Scenario-Based Conversational Intelligent Tutoring Systems for Decision-Making Skills

Eric A. Domeshek
Stottler Henke Associates, Inc.
Cambridge, MA
domeshek@stottlerhenke.com

ABSTRACT

U.S. warfighters are being asked to work in ever more complex operations and environments, where everyone from the general officer to the "strategic corporal" must engage in critical reasoning and creative thinking. Advanced military professional schools often address such skills, adopting a virtual apprenticeship or mentorship approach: students analyze past cases and work through problems while an expert practitioner probes their thinking and models good practice. As more of our military requires such skills, it is desirable to make this labor-intensive form of education readily available to a broader military audience.

The emerging technology of Intelligent Tutoring Systems (ITS) has successfully provided computer-based training with automated individualized assessment and instruction across a range of procedural and reasoning tasks. Scenario-based techniques can potentially allow ITS construction even when computerized expert performance models cannot be built. However, it remains difficult to simulate interaction with a cast of simulated characters, or to duplicate the Socratic probing of an experienced instructor.

We describe an approach to construction of scenario-based ITSs that (1) supports a range of interaction styles, including simulated discussion with scenario characters and an automated Tutor, encompassing both student-initiative and agent-initiative dialogs, (2) is supported by an authoring style that lowers per-scenario costs by encouraging behavior reuse and relying on an extensible rule language mapped from templates expressed as conventional spreadsheet data, and (3) delivers its sophisticated interactive simulation over the web using standard browsers without plugins.

An initial application to Chemical, Biological, and Radiological medical training was evaluated by a panel of 17 practicing emergency room physicians who played through a scenario and rated 11 key aspects of the system on their expected instructional effectiveness; ratings averaged 3.9 on a scale from 1-5. This same technology has been used to prototype training for interagency collaboration in stability and reconstruction operations.

ABOUT THE AUTHORS

Dr. Eric A. Domeshek is an Artificial Intelligence (AI) Project Manager at Stottler Henke Associates, Inc. He received his Ph.D. in Computer Science from Yale University, where his work focused on cognitive modeling and technology, most especially on development of Case Based Reasoning (CBR). While working as Research Faculty at the Georgia Institute of Technology, he helped launch the EduTech Institute, and became involved in educational applications of AI and CBR. He continued to work on educational and training technology while on faculty at Northwestern University's Institute for the Learning Sciences. For the last ten years, Dr. Domeshek has conceived and managed a variety of Intelligent Tutoring System (ITS) projects at Stottler Henke. His training research currently focuses on dialogue-oriented tutors such as the Enact tools described here.

Scenario-Based Conversational Intelligent Tutoring Systems for Decision-Making Skills

Eric A. Domeshek
Stottler Henke Associates, Inc.
Cambridge, MA
domeshek@stottlerhenke.com

PROBLEM

U.S. warfighters are being asked to work in ever more complex operations and environments, where everyone from the general officer to the "strategic corporal" must engage in critical reasoning and creative thinking. Advanced military professional schools often address such skills, adopting a virtual apprenticeship or mentorship approach: students analyze past cases and work through problems while an expert practitioner probes their thinking and models good practice. As more of our military requires such skills, it is desirable to make this labor-intensive form of education readily available to a broader military audience.

Within the U.S. military there is a tremendous ongoing commitment to ensuring our forces are the best trained in the world. Training and training development for all branches and all specialties is generally ongoing. Training objectives span a range of complexity from component skills, to operational skills, through higher-level decision-making, and on to effective team operations. Training mechanisms run the gamut from self-study materials, to distance learning, live courses, individual and group simulations, and ultimately live exercises of varying scales. Computer-based simulations have constituted a growing segment of the training spectrum, as massive investment and consequent technological advances have raised capabilities and lowered costs. Simulations are now widely used across the services to provide increased opportunities for practice and training at decreased cost.

However, the investment in technology for simulation-based training that has proven so useful in the combat branches has not always been extended to traditional support areas such as medical services, or to the recently emphasized non-kinetic Stability, Security, Transition and Reconstruction (SSTR) operations. Beyond DoD, the realm of Homeland Security has emerged as another set of areas where government must develop effective training. Teamwork and decision-making for these kinds of tasks requires just as much training as is devoted to command of tactical units. Performance on these tasks, too, benefits from extensive experience, expert coaching, and insight-producing after-action reviews; practitioners must learn to see the factors that should affect their decisions, and prepare for likely follow-on consequences, including the potential action of adversarial forces.

It is also largely recognized that simulations, by themselves, are not particularly useful as training; it is the coupling of simulators, with appropriately crafted scenarios, and expert coaching and feedback that provide the greatest benefit. However, the need for expert supervision drives up the cost and limits the availability of effective training. In response, much work has been devoted to coupling individually responsive automated instructional capabilities with simulations. The goal is to continue to improve the cost/benefit equation by minimizing the need for human observer/controllers in simulation-based training. Intelligent Tutoring Systems (ITSs) are an emerging technology that puts a simulated instructor in the computer box along with the simulated world (Ong & Ramachandran, 2000). The level of ambition and proven capability for ITSs has also been increasing, and research is pushing practical ITS tools from the level of straightforward procedural tasks (e.g., Munro & Pizzini, 1995) or closed-world formal reasoning tasks (Anderson, et al, 1985) to more open-ended analysis and decision tasks such as tactical decision-making (Domeshek, Holman, & Ross, 2002).

Despite progress, building affordable and effective ITSs for complex decision-making and team-coordination tasks remains a research problem requiring unique combinations of specialized expertise. Such an ITS, for instance calls for a careful and novel combination of simulation components and training scenario design. We cannot simply let a simulator run free, as the simulated world could easily enter states that are either not instructionally relevant, or which the system is not prepared to critique and tutor. Since there is no validated algorithmic approach to carrying out the kinds of jobs we aim to teach, we cannot apply the

common ITS approach of building a fully competent expert system, and then using that system to structure an "overlay" model of student competence, able to track good and bad performance on a wide range of generated problems. Instead, we have to characterize islands where general principles drive team members' behaviors, and supplement those with scripts and contextually-bound assessments designed to teach specific points in specific scenarios.

Problems with coverage actually exist on the simulation side as well as the tutoring side. Often no single validated simulator can accurately generate all world states that might be relevant to a training audience in complex domains such as medical response, SSTR operations, and Homeland Security. For instance Smith (2003) notes that *many* simulators already exist with relevance to Homeland Security and Emergency Medicine—all with different assumptions, foci, and limitations. We must be prepared to fall back on scripted events to maintain focus on pedagogically useful situations and assessable sequences of actions.

Finally, designing scenarios that exploit what is known about good pedagogy at this level, while building efficient effective automated behaviors and instruction, remains a challenge. Moving from individual training to individualized training in the context of multi-role teams brings further challenges, as well as opportunities for scaling existing and planned technologies.

SOLUTION

In this paper, we draw our primary examples from work we have carried out on scenario-based conversational ITSs for emergency medical decision-making in the context of civilian hospital-based teams confronting Chemical, Biological, and Radiological (CBR) contingencies. For the resulting Medical Emergency Team Tutored Learning Environment (METTLE), key issues include (a) diagnosis of early cases potentially leading to discovery of the existence or nature of a CBR event, (b) treatment of individual casualties of a CBR event, and (c) more systemic responses to the recognition of a CBR event directly affecting future medical operations.

We set out to develop training that can provide medical professionals (military and civilian) with extensive simulated practice and coaching on decision-making required to deal with medical emergencies (e.g., CBR attacks). Our goal was to make this training widely, easily, and cheaply available to the large number of professionals who might find themselves responsible for contributing in different roles to a coordinated response to such a situation.

Given the current state of technology, developing a web-hosted simulation-based ITS was among the most promising approaches to this problem. From a centrally administered server (or distributed family of servers as needed), users can cheaply and easily access such training, wherever they may be and whenever they have time. Likewise new insights regarding possible threats or approved response doctrine can drive creation or modification of training scenarios, and such updates can be deployed on the servers, becoming immediately available to the whole student community.

To ensure the widest accessibility of such training, the components that run on the student's client machine (in its web browser), should avoid making unnecessary assumptions about system capabilities. In particular, they should adhere to the most common (web) standards, minimize demands on network bandwidth, refrain from placing undue loads on client memory or processing, and function well with standard input/output capabilities. To enable effective training whenever an individual student has time, scenarios should be populated with simulated agents playing the roles of other team members. Variants of scenarios can be constructed to focus on the decisions and interactions appropriate to particular roles in the overall medical response.

No matter the role, the focus on complex decision-making in multi-character contexts dictates that student interaction with the system be characterized in large part by extensive dialog—both with simulated teammates and with the embedded Tutor. When decision-making is the focal task in a team environment, the interactions among team members are mostly about information exchange, task coordination, and responsibility allocation—all requiring extensive communication. When contextual judgment is the focal learning objective in a tutoring environment, the interactions between Student and Tutor most effectively dwell on decision rationale—requiring the teasing out of factors through ongoing (often Socratic) dialog.

We built a CBR medicine ITS—including runtime, authoring tools, and distribution environment—to satisfy the constraints above. We then abstracted the key ideas and techniques that enable this system into a more general framework. We use the acronym SPIRIT to summarize the key components of the training simulation approach enabled by this framework: *Scenarios*, *Principles*, *Issues*, *Roles*, *Interactions*, and *Tools*:

- **Scenarios:** Scenarios are constrained simulated experiences designed to raise particular issues and teach particular principles. The constraint on the experience comes from the initial simulation conditions, from consequent (conditionally) scripted events, and from biases in the responses of simulated agents. Within these limitations, an underlying simulator may calculate the world's evolution, perturbed by Student actions. For instance, a scenario might be designed to emphasize (among other issues) the consequences of appropriate or inappropriate decontamination or isolation regimens following a CBR attack.

- **Principles:** Principles are the main points a system is designed to teach. Principles may be generalizations, often at a level that might be characterized as "control knowledge" for application of a skill (e.g., in case of suspected biological attack, once you have identified a likely pathogen, be sure to check again for other possible agents that might have been combined with the one you have identified), or the principles may consist of more specific contextually bound knowledge (e.g., when dealing with inhalation anthrax, expect an incubation period of 1-7 days, but allow for the possibility of a period of up to 60 days).

- **Issues:** Issues characterize key choice-points in the decision-making required by a scenario. As such, they often serve as organizers of principles—the knowledge, skills, and control required to reasonably address the issue. Example issues might include when to order a particular diagnostic test, whether to place patient in isolation, or whether to recommend prophalactic treatment related to a suspected threat.

- **Roles:** Our approach is intended to train individuals in tasks where they must act as part of a team. We assume the team is not homogeneous—that is, different team members have different roles and thus different responsibilities (different issues, subject to resolution by different principles). Example roles include first-responder, front-line care-giver, consulting specialist, long-term care-giver, incident manager, resource planner, etc.

- **Interactions:** In our medical application domain, individuals cannot do their jobs without interacting with other characters. As noted, we developed computer-simulated agents to play Patients and team members, for convenience, flexibility, and focus. Those agents must support interactions where Student actions can elicit decision-making cues.

- **Tools:** Supporting appropriate actions typically involved offering Students an appropriate set of tools. Our example user interface provides a simulated Patient chart; various order forms, and a visible Patient to support examinations. A combination of dialog (interactions), multimedia presentations, and interactive visualizations is, in general, required.

The same approach and machinery have also been used to prototype a training scenario for SSTR operations. In that prototype the Student plays the role of a member of a Provincial Reconstruction Team (PRT) in Afghanistan. There the focus is on Student interactions with simulated agents who represent members of the host-nation, interagency, and international community. In the sample scenario, the Student's job is to gather information to support the evaluation of a set of possible courses of action regarding construction of school facilities.

DESIGN

In this section we attempt to give a sense for the framework and its application—both from the Student's point of view, and from the perspective of the underlying machinery.

The Student's Experience

Figure 1 is a (partial) screen shot from our medical simulation. It shows the half of the main screen that focuses on interaction with simulated characters. In this case, the Student has selected the *Patient* from among the five character icons at the top of the screen. A picture of the Patient and the Student's most recent verbal exchange with that character take up much of the upper part of the screen (character utterances can also be delivered as recorded sound files). Options for interacting with the Patient appear in the lower part of the screen. In particular, the left margin has a set of buttons that pop up other windows supporting non-verbal interaction, while the main area contains a type-in box and a set of checkbox options offering possible interpretations of what has been typed.

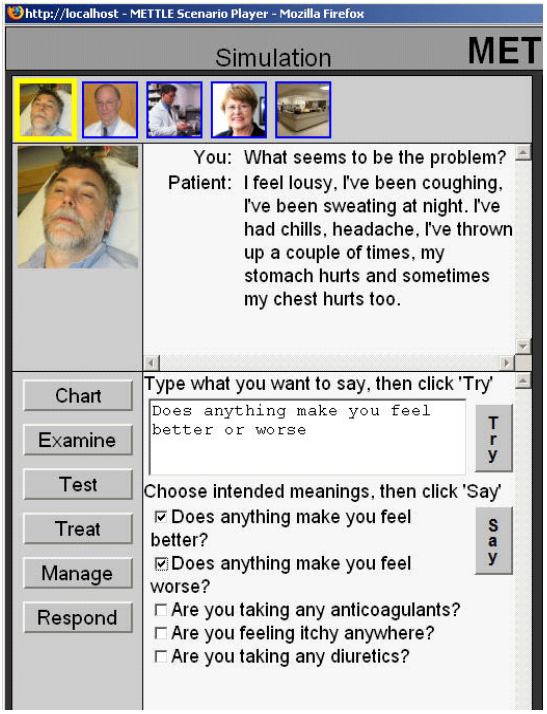

Figure 1. Character Interaction.

Figure 2 shows the other half of the main screen, which supports interaction with the simulated Tutor. The sequence of "Hint?" "What?" "How?" and "Why?" buttons are available as appropriate to let the Student request guidance from the Tutor. The Tutor can also offer advice or feedback without being asked by the Student, based on actions the Student takes or fails to take (and the evolving scenario context).

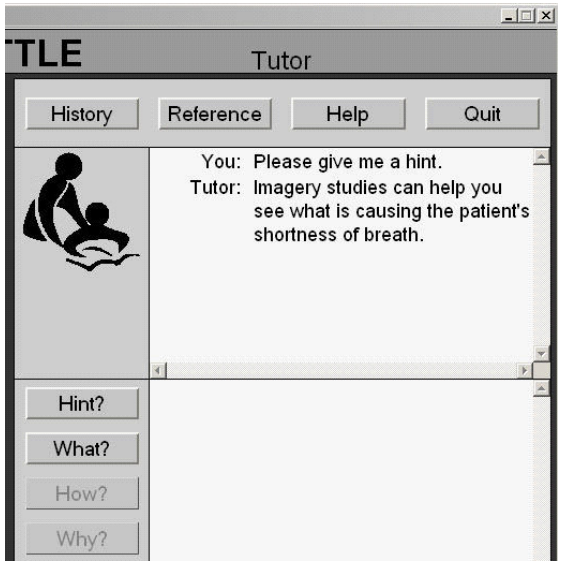

Figure 2. Tutor Hinting.

Figure 3 shows the Tutor leading the Student through a piece of a Socratic dialog aimed at exploring their evolving diagnosis. Often in such dialogs, as here, the fact that the Tutor is asking the questions means that Student responses come from a known class which can be enumerated (generally based on an underlying conceptual representation of the domain).

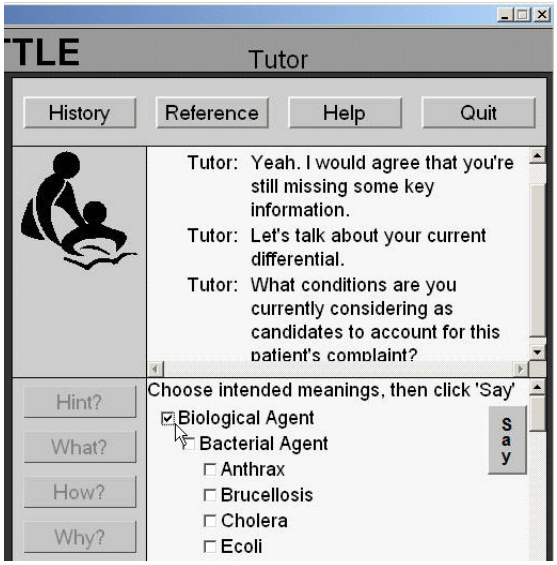

Figure 3. Tutor Dialog.

The screens above show several different modes of simulated conversation: Student initiative and Tutor initiative, text-based and multiple-choice. The system also supports other kinds of interaction. Figure 4 shows a use of interactive graphics to simulate aspects of a physical examination. Hot regions on the left-hand full-Patient images can be clicked, which produce close-up images on the right, as well as summarized findings for the selected region at the bottom.

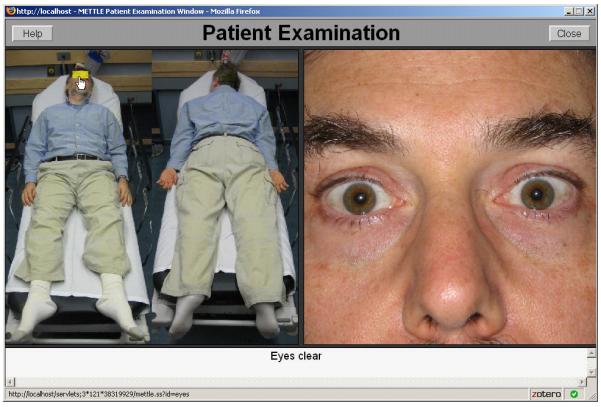

Figure 4. Physical Examination.

Figure 5 shows an example of one of the system's order forms. This form allows for ordering a certain class of tests. There are a variety of test order forms, and likewise forms for ordering various kinds of treatment (e.g., administration of antibiotics), as well as other forms for common hospital actions (e.g., Patient management and emergency room management).

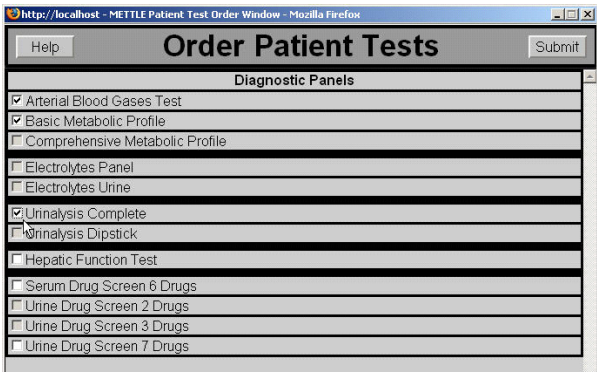

Figure 5. Test Order Form.

Figure 6 shows a piece of the simulated Patient chart, in this case reporting the results of one of the tests ordered in Figure 5.

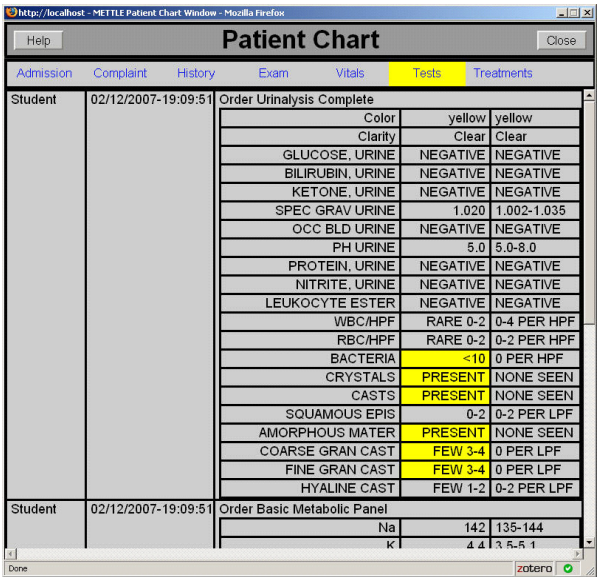

Figure 6. Patient Chart.

Behavior Authoring

There is much to say about designing and constructing training scenarios of the sort suggested here. However, we will limit our attention to techniques for authoring simulated agent behavior. Earlier scenario design steps require us to identify the scenario Patients, their conditions, key learning objectives, cast of characters, scene structure, important states, and major aspects of scenario flow.

The ITS framework described here adopts a *theater* metaphor when structuring agent behaviors. Each *character* is assigned a *script*, composed of *lines*, which specify behaviors they should exhibit. Since a simulation is interactive, the lines are not simply executed in sequence from beginning to end. Rather each line is specified with a *cue* that tells the agent when to carry out the *action* specified in the line. Cues typically test for things the Student has said or done. Scenarios scripts may also be divided into *scenes*.

To facilitate behavior authoring, agent scripts may be assembled by merging pieces that are general to the character, pieces that are appropriate to the entire scenario, and pieces that only apply during a particular scene. If there is an important class of character likely to recur across a set of scenarios, it may make sense to define (parts of) commonly recurring behaviors in a reusable way. In our medical system we treat the Patient in this way, and pre-define the cue conditions for roughly 550 conventional behaviors—things like recognizing when the Student has asked any of about 300 diagnostic interview questions, carried out physical examination actions, or ordered any of a battery of tests.

In our anthrax scenario the only Patient behaviors that were created totally de-novo were those dealing with questions about the specific acquaintances who were also sick, and those dealing with scenario flow control (e.g., moving among scenes); in all, it came to about 20 totally novel behaviors. This is in contrast to the roughly 550 conventional behaviors built on Patient defaults. Authoring the customized responses to the default questions requires some mild creative writing combined with a solid medical understanding of how a Patient with the target condition would likely respond. For any target condition, the vast majority of the questions are either irrelevant or present opportunities to introduce pedagogically useful distractions. As one example, in early scenario design discussions we considered the possibility of having our Patient be on drugs for mild schizophrenia, simply because there is a syndrome associated with such drugs that presents with some similarities to anthrax.

Our primary means for authoring script lines was a pre-formatted spreadsheet. We prepared a spreadsheet with reusable Patient behavior cues. For a first pass at authoring a new scenario, the author need only run down the list of such behaviors and enter the particular

simulated Patient's responses (i.e., the text of what the patient would say, along with optional recordings of those utterances), the images and findings to be associated with physical examination actions, and the test-result HTML files to be merged into the Patient chart in response to particular tests being ordered.

Totally scenario-specific behaviors require a bit more effort to specify, since the cues have to be entered as well as the responses; still custom interview question are not terribly hard to create. As an example, we show the cue and response for the custom behavior that watches for a query about our anthrax Patient's last contact with their cousin:

```
(or   (hear "When did you last see
             your cousin?")
      (hear "How long ago were you
             last with your cousin?")
      (hear "How many days has it
             been since you spent
             time with your cousin?")
      )
(say "We went to a basketball game
      together with another friend of
      mine maybe 5 or 6 days ago.")
```

Tutor Behavior Annotations

In the framework being described here the bulk of Tutoring is specified as annotations on the behaviors of other simulated characters—those behaviors critical either to learning or to maintaining progress through the scenario. Generally only a subset of all agent behaviors bear directly on course curricular objectives and the critical flow of any particular scenario; if there are large sets of pre-defined recurring behaviors (as for Patients), the relevant subset in any given scenario is likely to be quite small. Important behaviors are annotated with curriculum points, Tutor comments (proactive prompts, available hints and explanation, and reactive feedback), and relevance conditions.

While there may be very little in the way of strict procedures that a Student should be following, there are generally conditions determining the relevance or importance of their taking particular actions. These conditions should be translated into observables in the simulation environment (most frequently into checks on the scenario history to see whether or not the Student has yet taken some actions) and turned into Tutor tests. When does it become relevant to order basic blood (cultures, gasses, and metabolism panel) and urine tests? Pretty much as soon as the Patient is set up in the ED and an IV established. When should a test of CSF be ordered? For our anthrax scenario Patient, basic test results should suggest the possibility of meningitis and provide motivation for such tests, but even then, CSF should not be collected until a head CT has been reviewed. Such tests can be written in terms of combinations (`and`, `or`, `not`) of `line` test operations.

Given behaviors worth discussing and the conditions under which those behaviors are appropriate, we provide for seven different kinds of Tutor utterance annotations that fall into three categories:

- **Proactive Prompts:** When defined, a `pro` Tutor utterance will be output at some specified time after the relevance conditions for the behavior become true.

- **Hints/Explanations:** When defined, a cascade of `hint`, `what`, `how`, and `why` utterances (not all of which need be provided) will be offered to the Student (by enabling the corresponding Tutor buttons on the Player page) after the relevance conditions for the behavior become true. A limited set of hints will be offered at any one time, with the Tutor rotating among those associated with the highest ranked active behaviors.

- **Reactive Feedback:** When defined, a `yup` or `nope` utterance will be output depending on whether, (for `yup`) the Student triggers the associated behavior after its relevance test becomes true and optionally before the `when` time expires, or (for `nope`) the Student triggers the behavior either before it is appropriate, or optionally after the `when` time expires.

From an authoring perspective, the questions are: (1) which kind of utterances to associate with which behaviors, (2) what delays/time-limits to specify, and (3) what ranks to assign. In general, we tend to use proactive prompts (often combined with hints/explanation on the same or related behaviors) to try to ensure that Students don't get bogged down and fail to make progress because they miss out on some critical step in the scenario. We use hints/explanations relatively frequently, discussing many actions that a reasonable physician might take at various stages of the scenario, often explaining why they would be reasonable in the context (given the associated Tutor test is true). We use reactive feedback fairly sparingly to highlight important steps or missteps; our assumption is that physicians do not want to receive too much congratulation or critique from a machine.

EVALUATION

This section reports an evaluation of our initial application for Chemical, Biological, and Radiological medical training

Evaluation Method

We identified a random selection of emergency physicians from on-line sites, and mailed out letters soliciting their participation in our Evaluation Study. The letter explained the nature of the study, the amount of time expected (up to 2 hours), and offered $100 compensation for their time if they participated. In response to 2500 such letters[1] we ultimately received indications of interest from 55 physicians[2] in the form of emails to a special address we had set up for the purpose.

With a goal of 20 respondents, we mailed out 30 evaluation packets. Each packet contained (a) a cover letter explaining the details of the evaluation process, (b) a CD ROM containing the system's Server software with supporting video introductions to its use and related browser configuration issues, (c) an Evaluation Feedback Form, (d) a Payment Sheet, and (e) a stamped return envelope.

The materials in the packet also included email and telephone contact information for use in the event the physicians ran into difficulty with the software. We ended up exchanging support email with 5 physicians, and speaking with 2 of those 5. We identified 4 issues that led to technical difficulties, the first 3 of which we were aware of and had tried to account for in the instructions we distributed: (1) the Server needs to be copied from the CD onto the user's hard drive so that it can write intermediate files as it runs, (2) the software is only compatible with relatively modern browsers including Internet Explorer v7 and Firefox v2, (3) the browser must have "pop-up blocking" turned off for the software to work, and (4) one user reported incompatibility with Windows Vista Ultimate edition based on a conflict or restriction related to use of the standard web server port 80. To address issues 1 and 4, when we sent out our second wave of evaluation packets, we offered subjects the option of *not* installing the Server, but rather running off of a server hosted in our offices.

[1] There was a 10%-15% rate of returned undeliverable solicitation letters due to "bad" addresses.

[2] This figure includes what appears be a cluster of perhaps half-a-dozen word-of-mouth referrals within one medical school.

Evaluation Results

We received responses from 17 evaluation subjects. Table 1 presents the data we collected characterizing our subjects' experience in medicine and medical training. In this and all following tables, the bottom two rows show the average and standard deviations of the subject data in the body of the table. Here, the first column contains number of years in the medical field, post-bachelors; we had subjects who were relatively freshly out of medical school as well as veterans, with the average at a healthy 16 years of experience. The other five columns requested subject self-assessment experience ratings on a scale of 1 ("not experienced") to 5 ("very experienced) bearing on several aspects of medical practice, education, and training technology.

Essentially all subjects rated themselves highly on experience in emergency medicine. There was more diversity in their experience developing and delivering medical education/training (not always correlating with seniority in the field), averaging out in the middle of the scale. Interestingly, *use* of Computer-Based Training (CBT) also averaged out in the middle of the scale; it seems that some forms of CBT have become reasonably wide-spread in the medical field. On the other hand, significant experience with *development* of CBT was rare (though half the subjects claimed *some* experience in that area).

Table 1. Subjects' Levels of Experience

Subject	Time in Field (Years)	Emerg. Medicine (1-5)	Develop Training (1-5)	Deliver Training (1-5)	Using CBT (1-5)	Develop CBT (1-5)
1	24	5	4	4	3	1
2	16	4	2	2	2	1
3	31	4	1	2	3	1
4	12	5	5	5	3	2
5	3	5	3	4	3	1
6	24	5	4	4	5	3
7	30	5	4	4	3	3
8	19	4	1	1	2	1
9		4	4	3	4	2
10	8	5	3	4	2	1
11	6		4	4	4	4
12	1	4	1	1	3	1
13	26	5	2	2	2	1
14	11	5	3	3	3	3
15		5	2	5	5	1
16	6	5	3	4	3	3
17	24	5	4	3	3	2
Mean	16.07	4.69	2.94	3.24	3.12	1.82
StdDev	10.07	0.48	1.25	1.25	0.93	1.01

Table 2 presents data we collected on the amount of time subjects spent on the evaluation. These values are fairly homogeneous and reflect reasonably well our expectations and advice, with a few notable exceptions (primarily a couple of outliers on the high side for set-up time). The most important information here is that

subjects did spend significant time with the simulation, averaging out to just a bit more than an hour. We expect that the high numbers for Set-Up (e.g., installation of the software) and Video & Help (e.g., pre-scenario orientation to the software's use) reflect relative inexperience and/or insecurity with computer use (and perhaps difficulty with some of the issues described in the Method section above). In any case, in a real fielded version of such a system, individual users would not be asked to install their own Servers.

Table 2. Time Spent on Evaluation Tasks

Subject	System Setup-Up (Mins)	Video & Help (Mins)	Scenario Run (Mins)	Results Write-Up (Mins)
1	10	15	45	5
2	10	10	45	5
3	30	30	120	10
4	10	5	60	10
5	20	15	45	5
6	15	10	60	3
7	60	30	60	10
8	5	5	50	5
9	15	10	90	10
10		5	60	15
11	10	15	60	5
12	90	5	75	10
13	30	10	60	10
14	30	30	60	10
15	25	20	60	10
16	20	15	90	10
17	30	30	60	7
Mean	25.63	15.29	64.71	8.24
StdDev	21.75	9.43	19.40	3.11

Table 3 starts to get at the instructional effectiveness of our approach to medical training. We asked subjects to rate how the approaches illustrated in our prototype were likely to compare on a range of training objectives against more traditional approaches such as attending lecture or reading articles. Again these data are ratings on a 1 to 5 scale, this time ranging from "Much less effective" to "Much more effective" (higher numbers are better). A value of 3 was anchored to "As effective" as these traditional methods of instruction. The four objectives we surveyed included (a) exposing students to uncommon situations with accompanying knowledge and skills, (b) allowing students to practice applying such knowledge and skills, (c) acquiring proficiency at emergency medical response, and (d) identifying gaps in knowledge and skills.

Almost all subjects rated the simulation as likely to be at least as effective as traditional instructional methods with regard to all four learning objectives. Subject 7 was an outlier in this regard, and Subjects 13 and 16 also each assigned one dimension a relatively low rating of 2 (note that Subject 2 provided *no* ratings whatsoever). The averages come out between 3.7 and 4.1. *Practicing* use of knowledge and skills and *identifying gaps* come out slightly on the higher side. *Exposing* students to uncommon situations and achieving *proficiency* come out slightly on the lower side. There is some logic to these relative rankings, as a simulation is good for practice, and one side-effect of (critiqued) practice is to recognize where your gaps are. On the other hand, it may be that readings and lectures are just as good for simply exposing students to odd cases. Finally, we suspect fidelity limitations of the simulation may lead subjects to hedge about how effective it can be at helping students achieve proficiency in practice.

Table 3. Subjects' Ratings for Effectiveness of Instructional Method

Subject	Expose to Uncommon Cases, Knowledge, & Skills	Practice Applying Knowledge & Skills	Gain Proficiency at Skills	Identify Knowledge & Skills Gaps
1	3	4	4	4
2				
3	4	4	3	5
4	5	5	5	5
5	4	5	5	5
6	3	3	3	3
7	3	2	2	2
8	4	5	4	5
9	5	4	4	5
10	3	4	3	4
11	4	3	3	3
12	4	4	3	3
13	3	3	2	4
14	4	5	5	5
15	4	5	5	5
16	2	4	4	3
17	4	4	4	4
Mean	3.69	4.00	3.69	4.06
StdDev	0.79	0.89	1.01	1.00

Table 4 reflects data from a final set of questions intended to dig deeper into the particular aspects of our prototype that might account for different kinds of instructional effectiveness. We asked subjects to rate seven different aspects of the system, again on a scale of 1 to 5, this time ranging from "Not effective" to "Highly effective" (again, higher numbers are better). The seven system dimensions cover general assessment of scenario structure, four specific aspects of the simulation, and two aspects of automated Tutor support. Again ratings cluster around a value of 3.9.

We find it encouraging that the overall scenario structure gets a solid 4+ rating, and that the two aspects of the Tutor come out at 3.9. The lowest ratings are associated with areas that had known limitations. Feedback during our formative evaluation after the 2nd development cycle identified limits on the extent to which a low-fidelity, non-immersive simulation could support cues and actions a doctor might use in the real

world. In our subsequent design and implementation we fleshed out the Patient behaviors within those limitations (though clearly we could still do more, e.g., in terms of adding sounds and more useful visuals to the Patient examination). However, we recognized that we were not able to push as far on refining the universe of student actions beyond the diagnosis phase. Likewise, we believe we could push further on the extent to which interactions with other actors are fleshed out (currently our lowest score at 3.7).

Figure 7 summarizes all of the ratings data presented above as a series of ratings value bar charts to emphasize the distribution of subject responses. The top row of four graphs shows the *relative effectiveness* ratings. The bottom two rows (comprising seven graphs) show the *specific feature effectiveness* ratings. The clustering of most values around a rating of 4 is visually apparent. The skewing of the Tutor ratings (the last two graphs in the bottom row) to the high side is also suggested.

Table 4. Subjects' Ratings for Specific Aspects of Training System

Subject	Scenario Challenges Knowledge & Skill Application	Patient Provides Decision Cues	Chart Organizes Data & Supports Decisions	Other Actors Provide Decision Cues	Simulator Interaction Allow Student to Show Skills	Tutor Explores Rationale & Decisions	Tutor Feedback Promotes Learning & IDs Gaps
1	4	3	3	2	2		3
2							
3	5	3	4	3	4	3	4
4	5	5	5	5	5	5	5
5	5	5	4	5	5	5	5
6	3	3	3	3	3	3	3
7	3	3	3	3	4	4	5
8	4	4	4	4	3	5	5
9	5	4	5	5	4	5	4
10	3	2	2	3	3	3	3
11	3	4	4	4	4	4	4
12	4	3	5	3	4	4	4
13	4	4	5	3	2	2	1
14	4	5	4	3	4	3	4
15	5	5	5	5	5	5	5
16	4	3	4	4	4	3	4
17	4	4	4	4	4	4	4
Mean	4.06	3.75	4.00	3.69	3.75	3.87	3.94
StdDev	0.77	0.93	0.89	0.95	0.93	0.99	1.06

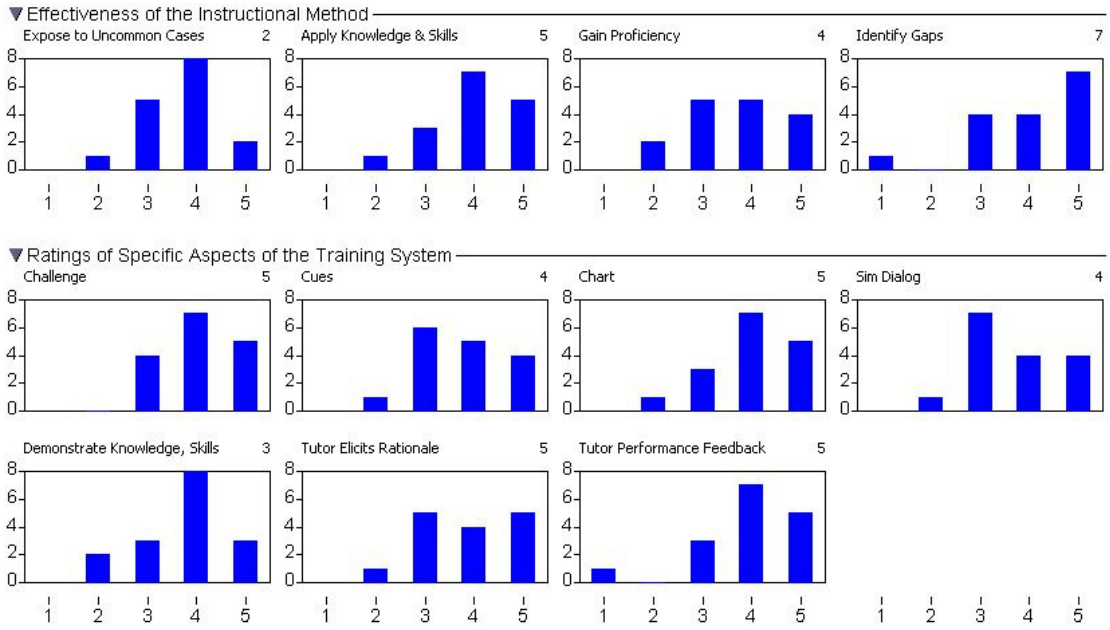

Figure 7. Evaluation Ratings Distributions.

CONCLUSION

The ITS framework described here offers discrete event simulation, dominated by agent behavior and language-based interaction. It provides a simple framework for scripting agent behaviors, and offers two dialog support mechanisms: student-initiative and agent-initiative. It integrates Tutor behaviors as annotations on underlying agent behaviors. The system adopts a web-application format, emphasizing delivery of a richly interactive user interface through a standard web browser. We are strongly encouraged by the results to date in the areas of scenario-scripted language interaction and web delivery of interactive simulations.

To make the simulation machinery run, we defined supporting data representations for *curriculum*, *domain concepts, scenarios*, *agent behaviors* (including several

classes of Tutor behaviors), *extended dialogs*, and *interaction mechanisms* (e.g., live images and forms). Especially as we moved to web-based delivery, we emphasized use of conventional media formats to the greatest extent possible, and adopted the convention that most system data should have a human-readable textual form enabling direct inspection and editing. Using such formats we can build encodings for relevant conceptual spaces identified through domain analysis. In the case of METTLE, these included *CBR agents* and other *conventional medical conditions*, *diagnostic tests*, *possible findings*, and *relevant treatments*. We can also build up inventories of reusable behavior fragments, such as the range of *diagnostic interview questions*. We can also build a set of supporting interactive forms, such as *test* and *treatment* order forms. All of this makes it easier to develop additional training scenarios by leveraging pre-existing content.

Decisions about data representation, work with web-based UIs, and experience building and using earlier generations of authoring tools all fed into decisions about authoring techniques. We do not believe there is any one-size-fits-all approach to authoring. Tools appropriate to one class of user may be annoying and frustrating (or incomprehensible) to others. Further, putting together a complete scenario requires a range of skills, most likely embodied in a range of individuals. When it comes to developing custom authoring tools, many costs—both obvious and subtle—must be recognized. In general, tool suites struggle to keep up with changes in the underlying models of what is being authored. Tool limitations become especially noticeable when it comes to authoring high volumes of content (e.g., hundreds of behavior script lines).

Such considerations motivated experiments on using Commercial Off-the-Shelf (COTS) tools for authoring. Our experience suggests that for any given type of data, it is effective to (a) define a simple inspectable textual format, (b) build code for loading and validating data in that format, (c) define translators allowing transformation of data from alternate (COTS tool) formats into the underlying text format for import and validation, (d) if necessary prepare templates, style sheets, macros or other customizations of the COTS tools, (e) set up the environment with a smooth cycle for COTS editing, translation, import, and validation, making it easy to catch errors introduced through looser authoring, and finally (f) where performance becomes an issue, add a compiler to pre-process the textual form in 'a' for more efficient runtime use. It is easier to build robust efficient text-to-text translators and validators than to build UIs that compete with the immense development time invested in COTS applications like spreadsheets—not to mention the millions of hours of cumulative user experience and testing. As with dialog scripting and web interfaces, this is a direction we expect to pursue further in the context of future projects.

Finally, our evaluation of the CBR prototype has been very encouraging. A wide range of emergency physicians (in age, gender, location, and experience), subject to all the time pressures of professional life, and offered very little support, were able to install and run the system, and for the most part enjoy the experience and judge it to have substantial educational potential.

ACKNOWLEDGEMENTS

We would like to thank the U.S. Army Telemedicine and Advance Technology Research Center (TATRC), and in particular, Mr. Harvey Magee for their support in the development of METTLE under contract W81XWH-04-C-0067. Discussion of application of the Enact tools to SSTR training is based upon work supported by the ERDC-CERL Contracting Office and the Small Business Innovative Research (SBIR) Program under Contract No. W9132T-08-C-0012. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of TATRC, the ERDC-CERL or SBIR Program.

REFERENCES

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

Domeshek, E, Holman, E. & Ross, K. (2002). Automated Socratic Tutors for High Level Command Skills. Proceedings of the 2002 Interservice/Industry Training Simulation and Education Conference.

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

Ong, J. & Ramachandran, S.(2002). Intelligent Tutoring Systems: The What and the How. *Learning Circuits*, February, 2000. Available at http://www.learningcircuits.org/feb2000/ong.html

Smith, R. (2003). Application of Existing Simulation Systems to Emerging Homeland Security Training Needs. In Proceedings of *Simulation Interoperability Workshop—Europe 2003*.