 1 shows the high-level architecture of Adapt-MD. The instructional content (SCOs) can be created by different authors at different times. Instructors and course facilitators use the **Student Management Tool** to create initial student models. This includes providing information about the training teams, and background information on the team and its members. The **Authoring Tool** enables course developers to specify the domain principles, the content presentations, assessment activities, instructional activities, simulation scenarios, and pedagogical strategies.

We will now describe the ITS and the authoring tool in greater detail.

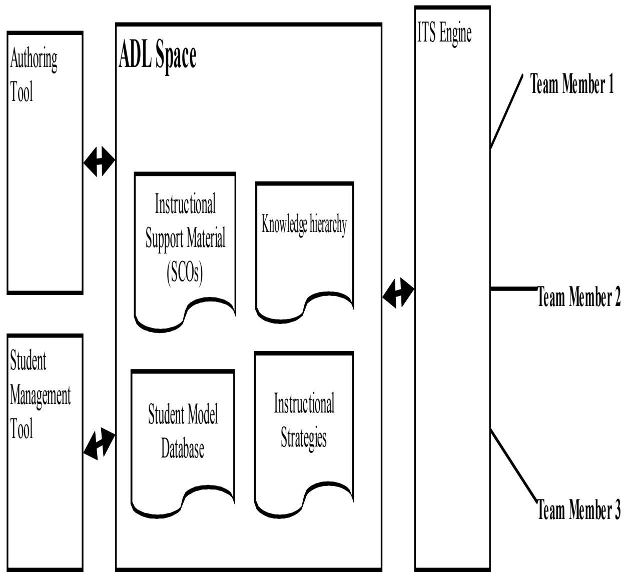


Figure 1: High-level architecture of Adapt-MD

Adapt-MD ITS

Figure 2 shows the high-level architecture of ADAPT-MD ITS. The **ITS Engine** module is the core, domain-independent component that drives the ITS. It uses data in the form of presentation content, domain knowledge, and instructional strategies to plan and deliver assessment and instruction that is customized to each individual student. The **Instructional Interface** presents the assessment/instruction to the student and includes all the interactive elements. Since ADAPT-MD is based on the learning-by-doing approach, the simulator is an essential component of this interface. The domain knowledge, presentation content, instructional strategies and the Instructional Interface together form the domain-dependent elements of the ITS. By this we mean that these elements will have to be developed anew for each domain of application. The ITS Engine, in contrast, is domain-independent and can be used to create ITSs for new domains with no modifications. The following sections describe these components in greater detail.

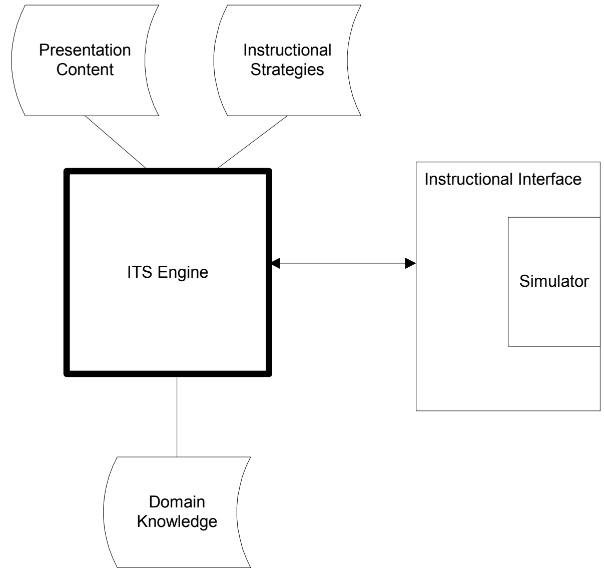


Figure 2: High-level architecture of the ITS

ITS Engine

The ITS Engine is the domain-independent component of the architecture. Figure 3 shows the architecture of the ITS Engine.

The **Assessment Manager** maintains the student model by aggregating student performance data over the long-term. It also performs diagnosis, i.e., attributing causes to observations of a student's performance in simulations and other interactions. Consider the example of a simulation where a chamber operator (the student) does not shut off descent when the chamber reaches the bottom of the dive. There are several possible causes of the error. The student may not know the bottom depth for that particular treatment, or the student may not know how to stabilize depth after a descent, etc. This component finds the most likely cause based on the student's past performance. The tutor uses Bayesian reasoning to determine likely causes. The domain principles and the possible interactions are represented as Bayesian network. Observations of student's performance are presented as evidence to the network and Bayesian inference algorithms is used to propagate this evidence through the network. Evidence accumulated over several student interactions is used to determine the most likely causes of errors.

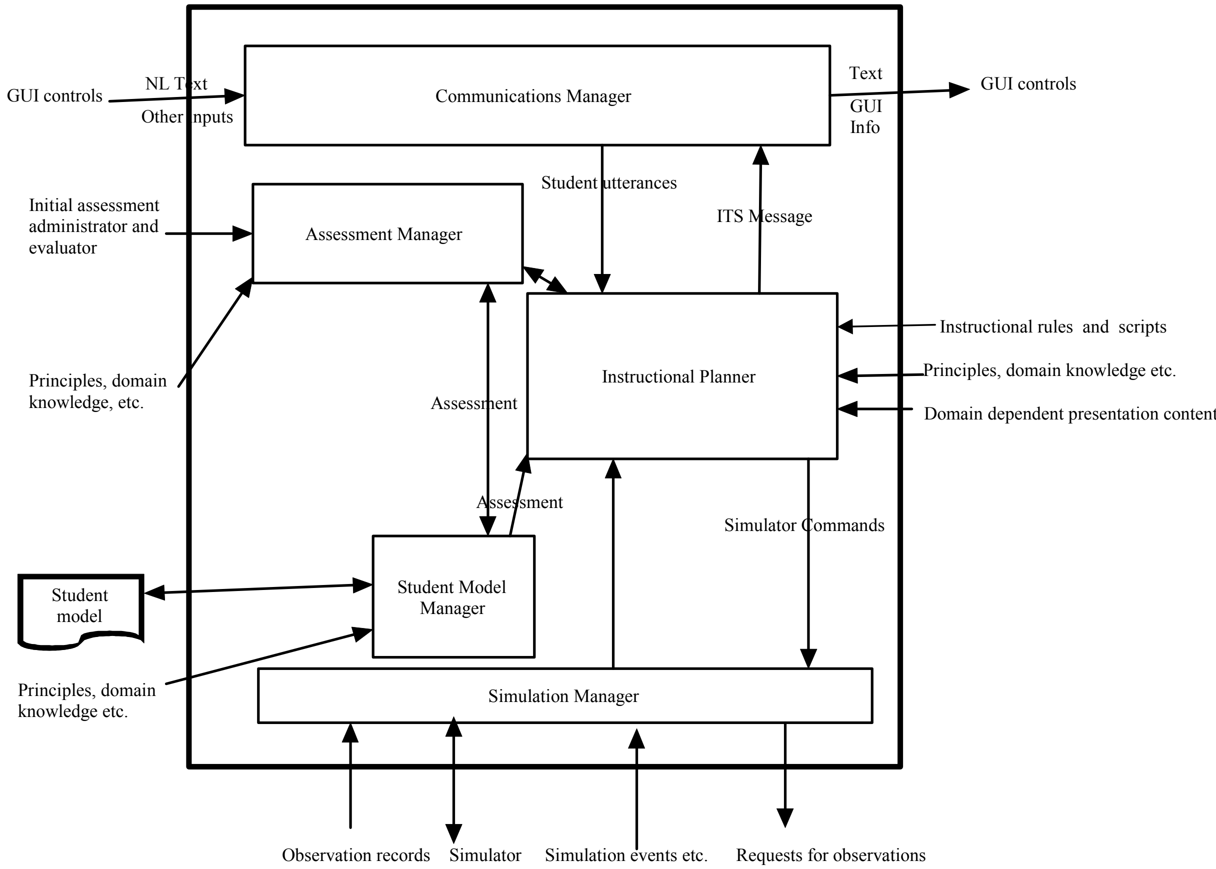


Figure 3: High-level architecture of the ITS Engine

The **Instructional Planner (IP)** uses the student model and a library of instructional rules and scripts to plan interactions with the student to achieve learning and to respond to interactions initiated by the student with appropriate instructional steps. Instructional planning includes planning initial assessment, planning initial lessons, planning responses to student errors within scenarios, determining the amount and kind help to be given to students during a scenario (including visual aids), determining the next task(s) to be assigned to student, determining the sequence of interactions with the student for pre-simulation briefing, determining the sequence of interactions with the student for after action review, providing appropriate feedback, selecting/generating questions to pose to the student, planning responses to student's questions, and sending relevant observations and assessments to the assessment manager.

The IP uses instructional strategies created using an authoring tool; this provides the flexibility of creating courses with varied pedagogical approaches. Instructional strategies are represented in the form of visual scripts. **Figure 4** shows a script that specifies a high-level sequence of instructional actions. Each script is associated with a specific instructional goal. The IP analyzes the student model to formulate instructional goals, and finds and executes instructional scripts that best address these goals. The IP uses rules to select from multiple scripts addressing the same goals.

The **Student Model Manager** is responsible for verifying student identity, for retrieving the corresponding student model, for updating and maintaining the student model, and for saving it when the student leaves the tutor. The ITS monitors the team and each of its members as they perform team exercises. Their performance is evaluated to determine the strengths and weaknesses of the team and its members in terms of the concepts and skills that they have or have not mastered. This information is maintained in the student model which facilitates the customization of instruction to each student. In this case, the student is the entire team. However, it is not enough to model just the team as a whole. Since a team is composed of individuals performing interdependent tasks, the ITS maintains a model of each individual team member, their contribution to the overall

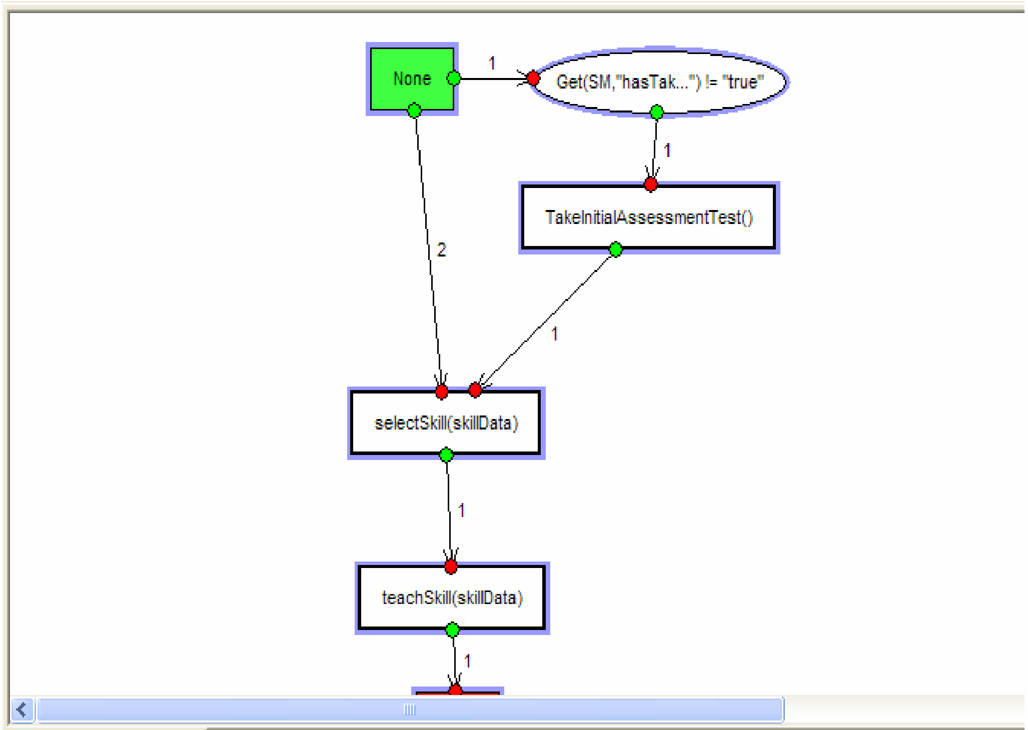


Figure 4: Example of an instructional script

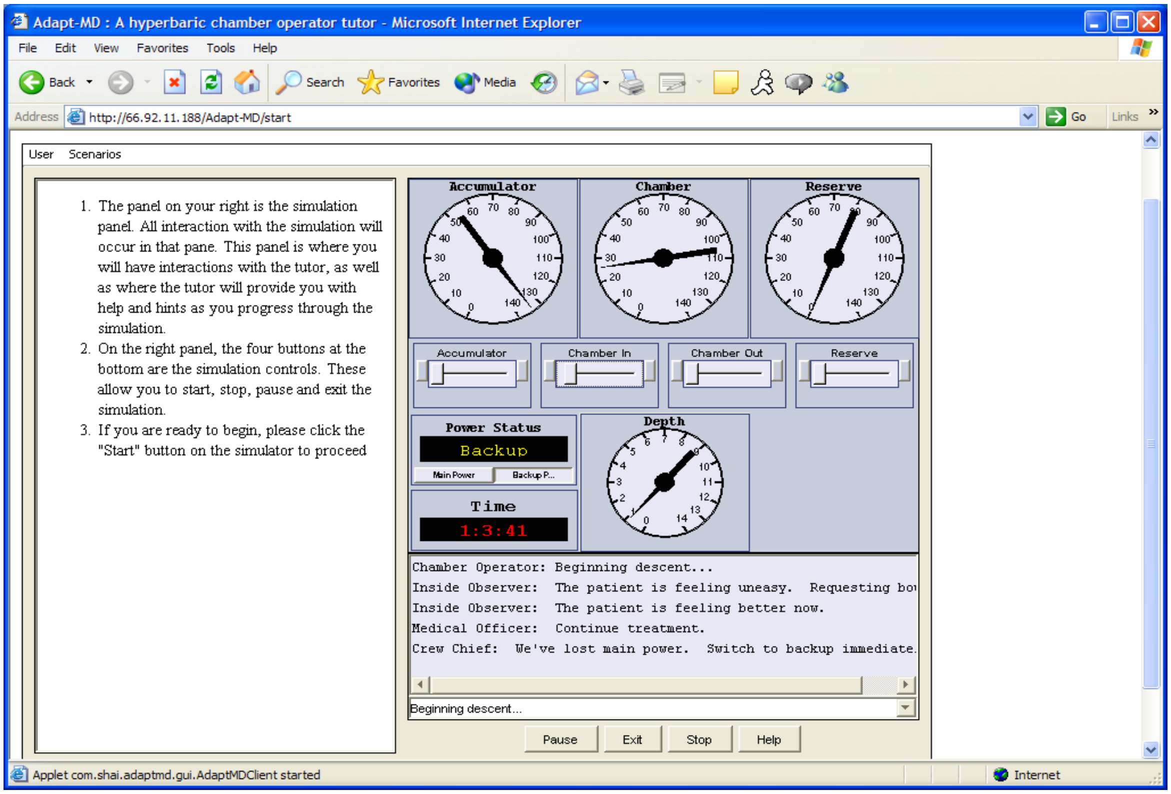
performance of the team, as well as their task-related expertise. In addition, the ITS maintains a model of each team member's knowledge of the other tasks being performed by his teammates. Effective teamwork imposes the constraint that the solution to a sub-problem addressed by a team member should be compatible with the solutions generated by the other team members. Thus, knowledge of the other tasks within the team and the constraints imposed by them is very important. For example, during hyperbaric treatment emergencies such as power loss, the medical officer needs to know how long the chamber can continue to be pressurized before making a decision on continuing treatment. The chamber operator is in charge of determining how long the chamber can be pressured on backup power. Thus, the medical officer and the chamber operator have to communicate and co-ordinate while making a decision to abort or continue treatment in the event of a power loss. Finally, the student model represents each team member's awareness of the proficiency of his teammates at their tasks. Such knowledge is important for adjusting one's own role in the team to compensate for lack of proficiency in other team members. If, in the example above, the medical officer is aware that the chamber operator is new to the task and may have trouble calculating the amount of pressurized air available, the medical officer would need to double-check the information provided by the chamber operator.

The **Communication Manager** acts as an intermediary between the domain-dependent ITS GUI (Graphical User Interface) and the domain-independent ITS Engine, translating the messages between the two into forms that are comprehensible to each. This separation between the GUI and the ITS Engine facilitates the modification of the GUI with minimal modifications to the engine. Thus, different domains can very easily customize the look and feel of the ITS.

Simulator

Training systems developed using the Adapt-MD ITS Engine will need to develop a simulator that is specific to the training domain. Simulations form the heart of the "learning-by-doing" approach embodied by Adapt-MD.

We have developed a prototype simulator for the domain of hyperbaric treatments. **Figure 5** shows a screenshot from the prototype simulator developed for



training hyperbaric treatment providers in emergency procedures. The slide controls in the middle

Figure 5: Prototype simulator for training hyperbaric treatment providers

allow the trainee to control the state of the chamber. The gauges provide information on the chamber state. The chat interface in the bottom right panel enables communication between the trainee and simulated team members. The tutor communicates with the user via the panel on the left. The objective of the simulator is to provide as realistic an interaction as feasible to achieve the benefits of the "learning by doing" approach. With this simulator, the trainee will be able to practice dive procedures in the context of specific scenarios. Notice that the simulator is embedded within a web browser. Currently the simulator is implemented as a Java applet.

In addition to serving as instructional tools, simulations also provide invaluable assessment opportunities. The tutor monitors the actions of each participant during a simulation, assessing his/her performance against a "ground truth" model that specifies the expected performance. In our experience, domains like military decision-making, medical decision-making etc. tend to be more open-ended than traditional ITS domains like Algebra. In the former, there is usually no single sequence of actions that is appropriate for a given situation. Rather, the correct action to perform depends on the current context and the previous actions taken by the participants over the course of the scenario. Moreover, the idea of "correctness" is one of degrees rather than a cut-and-dried classification. This makes for a challenging assessment problem.

Adapt-MD uses Finite State Machines (FSMs) to assess students' actions in a scenario. FSMs are used to represent and recognize context-dependent sequences, in this case, sequences of actions. Each FSM is a sequence of states with conditional transitions between them. The states capture the simulation context and the transitions specify the evolution of the context following particular actions by students. Each scenario is associated with a set of FSMs representing the assessment model for the scenario. Students start out at an initial state with respect to each FSM at the start of simulation. Their subsequent actions take them down a particular path of states and transitions through each FSM. Each state and transition is annotated with information specifying the degree of correctness of students' actions and the contexts created by these actions. The annotations can also assign credit and blame to particular domain knowledge elements like concepts and rules in order to facilitate student modeling. As students traverse the paths, the annotations produce an evaluation of the student's performance. FSMs are a powerful way of representing dynamic behavior and evaluating students in a realistic simulation that does not place too many unrealistic restrictions on what they can or can not do at any point. FSMs, being visual, also facilitate more intuitive authoring by non-programmers. **Figure 6** shows an example of an FSM machine for evaluating a student's proficiency in following emergency procedures in response to a power loss during a hyperbaric treatment. Rectangles represent states and ellipses represent conditional transitions. The green rectangle is the start state. Upon power loss, the machine checks to see if the student has responded quickly enough. If this is true, then the machine checks to see if the student has closed all valves. The machine also provides feedback to the student upon detecting various conditions. This machine only shows the correct actions. Responses to incorrect or unexpected behavior can also be similarly represented. For example, we can specify tutor responses for cases where a student's response is too slow, or where the student fails to close all the valves, etc.

The simulator also provides intelligent entities for interacting with the scenario participants. The hyperbaric simulator includes an automated intelligent patient who reacts dynamically to the participants' actions. Intelligent behaviors can also be attributable to non-living entities like equipment, environment etc. For example, the scenarios can include equipment that fail based on a realistic model of equipment faults. Such behaviors can be created using the FSM representation discussed above.

Extending the idea of automated intelligent behavior further, ADAPT-MD provides a facility for creating simulated team members with intelligent, adaptive behaviors. This expands the applicability of ADAPT-MD to situations involving partial teams or individuals with simulated entities filling in for missing team members. SMEs can create intelligent entities with different models of expertise that adapt to students' proficiency levels. For example, SMEs can create several entities that can fill the same team role at varied expertise levels. An expert (or a "perfect") entity may be chosen for a scenario with a novice trainee where the emphasis is on enhancing the individual task skills of the trainee. As the trainee's expertise level evolves, however, less expert entities may be chosen to provide opportunities for training in team work skills. We are using an existing commercial tool, SimBionic, to create such intelligent behaviors. SimBionic uses the FSM mechanism described above to specify and realize intelligent behaviors for simulated agents. Figure 7 shows a very simple behavior for a simulated inside observer (IO) in a hyperbaric treatment scenario. The shaded rectangle labeled

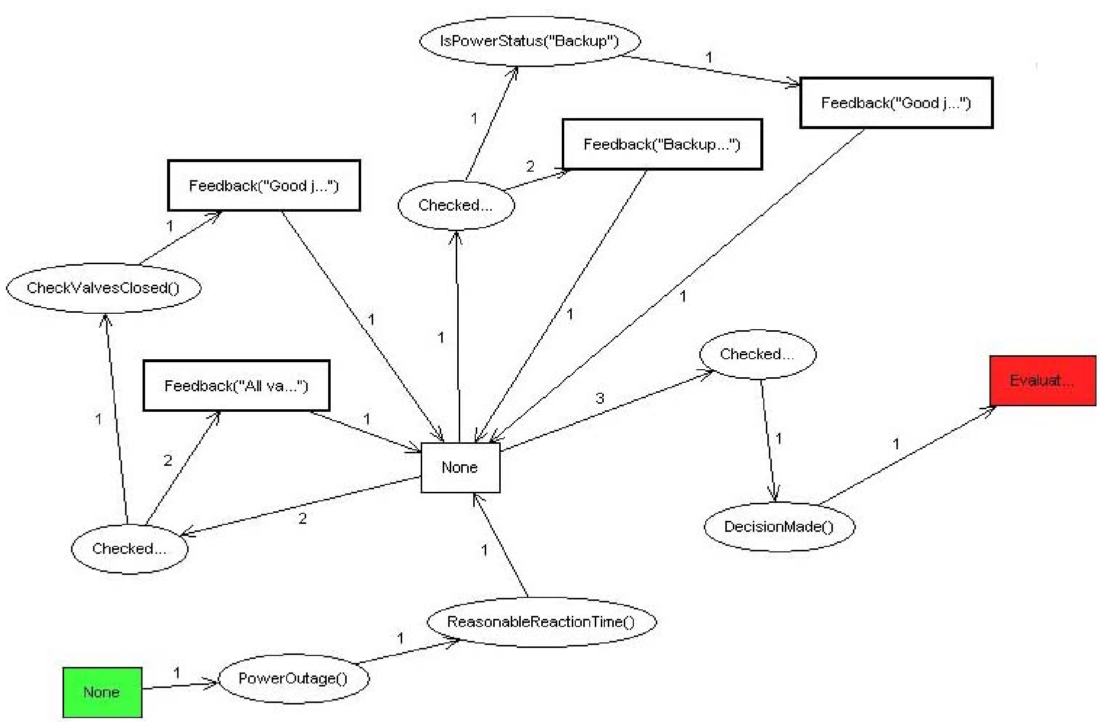


Figure 6: Example of an FSM for evaluating student actions

"None" indicates the start of the behavior. The simulated IO monitors the chamber depth and sends a message to the chamber operator at a specified depth that the patient is experiencing some ear pain. The IO then waits for the chamber to ascend to a certain level at which time it sends a message that the patient is feeling good and the chamber operator can resume the descent. While the example shown is very simple, entities with sophisticated behaviors can be easily created using this environment.

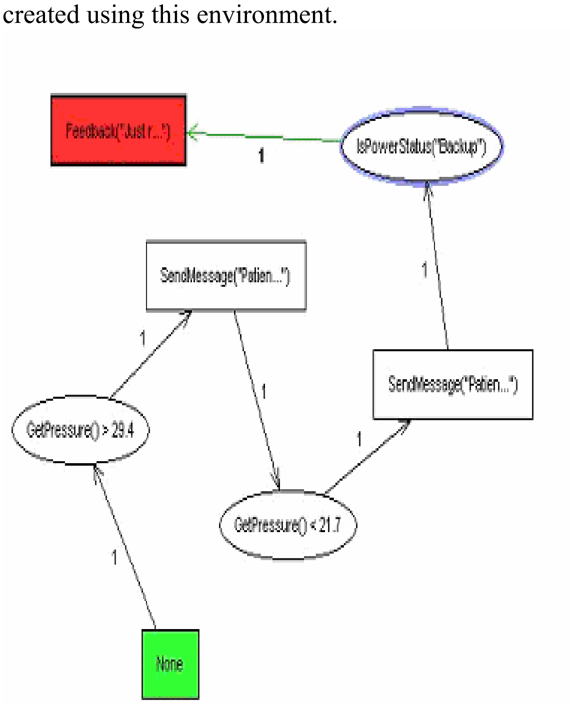


Figure 7: Example of a behavior specification for an intelligent simulated entity.

Authoring Tool

The AIS authoring tool is a collection of pluggable tools that offer facilities for specifying domain knowledge, content, scenarios, and instructional behavior.

The **Domain Ontology Authoring Tool** enables course authors to specify the domain principles, concepts, skills, definitions, etc. at any level of granularity and specify relations between them. Authors are required to specify pre-requisite relationships between domain elements. Authors are free to specify other relationships that can be exploited by instructional behaviors, also specified by course authors.

The **Content Presentation Authoring Tool** allows authors to associate various kinds of didactic or interactive presentations with elements of domain ontology. These are used to provide overviews, reviews, and remediations on specific topics as deemed necessary by the tutor.

The **Scenario Editor and Browser** enables the creation of scenarios and also includes a facility for authors to browse and search the scenario database. Scenarios are associated with specific domain knowledge elements and are assigned complexity levels. This information is used by the tutor to select appropriate scenarios to present to students based on the model of their skill and knowledge proficiencies. Whereas the other authoring tool components described above are domain independent, this tool is domain dependent and a new scenario editor has to be developed for each new domain.

Authors can use the **Questions Editor and Browser** to create interactive questions and discussions to be associated with specific domain knowledge elements. These are be used by the tutor to assess a student's knowledge, diagnose his/her learning difficulties, and provide instruction.

Finally, the **Instructional Behavior Editor** provides a visual interface for specifying instructional scripts and rules for the tutor. This tool uses the concept of Finite State Machines (FSM) to represent the desired tutor behaviors. Recall that FSMs provide a powerful way for specifying context-sensitive behavior with the added advantage of visual authoring. Figure 4 shows an example instructional script authored using this tool.

The Adapt-MD authoring framework is designed to be completely configurable, with the component tools included as plug-ins. Thus, components can be added, removed, or modified selectively based on the needs of a domain. Each authoring component can also be configured to suit the needs of the domain. For example, the domain ontology authoring tool allows course authors to specify ontology types and relationships that are customized to their needs.

Instructional Principles

Historically, ITS research has focused on modeling and representation issues. Recently there is an increased focus on instructional principles, because no amount of technology can replace the effective instructional and tutoring strategies. The instructional principles to be used will vary with the domain of application. The Adapt-MD authoring tool facilitates the creation of new instructional principles for each domain.

For the domain of training hyperbaric treatment providers, we analyzed the course learning objectives and classified them according the categories specified in [Gagne and Medsker, 1996] and have included instructional techniques recommended for each category. We are using the following instructional techniques for the domain of training hyperbaric treatment providers:

1. **Ordered presentation**: The domain concepts, principles and procedures are organized in a network reflecting pre-requisite relationship. The tutor covers topics so that pre-requisite skills are taught first. All presentations of definitions and procedures provide hyperlinks to review materials on pre-requisite knowledge.
2. **Learning-by-doing**: Simulations of realistic medical situations and procedures is a very important part of the instructional strategy adopted by the proposed system. Figure 5 shows the main interface of the simulator prototype for training in hyperbaric chamber emergency procedures. Each simulation exercise is preceded by a pre-exercise briefing that allows the student/team to review key concepts and allows the tutor to further assess the student through questioning. This is a good time to review pre-requisite knowledge and the previous occasions when the student might have attempted similar exercises. The depth and extent of this pre-exercise briefing depends on students' mastery levels and learning styles, with some students being given opportunities for detailed review, and some being allowed to skip ahead to the exercise. At the end of each simulation, the tutor presents an after-action review of trainees' performance, both on an individual and team level. This review also provides opportunities for eliciting justification for student actions and for reflection (e.g. encouraging the student to compare current performance with past performance, encouraging them to articulate what they have learned, etc.). Again the nature of this review depends on the student and team models.
3. **Regulation of challenge level**: ADAPT-MD regulates the challenge level of the exercises presented to students based on information in student models. Exercises should be challenging enough to present learning opportunities and keep the students from getting bored, but not too challenging as to frustrate them.
4. **Scaffolding and fading**: To help a student complete exercises that are slightly beyond their current expertise levels, the tutor provides scaffolding in the form of hints, modeling, job aids, and advice. These are gradually withdrawn as a student's expertise level improves to ensure that they can perform tasks independently.
5. **Instruction of concepts and definitions**: The tutor supports the learning of procedural and decision-making skills by providing instruction on related concepts, definitions and discriminations. Examples, compare-and-contrast examples, and exercises are used to teach this kind of knowledge.
6. **Additional strategies**: The strategies listed above form the core of the instructional approach. Some additional strategies may be incorporated to address meta-cognitive issues like discussions of justifications of student actions, self-monitoring and reflection, adaptations to learning styles, and motivational issues.

RELATED WORK

[Shaw et. al. 1999] have developed an agent-based intelligent tutoring system called ADELE for teaching medical procedures and operations to medical professionals as a part of a continuing education program. This is a simulation-based course and is web-delivered. ADELE presents cases to the students and assesses their performance to dynamically update the student model. ADELE does not address team training, nor does it dynamically assemble courses for students based on their needs.

Several researchers have addressed the cognitive aspects of team performance, and have developed systematic strategies for building effective teams [Klein, 1998; Swezey and Salas, 1992]. Our research draws upon such literature to develop a cohesive model of the factors that contribute to effective team performance and to develop a set of metrics for measuring the effectiveness of a team in a given situation.

[Hsieh et. al. 1999] have developed a domain-independent authoring tool, XAIDA, for generating instruction from specifications of knowledge about device maintenance procedures. This is similar to the proposed system which also presents dynamic courses from specification of knowledge about medical operations. XAIDA does not possess the ability to dynamically construct courses from a vast repository of learning objects.

EVALUATIONS AND LESSONS LEARNED

We have used Adapt-MD to create a prototype ITS for training in procedures related to hyperbaric treatments. We demonstrated the prototype to some instructors and SMEs. The response has been positive with the instructors stating that such a tool would be very helpful in increasing the expertise of novice chamber operators and treatment providers. A large-scale evaluation is planned for the future.

Development of an initial prototype has proved very valuable to this effort. The SMEs involved in the effort, who did not have previous experience with ITSs, could not visualize a simulator-based ITS for this domain and as such were not able to estimate its usefulness. A quick prototype developed within the next few months showing the simulator and some intelligent coaching techniques helped communicate the key features of this approach and led to fruitful discussions of extensions and additional capabilities. The prototype, thus, established a common frame of reference for future communications.

CONCLUSION AND FUTURE WORK

We have described an approach to developing a simulation-based ITS solution to the problem of training military medical personnel in time-critical procedures. We plan to enhance the prototype in several ways. Once the ITS Engine and the Authoring Tool have been developed, we, together with SMEs, will develop instructional content and scenarios for the simulation exercises. We will evaluate the effectiveness of the ITS with qualitative and quantitative studies conducted with student participants. We will also evaluate the usability of the authoring tool through user studies.

Down the road, we also plan to evaluate the generality of the Adapt-MD framework to create ITSs for other medical training domains. The robustness of the instructional planning and student modeling techniques will be evaluated and enhanced.

REFERENCES

Anderson, J. R., Boyle, C., and Reiser, B. (1985). Intelligent Tutoring Systems. *Science*, 228, 456-462.

Bloom, B., S. (1984). The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), 4-16.

Dodds, Philip. (Ed.) (2000). Sharable Courseware Object Reference Model, Version 1.0. http://www.adlnet.org.

Gagne, R. M., and Medsker, K. (1996). The Conditions of Learning: Training Applications. American Society for Training and Development.

Hosek, Susan D., and Cecchine, Gary (2001). Reorganizing the Military Health System: Should There Be a Joint Command?. *RAND Corporation Research Report # MR-1350-OSD*.

Hsieh, Patricia Y., Halff, Henry M., Redfield, Carol L. (1999). International Journal of Artificial Intelligence in Education, Vol. 10, pp. 1-45.

Klein, G. (1998). Sources of Power: How People Make Decisions, The MIT Press.

Nichols,P., Pokorny, R., Jones, G., Gott, S. P., and Alley, W. E. (1991). Evaluation of an avionics troubleshooting tutoring system, *Technical Report, Human Resources Directorate.* Brooks AFB, TX.

Regian, J. W. & Elliott, L. R. (2000). Performance for Distributed Teams: A Cognitive Approach to Team Training and Performance Research (TP-AFRL/HE/AZ/TP 2000 807). Human Effectiveness Directorate, Mesa, AZ.

Schank, R. (1995). What We Learn When We Learn by Doing. *Technical Report no. 60*, Institute of Learning Sciences, Illinois.

Shaw, E., Ganeshan, R., Johnson, W.L., and Millar, D (1999). Building a Case for Agent-Assisted Learning as a Catalyst for Curriculum Reform in Medical Education. In *Proceedings of the Int'l Conf. on Artificial Intelligence in Education*, July, 1999.

Swezey, R., and Salas, E. (1992). Teams: Their Training and Performance, Ablex Publishing Corporation.