Uncovering the Hidden: Tradeoffs in Rationale Elicitation for Situated Tutors

Sowmya Ramachandran,
Eric Domeshek, Randy Jensen
Stottler Henke Associates Inc.
San Mateo, CA
sowmya@shai.com, domeshek@shai.com,
jensen@shai.com

Anthony Aukamp
Comtech TCS
Pensacola, FL
aukamp@comtechtel.com

ABSTRACT

Deep assessment of a student's performance on cognitive tasks requires understanding not just their actions but also their rationales. Human instructors often attempt to gain insights into a student's reasoning through several channels: they observe patterns of attention or false starts; they listen to the student thinking aloud; or they ask and elicit explicit statements of rationale. Automating such assessments has the potential to increase the effectiveness of Intelligent Tutoring Systems (ITSs) but can be computationally challenging. There is an additional consideration when automated tutoring happens in the context of simulation-based training, where the focus is on providing an immersive experience that is reflective of real-world task performance. Inserting these types of assessments into a simulation context runs the risk of negatively impacting learner immersion and engagement.

This paper presents an analysis of the trade-offs to consider in designing automated approaches to eliciting information about student reasoning, and their impact on the development of simulation-based ITSs. There are a variety of ways by which an ITS may ask students to state their reasoning or explain their actions explicitly in a manner that can be automatically assessed by the tutor. However, the interface for providing student rationales and the technology needed to assess these inputs must be carefully considered to avoid pitfalls such as giving away the problem solution, imposing additional cognitive load, or impacting trainee engagement. This paper illustrates these trade-offs with practical examples and describes a balanced design approach that was used successfully in an ITS for a troubleshooting domain. The rationale elicitation technique used in this ITS has received high ratings for usability and effectiveness in a controlled validation study.

ABOUT THE AUTHORS

Dr. Sowmya Ramachandran is a research scientist at Stottler Henke Associates, where her research focuses on the application of Artificial Intelligence (AI) and Machine Learning to improve education and training. She leads research and development of intelligent tutoring systems (ITS) and ITS authoring tools for a diverse range of military and civilian domains. Dr. Ramachandran headed the development of ReadInsight, an intelligent tutor for teaching reading comprehension skills to adult English speakers. She also led the development of a tutor for training Tactical Action Officers in the Navy. This system uses natural language processing technologies to assess and train TAOs and is currently in operational use at the Surface Warfare Officers School. Most recently she led the development of an ITS for training U.S. Navy Information Systems Technicians in troubleshooting and maintenance skills. Dr. Ramachandran holds a Ph.D. from The University of Texas at Austin. For her dissertation, she developed a novel machine learning technique for constructing Bayesian Network models from data.

Dr. Eric Domeshek is an AI project manager at Stottler Henke Associates, Inc., where he leads and supports projects applying AI technology to problems in training and decision support. He has worked on a wide range of ITSs and related training, education, and simulation environments spanning applications to military tactics, medical diagnosis, engineering systems management, business decision-making, and historical analysis. He is particularly interested in exploration of Socratic tutoring techniques and the development of authoring tools. He led the work on the authoring tools for the Intelligent Tutoring Authoring and Delivery System (ITADS) ITS development recently for the U.S. Navy. Dr. Domeshek received his Ph.D. in Computer Science from Yale University, focused on case-based reasoning. For his dissertation, he developed representations of decision rationale for social situations, intended to support case retrieval; this included extensive representations of characters' relationships, traits, and motivational structures. He served as research faculty at the Georgia Institute of Technology College of Computing, where he contributed to the development of a line of case-based design aids. He was also an assistant professor at Northwestern University, developing goal-based scenario training systems at the Institute for the Learning Sciences.

Randy Jensen is a group manager at Stottler Henke Associates, Inc., working in training systems since 1993. His research areas include adaptive training, distributed learning, game-based training, behavior modeling, and natural language processing. He has led projects to develop Intelligent Tutoring Systems and automated after-action review tools for the Army, Air Force, Navy and Marines. Recent work includes a model-based performance assessment capability for training troubleshooting skills in an ITS for the U.S. Navy. He also recently led the development of a game-based trainer for small unit tactical decision-making at the United States Military Academy at West Point. Mr. Jensen holds a B.S. with Honors in symbolic systems from Stanford University.

Anthony Aukamp is an instructional systems designer at Comtech TCS, where he supports integration of technologies and instructional design projects. He has been working with training, education and simulation systems with the U.S. Air Force and Navy since 2006. His research focuses on ways to facilitate learning through effective and efficient uses of technology. Recent work has included visualizing instructional graphics, the user interface and interactions for ITADS ITS development. Mr. Aukamp holds a B.S. in information technologies from University of West Florida.

Uncovering the Hidden: Tradeoffs in Rationale Elicitation for Situated Tutors

Sowmya Ramachandran,
Eric Domeshek, Randy Jensen
Stottler Henke Associates Inc.
San Mateo, CA
sowmya@shai.com, domeshek@shai.com,
jensen@shai.com

Anthony Aukamp
Comtech TCS
Pensacola, FL
aukamp@comtechtel.com

INTRODUCTION

Coaching students effectively through cognitively challenging tasks requires an assessment not just of their overt actions but also of the reasoning that led to those actions. Human tutors can gain insight into the reasoning behind a student's actions through several channels: they can observe patterns of attention or false starts, they can listen to student self-commentary, they can require the student to "show their work," and they can ask and elicit explicit explanations and justifications (Fox, 1993). For example, the following is an excerpt from a transcript of a math tutoring session. The student, in the process of solving a problem, says the following to the human tutor:

> *Um, this, okay, secant, of theta, I know eqs – equals three. Now, so but – my equation here is secant squared theta. So what I want to say is the square root of secant squared theta would equal three, right?*

The student is thinking aloud, providing the tutor with an insight into his/her thought process and his knowledge of the underlying principles, as well as indicating that he does not have a high level of confidence in his reasoning. These are important clues that should inform the coaching decisions made by the tutor.

Efforts at mimicking human tutoring through automation—a goal of Intelligent Tutoring Systems (ITSs)—face the challenge of eliciting such rationales while satisfying several other constraints:

- There must be a technologically feasible mechanism;
- The mechanism must be cost-effective to implement; and
- The mechanism must be easily usable by students within the tutoring context.

AutoTutor is an example tutoring approach that employs natural language processing techniques to conduct tutorial dialogs and elicit rationale for certain types of domains—e.g. qualitative physics problem solving (Graesser, et al., 2005). ComMentor was a training system that used dialog-based argumentation, similar to AutoTutor, to teach tactical decision-making skills by exploring the rationales for alternate decisions and their impact on scenario outcomes (Domeshek, Holman, & Ross, 2002). These approaches can produce behavior that mimics some of the most effective human tutoring strategies known. However, when attempting natural language processing, there is a stiff tradeoff between quality and cost. AutoTutor started out using very simple text processing approaches that scale well, but that cannot distinguish between critically different meanings. For instance, without a syntax module, "disconnect the printer from the network and connect it to the computer" is no different to the system than "disconnect the printer from the computer and connect it to the network." Adding linguistic knowledge and competence raises costs. In contrast, ComMentor abandoned unstructured language for structured forms, short noun phrase parsing, and an ability to point at elements in the work environment. ComMentor's successors have explored both structured choice input mechanisms and simple text matching techniques supplemented by student confirmation of matches.

Many systems that emphasize dialog place a priority on exploring reasoning and decision rationale over interacting with a simulated environment. For example, in systems that use AutoTutor for physics problem solving, engaging in a dialog with the tutor has been the main mode of activity. Similarly, with ComMentor, the exploration of rationales via dialogs was the main focus of the exercises; the problem-solving happened in the context of such dialogs.

In contrast, embedding rationale elicitation within the context of simulations is largely unexplored. Here we are talking about systems where simulation is the focal mode of training, with rationale elicitations used to support assessment and coaching. The challenges of eliciting and assessing rationales are compounded in such simulation-based training systems because they place great emphasis on learner engagement and immersion in real-world tasks. So our three initial constraints of *feasibility*, *affordability*, and *usability* are supplemented by concerns about *user engagement* and *cognitive load*.

In this paper we discuss these challenges in the context of ITADS (Intelligent Tutoring Authoring and Delivery System), a trainer that uses simulations to teach troubleshooting skills. We describe the approaches used for rationale elicitation, the challenges presented by these approaches in terms of usability and acceptance, and conclude with some general design guidelines.

BACKGROUND: ITADS TRAINING APPLICATION

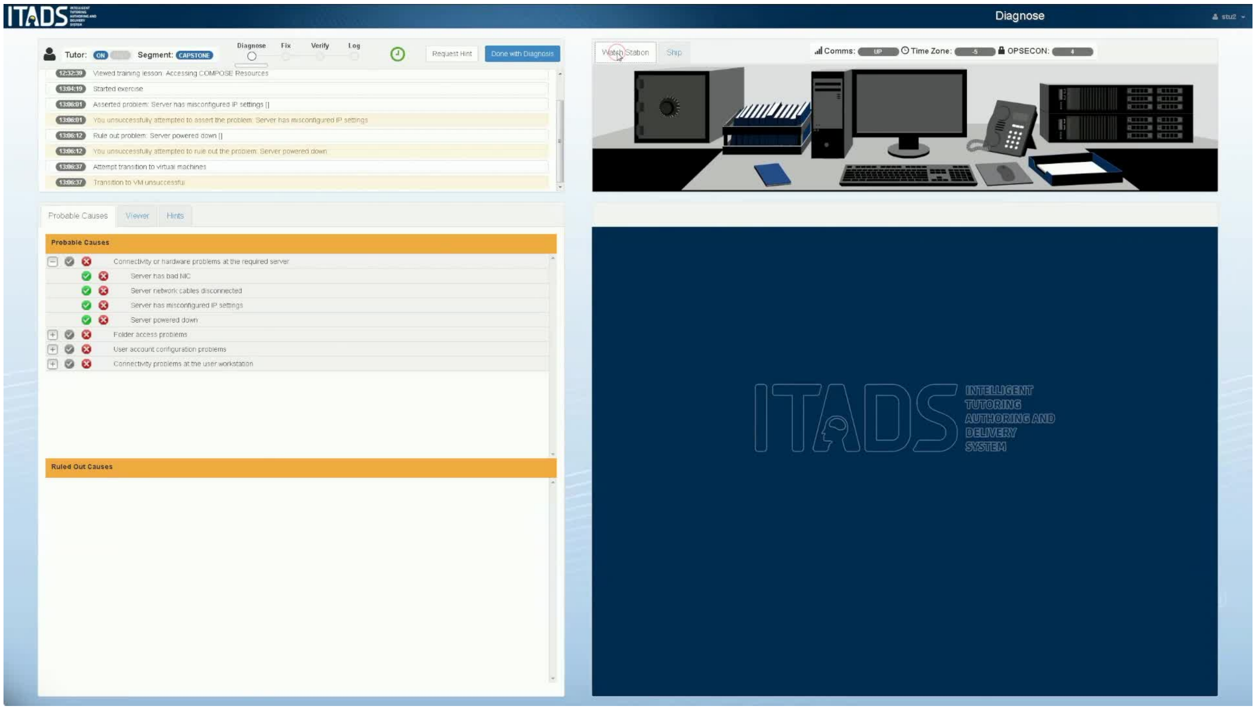
ITADS is an intelligent tutoring system for training U.S. Navy entry level Information Systems Technology (IT) support staff. In particular, it aims to teach fleet-specific skills and knowledge to new recruits undergoing training at the Navy's IT-A school. ITADS uses the problem-based learning approach to teach troubleshooting skills. The majority of its training is conducted in the context of real-world problems as encountered in a simulation environment. A training scenario presents a student with an IT trouble ticket that he/she must address following the Navy's six-step troubleshooting procedure (CISCO, 2012). The simulation is composed of a network of virtual machines (VMs) mimicking workstations, servers, and routers as configured on a typical Navy ship. For each scenario, a fault is then introduced within this environment. The student's task is to perform tests on the VM network to identify and fix the fault. The target audience for the ITS being new recruits who attending the Navy's IT-A school have no on-the-job experience with troubleshooting fleet IT systems. The ITS is intended to fill the gap between classroom IT training that is not specific to Navy IT systems and on-the-job skill required on the fleet. However, even if they have no experience troubleshooting Navy specific systems, it is likely some students have prior experience with troubleshooting other systems.

The simulation is designed to immerse students in real-world tasks and thus enhance training effectiveness and learner engagement. ITADS automatically assesses performance and provides adaptive coaching and feedback. The assessments are also used to maintain a dynamic student model representing the mastery of the student on domain knowledge, skills and abilities (KSAs). Figure 1 shows a screen capture of the ITADS student interface. A detailed view of the relevant portion of the tutor interface will be shown later. For the moment, it is sufficient to consider the general layout.

The right half of the screen is the simulation and left half of the screen contains tutor interactions. At the top of the left side is a panel that shows an ongoing transcript of student actions and tutor feedback. Students can request hints, which are also presented in this panel. At the center of the left-hand side is a panel titled "Probable Causes." This will be the topic of discussion in the following sections.

As mentioned, one main objective of ITADS is to teach troubleshooting skills. The Navy prescribes the following six-step troubleshooting procedure.

1. Identify and replicate the reported problem
2. Establish a theory of a probable cause
3. Test the theory to determine the cause
4. Establish a plan of action to resolve the problem and implement the solution
5. Verify full system functionality and, if applicable, implement preventative measures
6. Document findings, actions, and outcomes



```
12:32:39 Viewed training lesson: Accessing COMPOSE Resources
13:04:19 Started exercise
13:06:01 Asserted problem: Server has misconfigured IP settings []
13:06:01 You unsuccessfully attempted to assert the problem: Server has misconfigured IP settings
13:06:12 Rule out problem: Server powered down []
13:06:12 You unsuccessfully attempted to rule out the problem: Server powered down
13:06:37 Attempt transition to virtual machines
13:06:37 Transition to VM unsuccessful
```

Probable Causes

- Connectivity or hardware problems at the required server
  - Server has bad NIC
  - Server network cables disconnected
  - Server has misconfigured IP settings
  - Server powered down
- Folder access problems
- User account configuration problems
- Connectivity problems at the user workstation

Ruled Out Causes

Figure 1. ITADS Tutor Interface

Given an IT troubleshooting problem in the form a symptom report (also called a trouble ticket), students are expected to form hypotheses about underlying faults. Based on a mental model of IT systems, they must select actions to perform to test their hypothesis, and observe and interpret the results of those actions. The refutation of candidate faults and the selection of a root cause are the most important inferences students must learn to make. Students adopt troubleshooting strategies based on their mental models and currently available information, perform actions based on their intentions, observe the results, interpret the results, and make inferences to refine their hypotheses. This cycle is repeated until they identify the fault (Jensen, Ramachandran, Domeshek, Tang, & Marsh, 2016).

Since training troubleshooting skills is an important objective for ITADS, assessing and coaching the knowledge supporting troubleshooting inferences is critical. When it comes to assessing student expertise and knowledge about the target system at a functional and system level, the inferences and strategies are at least as important as the actions performed. Unfortunately, inferences and strategies are not directly observable by the automated tutor. The problem, then, is how to augment the simulation-based tutor so it can elicit some useful aspects of the student's decision-making rationale, while keeping the focus on the troubleshooting process rather than making rationale dialogs the centerpiece of interaction.

RATIONALE ELICITATION IN ITADS

There are many potential ways an ITS elicit student rationale (Lajoie, Faremo, & Gauthier 2006; Woolf 2008). They may be characterized along the following dimensions.

- **Interaction modality:** ITS developers have experimented with a range of primary input modalities, including multiple-choice, form-based utterance construction, free-from text inputs that are assessed by simple text-matching, or more elaborate natural language processing (NLP*)*. There can also be speech-based versions of many of these options. Multiple-choice interactions offer ease of use and have low impact on cognitive load. Furthermore, they are not too intrusive and are not likely to negatively affect engagement. The technology costs of assessing such interactions are also very low. However, there is an increased content authoring burden compared to free-form text inputs that are automatically assessed using more costly NLP techniques. Finally, multiple-choice interactions also have the effect of prompting students that can be avoided by the use of free-form text inputs.

- **Interaction extent/purpose**: When an interaction about rationale occurs, it can be limited to a quick *single* turn, or it can be allowed (or required) to become an extended multiple turn interaction—such as an extended dialog sequence. The potential uses for multiple turns can vary; they can be used for clarification, elaboration, correction, etc. Figure 2 shows an example multi-turn dialog. The system presents each dialog turn as a question, followed by a set of answer choices (not shown). The student selects one or more of these choices as his or her response. The tutor evaluates the response and follows up with another question or a summary explanation of the discussion. Figure 2 shows a three-turn dialog. Note that turns follow in sequence, i.e. the student will only see the question for the first turn. The question of the second turn is revealed once the first turn has been completed.

  An important benefit of multi-turn dialogs is that they enable in-depth, adaptive exploration of student reasoning. The tutor can respond to answers with follow-on queries to probe their reasoning in greater depth. This resembles Socratic tutoring techniques that have been shown to be an effective teaching approach (Rosé, Moore, Vanlehn & Allbritton 2001). The cost of multi-turn dialogs, however, is that engaging students in long-drawn out exchanges might create a high-cognitive load and is certainly intrusive—tending to disrupt their immersive problem-solving experience. Additionally, the more turns in a dialog, the more complex the authoring effort.

- **Interaction initiative:** It is possible to construct tutor-initiative systems where only the tutor decides when to ask the student for aspects of their rationale (tutor-initiative). This has been the approach taken by a number of ITSs (Woolf 2008). It is, of course, also possible to build mixed-initiative systems that allow either party to open (a round of) rationale discussion. Balancing the two (student initiative vs. tutor initiative) in a mixed-initiative system can be a challenge and must be carefully designed. Interactions that are largely student initiated are less intrusive than those that are tutor initiated. However, the risk is that the tutor does not always get the information it needs simply because a student chose to not interact extensively. In such cases, the tutor must be robust enough to function with minimal rationale inputs from the student.

- **Interaction centrality:** Leaving aside speech-based interaction, all modalities require some graphical user interface (GUI) representation. Rationale GUI components can vary regarding how visible, central, and persistent they are. Rationale collection can be a constant and highly salient part of the user interface, a minor element, or something that appears only when needed (e.g., pops up on tutor-initiative). A rationale collection interface that is a salient part of the overall simulation can make it a part of the exercise flow, whereas popups can be disruptive. The persistence and constancy of the interface makes it easier for students to know what is expected, whereas each new popup places an additional cognitive burden of understanding the instructions and responding appropriately.

- **Interaction continuity:** Even when each rationale elicitation interaction is limited to a single action rather than an extended dialog, it is still possible for those actions to be placed in a larger *ongoing context*, especially if the rationale collection GUI components are central to the user interface and persistent over time. The simpler alternative is that each interaction is an *isolated* query or comment.

In developing ITADS, any solution with high module or content costs was eliminated from consideration, as this would have been inconsistent with the project's budget and objectives. Thus, we rejected designs requiring full natural language and speech processing.

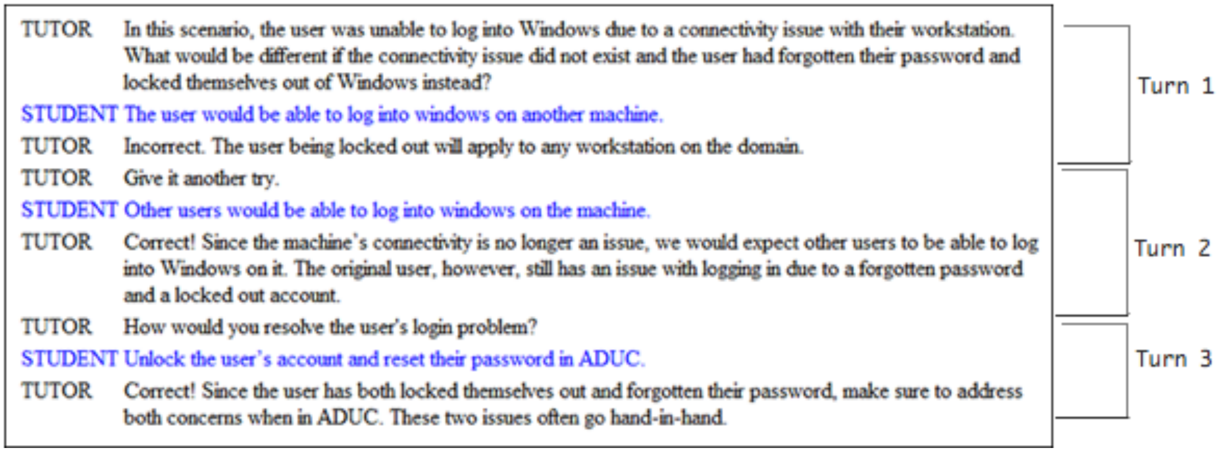
Our initial design used multi-turn dialogs for rationale elicitation. Despite their potential drawbacks, the benefit of in-depth probing and the potential impact on learning led to the selection of multi-turn dialogs as the top design option for ITADS rationale elicitation. A second choice concerned the format of student response inputs. As illustrated in Figure 2, we used a multiple-choice format for student responses to dialog questions in order to avoid the complexities of NLP.

We demonstrated this dialog-based approach to a panel of three IT-A school instructors initial design input. The instructors agreed that the dialogs add instructional value by amplifying the scenario content and serving to develop a strong mental model of the system. However, the sample dialogs raised concerns about excessive verbosity, presenting the risk that they would fail to engage their students, especially those who tend to skim over text. To mitigate concerns about student patience for text-based dialogs, a version of the dialog system that used text-to-speech technology for to read the tutor's dialog utterances aloud was demonstrated. It too was rated sub-optimal because the generated speech was perceived as too robotic.

Given that designing a dialog mechanism based on written language or any kind of spoken input was beyond the scope of this effort, we were left primarily with multiple-choice, form-based, or simple text approaches. However, even the form-based and simple text designs were ultimately deemed too expensive for this application. The instructors thought that the multiple-choice interaction would negatively impact student acceptance because it appeared very similar to traditional computer-based training courses with multiple-choice quizzes. This criticism was echoed by students in separate sessions. The instructors also felt that dialogs in the middle of an exercise—especially extended dialogs—would compromise the sense of engagement and immersion that a simulation-based training environment provides.

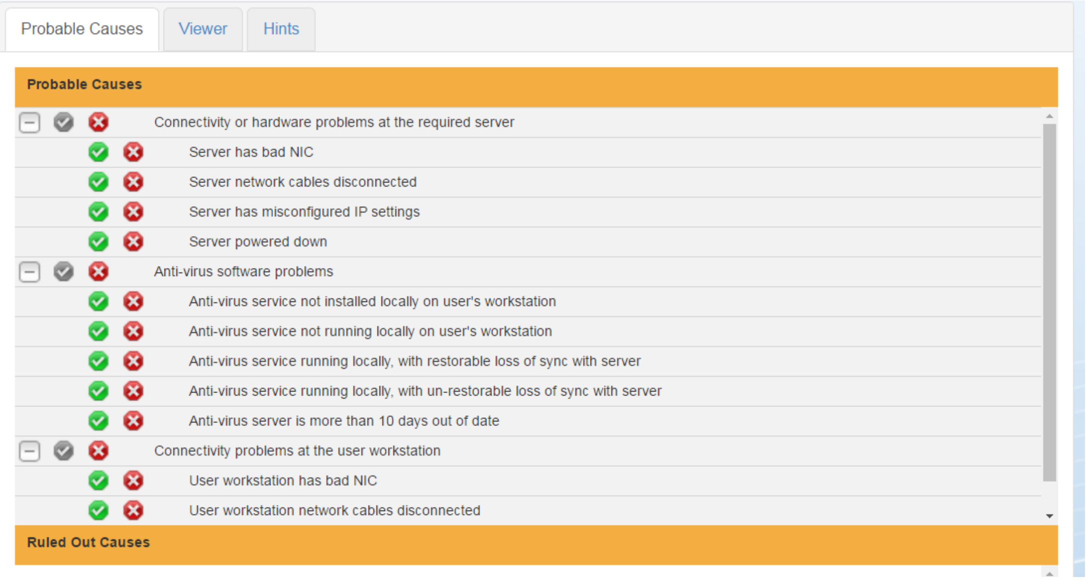
Since the overall feedback on the use of multi-turn dialogs for rationale elicitation during an exercise was negative, based on their disruption of student engagement in problem-solving, the initial design was modified to limit the use of dialogs during exercises to single turn interactions and only when absolutely necessary. Multi-turn dialogs were reserved for after-action review and even those were limited to one or two per exercise to address and assess critical troubleshooting knowledge and inference. Figure 3 shows an example of a dialog used during after-action review.

The design for rationale elicitation during an exercise was modified to trim its complexity and make it less intrusive. A custom user interface panel was designed, resulting in a continuously available mixed-initiative form of rationale collection, built around single actions that together form an ongoing interaction. Like the tutor developed by Lajoie, Faremo, & Gauthier (2006), ITADS presents rationales as a set of failure hypotheses that students update throughout an exercise. Figure 3 shows the user interface for hypothesis refinement called the Rationale Panel.



TUTOR In this scenario, the user was unable to log into Windows due to a connectivity issue with their workstation. What would be different if the connectivity issue did not exist and the user had forgotten their password and locked themselves out of Windows instead?
STUDENT The user would be able to log into windows on another machine.
TUTOR Incorrect. The user being locked out will apply to any workstation on the domain.
TUTOR Give it another try.
STUDENT Other users would be able to log into windows on the machine.
TUTOR Correct! Since the machine's connectivity is no longer an issue, we would expect other users to be able to log into Windows on it. The original user, however, still has an issue with logging in due to a forgotten password and a locked out account.
TUTOR How would you resolve the user's login problem?
STUDENT Unlock the user's account and reset their password in ADUC.
TUTOR Correct! Since the user has both locked themselves out and forgotten their password, make sure to address both concerns when in ADUC. These two issues often go hand-in-hand.

Figure 2. Example multi-turn dialog used for after-action review



Probable Causes

⊟ ✓ ✗ Connectivity or hardware problems at the required server
- ✓ ✗ Server has bad NIC
- ✓ ✗ Server network cables disconnected
- ✓ ✗ Server has misconfigured IP settings
- ✓ ✗ Server powered down

⊟ ✓ ✗ Anti-virus software problems
- ✓ ✗ Anti-virus service not installed locally on user's workstation
- ✓ ✗ Anti-virus service not running locally on user's workstation
- ✓ ✗ Anti-virus service running locally, with restorable loss of sync with server
- ✓ ✗ Anti-virus service running locally, with un-restorable loss of sync with server
- ✓ ✗ Anti-virus server is more than 10 days out of date

⊟ ✓ ✗ Connectivity problems at the user workstation
- ✓ ✗ User workstation has bad NIC
- ✓ ✗ User workstation network cables disconnected

Ruled Out Causes

Figure 3. ITADS Rationale Panel

The Rationale Panel is pre-populated with a set of hypotheses at the start of the exercise. The hypotheses are automatically generated from assessment model that establishes connections between observed system behavior and potential system faults (Jensen, Ramachandran, Domeshek, Tang, & Marsh, 2016). The assessment model was developed with close guidance from SMEs. The pre-populated list of hypotheses includes distractors that can be generated from the model and also specified by subject matter experts (SMEs) using the ITADS Authoring Tool.

Students can refute a hypothesis singly or as a group using the red "x" button. They can assert or confirm a single hypothesis using the green "check-mark" button. Currently ITADS operates on a single-fault assumption, which limits the assertions to a single hypothesis. Groups of hypotheses can be expanded or collapsed as needed. This is an example of a *single-turn* interaction because the tutor provides feedback after every rationale update but does not follow up with additional probing. It is an example of *ongoing* interaction because the panel remembers and displays the state of all considered hypotheses based on earlier actions. It is largely a *student-initiative* interaction because the student is free to choose the timing and extent of their rationale updates. Only one type of update is enforced and that is the assertion of their final problem diagnosis before students can move on to the fault repair phase.

Assessments of student performance are primarily based on Rationale Panel updates (Jensen, Ramachandran, Domeshek, Tang, & Marsh, 2016). Simulation actions are indirectly assessed based on the rationale updates. This gives students a greater degree of freedom to find alternate paths to a diagnosis since the tutor does not force them into any particular scripted action sequence. Although there are assessments that monitor student activity to identify actions that are strictly irrelevant or incorrect for the scenario, students are largely given leeway to perform exploratory diagnostic actions. For assessment of their reasoning process and system knowledge based on Rationale Panel updates, the tutor maintains a model relating simulator actions to fault hypotheses. When a student asserts or refutes a hypothesis, the tutor uses the model to check the consistency of the assertion (or refutation) with all the diagnostic information revealed to the student up to that point in the scenario (i.e., information given at the start of the scenario or revealed subsequently by student actions). An inconsistent hypothesis update is assessed as an incorrect inference. Timely refutations of hypotheses are encouraged but not required. Omissions of refutations, therefore, do not adversely affect exercise scores. However, the student is required to correctly assert the system fault for the scenario before they can move on the phase of fixing the problem[1]. When a hypothesis is asserted, the tutor automatically scores the remaining hypotheses as refutations due to the single-fault assumption.

---

[1] Students do have the option of asking the tutor for this answer if they cannot figure it out. This is to give them a way to move forward with the exercise even if they cannot successfully identify the root cause of the problem in the scenario.

FEEDBACK ON ITADS RATIONALE ELICITATION APPROACH

Instructors and students were closely involved in system development. Six workshops were held during the design and development phases where instructors and students used the tutor and provided feedback on the design overall, and on the rationale elicitation approach in particular. These evaluations focused on usability as well as pedagogy. For each workshop, we assembled a panel of three IT-A school instructors and a panel of five IT-A school students. Each panel member was given a few hours to interact with the tutor with help from the facilitators when needed. The panels were then asked to comment on specific aspects of the rationale elicitation mechanism and also provide suggestion for improvement. These informal, but deeply informative sessions, were conducted as free-flowing discussions. The sessions with instructors were held separately from those with students; each was unaware of the feedback provided by the other. The following subsections describe the feedback provided by each group.

Feedback from Instructors

The instructors reacted very favorably to the overall concept of rationale elicitation. They felt that it would be an effective means of reinforcing student mental models of IT networks. Fine-tuning the execution of this concept required a few iterations of design refinements. The end product of this process was well received both by instructors and students. Here we walk through the design iterations, specifically the feedback from instructors and students and the resulting design updates, because it potentially holds value to developers of other such systems.

Initially the instructors had reservations about the impact of these interventions on the level of immersion and engagement experienced by the students. Their comments on multi-turn dialogs were discussed above. Instructors had two main concerns about the approach represented by the Rationale Panel shown in Figure 3. :

1. It prompts the students about the hypotheses they should consider; and
2. It interferes with the tempo and the level of immersion provided by the simulation.

The concern about prompting stemmed from the observation that the Rationale Panel includes all the hypotheses a student should consider, given the statement of the symptom provided initially in the scenario. Instructors worried that this would discourage students from independently making the connections between a symptom and potential problems. An additional criticism was that students could potentially game the tutor by clicking each hypothesis until they found the correct answer. Finally, they observed that some students would go down the fault list item by item, instead of planning their course of action strategically.

The concern about gaming validated the design choice of tying assessments to the rationale updates. Each rationale update is assessed using a model-based inference approach (Jensen, Ramachandran, Domeshek, Tang, & Marsh, 2016). Clicking through the list of hypotheses is therefore not free; each incorrect assertion/refutation counts negatively for performance scores and mastery estimates of the curriculum. Furthermore, the acceptance of a rationale update is contingent on the actions performed, so that hypothesis assertions—even of the correct hypothesis—are rejected so long as the student has not gathered sufficient supporting evidence. Since an assertion is only allowed when it is justified, a student cannot tell the difference between a premature correct assertion and an incorrect assertion. Thus, the incentives for gaming are minimized.

This issue of prompting was partially addressed by introducing distractors in the form of additional hypotheses that may not actually be implied by the initial symptoms. To further mitigate this concern, the rationale elicitation interface was redesigned to transition from a pre-populated panel to a student-populated panel when student mastery on associated KSAs exceeded a fixed threshold. The student-populated panel is empty at the start of an exercise, whereas a pre-populated panel, as the name implies, includes a list of hypotheses that can then be refined by the student. Figure 4 shows the student-populated panel that starts out empty. Students can add a hypothesis by typing terms in the search box and finding appropriate items to add. The instructors felt that this design update effectively addressed the problem of prompting.

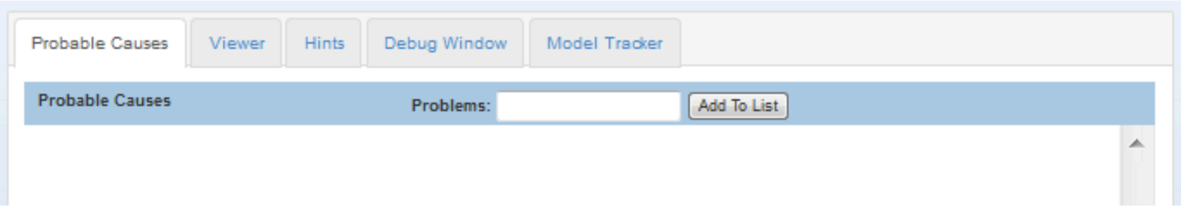


Figure 4. Student-Populated Rationale Panel

The residual criticism of the user interface was that the pre-populated list of hypotheses was sometimes too long; this would not only affect usability but could also overwhelm novice students (i.e. students who have not yet achieved a high level of mastery on the skills). This concern was addressed by grouping hypotheses into a hierarchy that could be selectively collapsed or expanded. This gives students control over how many items are visible at one time.

With respect to engagement, instructors felt that the Rationale Panel would be helpful to students but not to advanced students (i.e. students who have acquired high-levels of mastery on skills), who would perceive it as a nuisance. They thought that requiring students to update the Rationale Panel at frequent intervals would detract from the sense of immersion in the virtual machines. There were also concerns that such interruptions would increase the cognitive load on the students, who might lose track of their actions as they took time to update the panel. The approach of fading to a student-populated list of hypotheses fails to address this concern because there are more steps involved in asserting/refuting hypotheses than in the pre-populated version.

To address the perceived nuisance factor and concerns about loss of engagement, the tutor design was updated to enforce only the most important update—namely, the assertion of the correct hypothesis. Refutations of incorrect hypotheses were made optional. The scoring was designed to treat all remaining hypotheses as refutations at the time of the assertion of the correct one. This update to the design accommodates a range of preferences and expertise levels. Expert students are not compelled to reveal their inferences each step of way, though they might choose to. Novice students can use the Rationale Panel as a cognitive aid to record and check their inferences, though they are not compelled to do so. This design does, however, reduce the tutor's insight into a student's inferences. Thus transparency of reasoning was traded to achieve greater student engagement.

Feedback from Students

Student responses to the Rationale Panel and the dialogs were somewhat different from those of the instructors. The students valued the Rationale Panel for the cognitive assistance it provided, which they found very helpful, especially early in the training. Though the students are exposed to the process of troubleshooting during lectures, they do not have much experience with the process in practice. For some students, the Rationale Panel served as a checklist and a reminder of the hypotheses they should consider. It served as a cognitive aid to the troubleshooting process and was very helpful to students with limited experience in this task. They did perceive the prompting effect from listing hypotheses in the panel and independently suggested the approach of fading to a student-populated panel as a useful direction. The students, unlike the instructors, did not find that the Rationale Panel detracted from the immersive experience. As predicted by the instructors, the students felt that initial dialogs were too similar to traditional multiple choice quizzes.

Observations of students during the workshops revealed a range of behaviors when it came to actually updating the hypotheses after simulator actions. Most students methodically refuted hypotheses when necessary, while a few got immersed in the simulation and failed to update the hypotheses. A strict requirement to explicitly rule out a hypothesis in the Rationale Panel immediately following evidence of its absence from the simulation would have unfairly penalized students who became immersed in the experience.

VALIDATION STUDY: FINDING RELATING TO RATIONALE ELICITATION

We evaluated the usability and utility of the Rationale Panel as a part of a broader training effective validation (TeleCommunications Systems, Inc. 2015). We used a survey instrument to get student reactions to the final Rationale Panel design. The survey was conducted with ten students who use the ITADS tutor over a period of ten days and completed twenty-two training scenarios. This gave them ample opportunities to engage with the Rationale Panel. The survey instrument included questions about the usability and usefulness on the following components of ITADS: didactic presentations, training scenarios, the Rationale Panel, hints and feedback, and after-action review. Students filled out the surveys at the end of training segments.

The scores were coded on a range of -2 to 2, with -2 being the most negative reaction (i.e. Strongly disagree), 2 the most positive (Strongly agree) and 0 being a neutral reaction. The Rationale Panel received the highest satisfaction rating of all the five evaluated ITADS components, with an average score of 1.50.

The following table shows the breakdown of student reaction to the rationale panel. "Most Probable Causes" panel is synonymous with the Rationale Panel described in this paper. This evaluation indicates that the students found the Rationale Panel easy to use and helpful in solving the exercise.

Table 1. Drilldown of student reaction to the Rationale Panel

1. I clearly understood why I had to use the "Most Probable Causes" panel.	1.56
2. It was clear on how to use the "Most Probable Causes" panel.	1.51
3. I found the "Most Probable Causes" panel helpful when I was completing the exercises.	1.46
4. I think that working with the "Most Probable Causes" panel during these exercises will help me do a better job of problem solving when I am in the work environment.	1.49

LESSONS LEARNED

Uncovering students' normally unobservable reasoning processes can have great instructional utility for training cognitively intensive tasks such as troubleshooting, decision-making, and intelligence analysis. As we have seen, a number of different approaches have been developed for this purpose. However, in the context of simulation-based problem-centered training exercises, many techniques for uncovering hidden student cognition can work against maintaining immersion and engagement. Articulating one's reasoning is a reflective act, which takes a student out of the flow of the problem-solving activity. Certainly it can be pedagogically valuable to encourage students to think rather than to simply act. Yet one must avoid the pitfalls of turning a realistic and engaging training exercise into an abstract academic exercise. Simulations require performing actions that are similar to real-world tasks. Reflective tasks, on the other hand, require constructing responses to questions posed by the tutor and seem closer to tests than the real-world tasks being trained.

As a design guideline, extended discussions or activities relating to elicitation of reasoning are a better fit for an after-action review at the end of a simulation exercise. This presents its own set of design challenges. For example, reasoning is contextual, and establishing a specific problem-solving context at the end of exercise can be challenging (and verbose). This could be addressed by providing playbacks of recorded exercise activity to establish context. Alternately, dialogs relating to the thought processes behind an exercise can be structured as separate exercises of a more reflective kind, and hence with a very different feel. In any case, there is value in exploring design approaches that facilitate the separation of simulation performance and reflective activities without significant loss of context.

A related guideline is that it is useful to design interventions that are flexible enough to support different degrees of student engagement. We refined the design of the Rationale Panel to the point where the only direct information the student is required to provide is the final correct assertion of the root cause of the symptoms reported in the trouble ticket. While this reduced the amount of direct information available to the tutor, this had the positive effect of not interrupting those students who were deeply engaged with the simulation. The system, however, was flexible enough that students who wanted to engage more with the tutor had the option of doing so by refuting hypotheses as they worked through their investigations and receiving feedback while doing so.

A third guideline is to design rationale elicitation in forms that also serve as cognitive supports to help students achieve their simulation goals. Several students used the Rationale Panel as a “to do” list to keep track of their current hypotheses and plan their next actions. Cognitive supports that help with reasoning can do double duty as performance coaches and as assessment tools. However, it is important to withdraw these tools over time to ensure that students are able to reason effectively without them. An implication of this is that the tutor will lose some assessment data as a student’s mastery level advances. This trade-off may be acceptable, since fine-grained assessments are most valuable when students are the weakest at the target skills. Detailed assessments allow a tutor to address relevant gaps or confusions at a point where student understanding is most fragmentary and fragile—when saying the wrong thing or addressing the wrong issue is most likely to confuse the student. Coarse-grained assessments may be sufficient for students who have mastered the necessary knowledge and skills.

Finally, a user-centered design approach that emphasized frequent feedback from the stakeholders is invaluable. Users often welcome the opportunity to be involved in the development process. Though research and guidelines can point the research in a promising direction, user feedback makes it possible to personalize the design to particular training audiences. User involvement in the design process also helps ensure their final acceptance of the product with a positive impact on change management.

CONCLUSION

Understanding the reasoning employed by students is important for providing rich feedback and coaching, especially for tasks such as troubleshooting that require deep and accurate mental models. However, there is a tension between eliciting rationales and providing immersion. Frequent and forced interruptions of simulation activity to engage students in extended dialogs can disrupt the flow of the immersive simulation experience. This paper presents a case study in developing a rationale elicitation mechanism for a simulation-based intelligent tutoring system. It presents a walkthrough of the iterative design process, highlighting the concerns raised by instructors and students and how the design responded to those concerns. The issues raised in this case study and the design choices have relevance to other training tasks where reasoning and inference are critical learning objectives. The paper presented a design that effectively addressed the issues of usability and disruption. The resulting system received a high score from students for usability and utility in a controlled study.

ACKNOWLEDGEMENTS

This work was performed under a contract awarded and administered by the U.S. Navy, Naval Air Warfare Center Training Systems Division (NAWCTSD) in Orlando, FL for the Navy’s Center for Information Dominance (CID) IT A-School in Pensacola, FL. The authors wish to thank the entire ITADS team at Stottler Henke and ComTech Telecommunications Corp., as well as our Government sponsors, for their dedication to making this project a success.

REFERENCES

CISCO (2012). *IT Essentials 5.0, Chapter 6, Networks, Slides* 63-69.

Conati, C., & VanLehn, K. (1999). Teaching meta-cognitive skills: implementation and evaluation of a tutoring system to guide self-explanation while learning from examples.

Domeshek, E.A., Holman, E. and Ross, K.G. (2002). Automated Socratic Tutoring for High-Level Command Skills. In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference* (I/ITSEC, 2002).

Domeshek, E.A., Ong, J., and Mohammed, J. (2007). *Final Project Report for Architecture for Distributed, Experiential, Personalized Training (ADEPT)*. Stottler Henke Technical Report for NASA Contract NNJ06JA39C.

Fox, B. A. (1993). *The Human Tutorial Dialogue Project: Issues in the design of instructional systems*. Lawrence Erlbaum Assoc Inc.

Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. Education, *IEEE Transactions on Education*, **48**(4), 612-618.

Jensen, R., Ramachandran, S., Domeshek, E., Tang, K., & Marsh, J. (2016). Simulation awareness: Assessing performance with limited simulation instrumentation. Submitted to the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC 2016).

Lajoie, S. P., Faremo, S. O. N. I. A., & Gauthier, G. (2006). Promoting self-regulation in medical students through the use of technology. *Technology Instruction Cognition and Learning*, **3**(1/2), 81.

Rosé, CP, Moore, JD, Vanlehn, K & Allbritton, D 2001, 'A comparative evaluation of socratic versus didactic tutoring'. In 23rd Annual Conference of the Cognitive Science Society. pp. 869-874.

TeleCommunications Systems, Inc. (2015). ITADS Findings Report. Contract N61340-13-C-0014, CDRL A009.

Woolf, B. P. (2008). John Bransford (Editor), Ann L. Brown (Editor), Rodney R. Cocking (Editor) (2000), How People Learn : Brain, Mind, Experience, and School, National Academy Press. Morgan-Kaufmann.