ITADS: A Real-World Intelligent Tutor to Train Troubleshooting Skills

Sowmya Ramachandran[1*], Randy Jensen[1], Jeremy Ludwig[1], Eric Domeshek[1], and Thomas Haines[2]

[1]Stottler Henke Associates Inc. San Mateo, CA, 94402

[2]Comtech TCS., Pensacola, 32502

[*]Sowmya@stottlerhenke.com

Abstract

Real-world intelligent tutoring systems are important ambassadors for promoting wide adoption of the technology. Questions about affordability, quality control, operational readiness, training effectiveness, and user acceptance are significant in this context. This paper describes ITADS, an intelligent tutor developed to provide a problem-based, experiential learning tool to complement schoolhouse training. The goal was to train US Navy Information Systems Technology support staff in troubleshooting skills through the use of realistic simulations and automated assessment and feedback. This paper describes the tutoring system and a preliminary validation study of its training effectiveness. The results demonstrate that the system is effective in improving troubleshooting knowledge and skills. The ITADS system was successfully developed in twenty-six months from requirements to validation, following strict systems engineering procedures. The results of the training effectiveness study indicate that the ITS also leads to significantly improved performance among Navy IT recruits in troubleshooting tasks.

Keywords: Intelligent Tutoring System, Automated Performance Assessment, Troubleshooting Skills, IT Skill Training

1 Introduction

While research and development of intelligent tutoring systems (ITS) has been ongoing for decades and many significant advances have been made, tutoring systems for training real-world, professional skills have been emerging [2]. These systems are developed under the constraints of operational-level software development with its attendant emphasis on cost-effectiveness, ambitious development schedules, and rigorous software engineering processes. Proof of effectiveness is also an important consideration when developing such systems. This paper describes a highly immersive, situated intelligent tutoring system called ITADS, that targets professional training and was developed under real-world budgetary and schedule constraints. The focus of this paper in on the results of a study of its training effectiveness. We will first describe the ITS and then present the study.

2 ITADS Overview

ITADS is an intelligent tutoring system for training U.S. Navy entry level Information Systems Technology (IT) support staff. The target audience for the ITS are the new recruits who attend the Navy's IT-A school and have limited on-the-job experience with troubleshooting fleet IT systems. The ITS is intended to serve as a bridge between schoolhouse training and on-the-job skills required on the fleet.

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 [1].

The simulation consists of a dedicated virtual IT network of virtual machines (VMs) that is an exact representation of the Naval Shipboard IT network. Thus, the simulation is designed to provide real, hands-on experience of an IT watchstander's responsibilities. Each scenario has an associated VM network in which a fault has been introduced to reflect the training scenario. The student's task is to perform tests on the VM network to identify and fix the fault. 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). The Tutor can function in either of two modes – Tutoring On and Tutoring Off. In the Tutoring On mode, the Tutor provides full intelligent tutoring capability.

This includes monitoring and assessing student performance on simulation exercises as well as providing coaching, feedback, and an after-action review (AAR). In the Tutoring Off mode, the Tutor assesses student progress and performance on an exercise but does not provide coaching or feedback.

The 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, and 6. document findings, actions, and outcomes

Given an IT troubleshooting problem in the form a symptom report (also called a trouble or service ticket), students are expected to form hypotheses about underlying faults. Based on a mental model of IT systems, they must select actions to perform in the VMs 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.

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. 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.

A custom user interface panel was designed for rationale elicitation. Like the tutor developed by [3], ITADS presents rationales as a set of failure hypotheses that students update throughout an exercise. Figure 1 shows the user interface for hypothesis refinement called the Probably Causes Panel. The Probable Causes 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 [4]. 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.

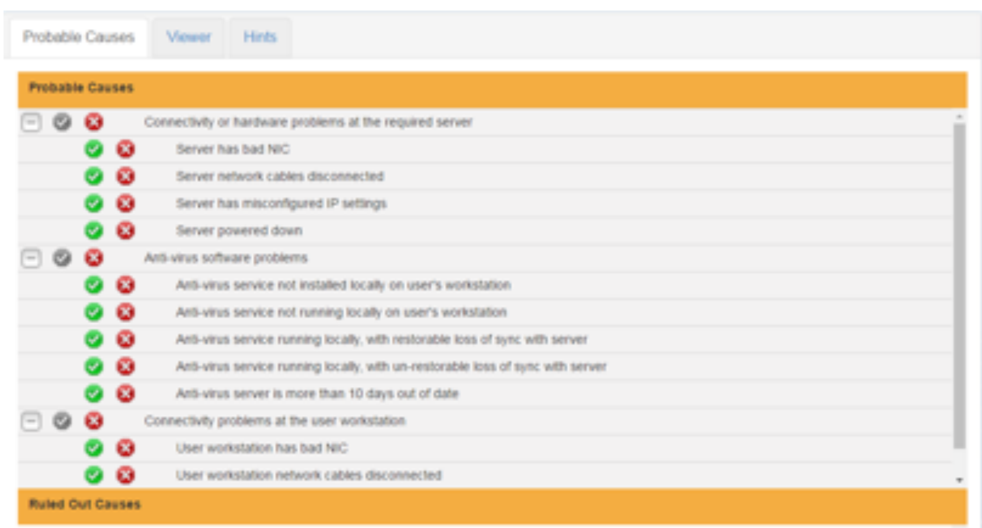

Figure 1. Probable Causes Panel in the Tutor GUI

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 a single-turn interaction in that the tutor provides feedback after every rationale update but does not follow up with additional probing. It is also an *ongoing* interaction because the panel remembers and displays the state of all considered hypotheses based on earlier actions. 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 Probable Causes Panel updates [4]. Simulation actions are indirectly assessed based on these 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.

For assessment of their reasoning process and system knowledge based on Probable Causes 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. When a hypothesis is asserted, the tutor automatically scores the remaining hypotheses as refutations due to the single-fault assumption.

There is coaching available in the form of feedback and hints throughout the diagnosis task. ITADS provides process-oriented feedback wherein an incorrect assertion leads to a tutor-generated message about the last action that provides evidence against that assertion. This context-sensitive feedback is automatically generated from the assessment model. The Tutor provides hints that are generated using a greedy search approach for determining the best solution path from the current state. When a student finishes an exercise, the Tutor presents an after-action review (AAR). Here the student can review a summary of their overall performance in the exercise, their performance on primary learning objectives for the exercise, and receive a final exercise performance score as determined by the Tutor. The AAR can also include interactive, form-based dialogs between the ITS and the student to promote reflection.

The ability to control and maintain content without relying on outside contractors was an important project requirement. The ITADS system includes an authoring tool to enable end user organizations to modify and create scenarios. Users can create, edit, and delete training scenarios, and scaffolding content like hints and dialogs. The authoring tool provides an integrated development environment for creating and editing various kinds of content data using a set of incorporated custom tools.

3 ITADS Training Effectiveness Evaluation

We conducted a summative evaluation to study the learning gains resulting from ITADS. We conducted a controlled experiment with two independent groups of students drawn from a population representative of the target trainees.

- The experimental group went through a complete program of training with ITADS.
- The control group only went through the didactic lessons.

Both groups were administered a common post-test. The differences in post-test performance of the two groups provided derived measures for comparison and analysis in the evaluation of earning gains attributable to the use of ITADS.

The purpose of the evaluation was to study the added value that ITADS provides over their existing training program. Therefore, we did not provide the control group with alternative programs of training on the same content.

The post-test was performed using ITADS. The Tutor Off mode of ITADS system was designed for performance assessment without any coaching in the form of feedback or hints. The post-test was administered in the Tutoring Off mode. In this mode, students are essentially working in a VM network, with one additional requirement of having to assert their diagnosis. An additional 'sim-ism' is the simulated Q&A with users reporting the fault listed in the trouble tickets. However, the Q&A is easy to use and often is not a significant component of a scenario. Thus, the Tutoring Off mode is a fair replication of a real-world assessment of troubleshooting performance. The post-test consisted of six scenarios delivered in a fixed sequence to all students. The same post-test was used for both groups. The following measures of performance were collected during this post-test: Exercise scores, Exercise transcripts, and Exercise completion times

The experimental group consisted of two batches with five students each. Students for the experimental group were selected from ongoing IT-A school classes. We planned for control group to have ten students who had completed the IT-A school training and were awaiting duty assignments. However, due to logistic and availability constraints, the evaluation was conducted with a control group of only five students. Participants in the experimental group spent ten days on the trainer, spread over nine weeks.

The control group received half a day of training on the use of ITADS (similar to the experimental group) and were given half a day to review didactic materials. They did not undergo any training with the ITADS simulations.

Assessment of student reaction was performed using surveys that were filled out by students at different stages of training. Additionally, a demographic questionnaire was administered at the beginning of the study. The control group was also administered a demographic questionnaire in the beginning and a satisfaction survey at the end of the segment.

To examine the hypothesis that ITADS is effective in teaching the knowledge and skills identified in the requirements, we compared the performance of the experimental and controls groups along these dimensions: 1. Post-test exercise scores, 2. Post-test exercise completion times, and 3. Successful completion rates on post-test exercises. On average the experimental group scored 19% higher than the control group. This difference is not statistically significant. In terms of post-test exercise completion times, on average the experimental group took an average of 18 minutes *less* than the control group to complete exercises, i.e. they were about 70% faster. This difference is statistically significant ($p<0.001$).

Given the control group's unfamiliarity with the tutoring side of ITADS (i.e. scoring metrics, rationale updates), we felt that comparing exercise completion rates would be fairer than comparing scores. A student is said to have successfully completed a troubleshooting exercise when they have successfully diagnosed the problem, completed and verified the fix, and entered a log. Note that the tutor allows students to "give up" in the diagnosis phase and continue with the fix. Thus, it is possible for students to successfully complete one or the other phase but not both. We compared the number of exercises that were completed by the students in two groups. There were three measures of completeness: successfully completing each of the diagnosis and fix phases, and successfully completing both. Table 1 shows these completion rates.

Analysis of the survey data showed that the trainees had a favorable reaction to ITADS. In particular, the trainees in the experimental group reacted very positively to the Probable Causes panel.

Table 1. Comparison of exercise completion rates

Average completion rate	Experimental Group	Control Group
Diagnosis phase	85%	63%
Fix phase	82%	53%
Both phases	62%	38%

4 Conclusions

Training with ITADS significantly and substantially improved the performance of the experimental group in comparison to the control group. The impact was more pronounced and significant on exercise completion times and rates than on scores. The experimental group scored on average 19% higher than the control group on the Capstone tests, though this difference is not statistically significant. They also completed the exercises about 70% faster. Finally, the experimental group students successfully completed all phases of troubleshooting exercises 63% more often on average than the control group. Future studies will study how well learners retain this knowledge after transitioning out of the schoolhouse.

While ITADS was primarily developed as a cost-effective tool to address critical job skills, it was also intended as a demonstration that effective intelligent tutors can be developed that address real-world considerations of cost-effectiveness, fast schedules, and formal systems engineering processes. The ITADS system was successfully developed in twenty-six months from requirements to validation, following strict systems engineering procedures. The results of the training effectiveness study indicate that the ITS also leads to significantly improved performance among Navy IT recruits in troubleshooting tasks.

5 References

1. CISCO (2012). IT Essentials 5.0, Chapter 6, Networks, Slides 63-69.
2. Folsom-Kovarik, J. T., Schatz, S., & Nicholson, D. (2011). Return on investment: A practical review of ITS student modeling techniques. *M&S Journal, Winter Edition*, 22-37.
3. 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.
4. 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).

6 Acknowledgments

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 Warfare Training (CIWT) IT A-School in Pensacola, FL.