Authoring plug-in tutor agents by demonstration: Rapid, rapid tutor development

Vincent Aleven and Carolyn Rosé
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA

Abstract We combined two existing methods for rapid tutor development: "plug-in tutor agents" [6] and an authoring tool suite (CTAT) that supports the creation of tutors "by demonstration" [2]. The combined approach, which has not been tried before, is suited for adding tutoring capabilities to an existing problem-solving environment, for example an off-the-shelf simulator. Connecting the components (i.e., the simulator and CTAT) requires programming but once that is done, "Pseudo Tutors" are created "by demonstration. Following this approach, we created plug-in Pseudo Tutor agents for a thermodynamics simulator, CyclePad [1], which were tried out in a classroom experiment involving 92 college students. The experiment demonstrates that the Pseudo Tutor technology is viable in a complex domain and that Ritter and Koedinger's protocol for the tool-tutor communication is suited for use in an authoring environment.

Introduction

It has long been recognized that it takes much time and effort to build an intelligent tutoring system. A number of approaches have been tried to bring down the development time and to make ITSs easier to develop. An approach that holds much promise is the use of "Plug-In Tutor Agents," [6], a way of adding tutoring to existing problem-solving environments or simulators (referred to as "tool") without having to build a complete tutor from scratch. Examples of this approach are the science learning space [3] and the Excel tutor [4]. A key contribution of this work is the specification of a protocol for the tool-tutor communication.

A different approach to rapid tutor development is the creation of authoring tools, which typically facilitate the development of the knowledge sources for an ITS. A wide array of tools have been developed and some have proven capable of supporting the development of effective tutoring systems [5]. The Cognitive Tutor Authoring Tools (CTAT) support the development of so-called Pseudo Tutors, which can be created without programming, namely, by demonstrating correct and incorrect solutions to tutor problems [2]. Pseudo Tutors are problem-specific, but in many domains they offer an attractive trade-off between development time and generality. Preliminary investigations indicate that it takes considerably less time to develop Pseudo Tutors than full-blown Cognitive Tutors [2].

Since authoring tools and plug-in tutor agents have complementary advantages, it seems natural to combine them. This poster describes the use of CTAT to add tutoring to the CyclePad thermodynamics simulator [1]. This AI-based system, shown on the right in Figure 1, lets students build and analyze thermodynamic "cycles" such as those underlying power plants, combustion engines, and refrigerators..

Developing Pseudo Tutors for CyclePad

To add tutoring to CyclePad, we hooked it up to CTAT (see also [6]): First, we made the simulator "recordable," so that it communicates all actions that the user takes in the simulator to CTAT, (more or less) conform to the protocol for tool-tutor communication laid out in [6]. This information enables the Pseudo Tutor to track students as they work through problem scenarios. This step was not especially difficult.

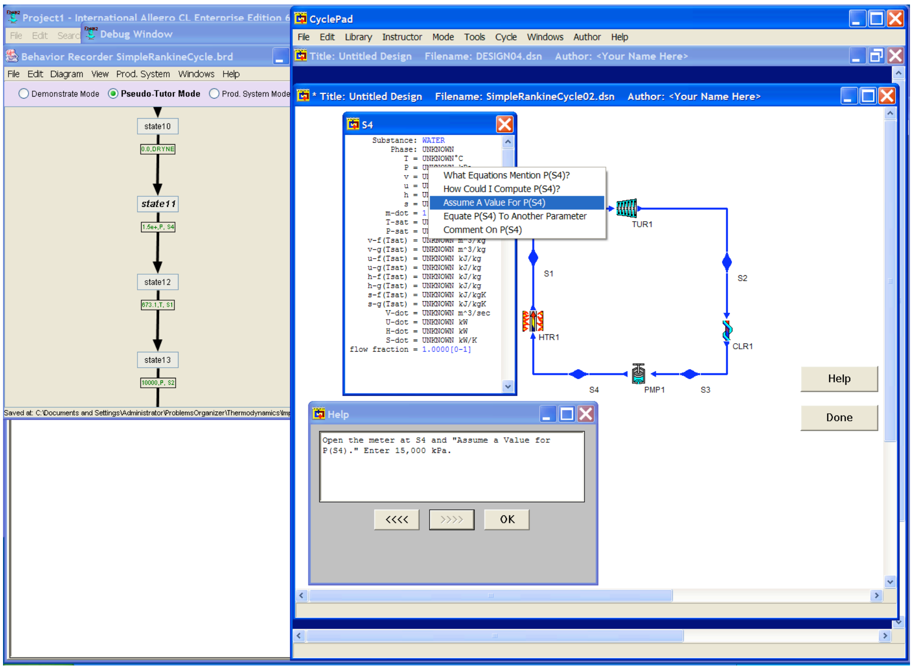

Figure 1: Using CTAT's Behacvior Recorder (left) to develop plug-in Pseudo Tutors for CyclePad (right)

Second, we made the simulator "scriptable," that is, capable of responding to scripting commands from CTAT. Scriptability is necessary so that the tutor can prevent incorrect student actions from being executed and prevent the student from undoing the effect of correct actions that have already been OK-ed by the tutor. Without this ability, the tutor may loose track of where the student is in the given problem scenario, which would lead to ineffective tutoring. It has been suggested [6] that the responsibility for undoing incorrect actions may be placed in the hands of the student, obviating the need for this kind of scriptability. However, we do not see that as a viable option for real-world tutors. Since CyclePad did not come with an undo facility, we implemented one ourselves, which required a fair amount of effort. The communication from tutor to tool roughly conformed to the protocol specified in [6].

Finally, we made the tutor visible in the CyclePad interface. This amounted mainly to adding a Help button and a messages window, in which CyclePad displays the tutor's hint and error feedback messages. With CyclePad hooked up to CTAT, we created Pseudo Tutors for scenarios in which students create, analyze, and optimize three different designs for the so-called Rankine cycle, a key thermodynamic design that is the basis for the steam-based power plants that generate the majority of the electricity in the US. These Pseudo Tutors were created in the usual manner, without programming: by demonstrating correct and incorrect steps, and annotating them with hints and feedback messages [2].

A classroom evaluation of the thermodynamics Pseudo Tutors

The thermodynamics Pseudo Tutor plug-ins were evaluated in a controlled experiment, carried out in an undergraduate thermodynamics course. The experiment (described more fully in [7]) involved 92 students, mostly sophomores, of whom 39 used the Pseudo Tutors. The goal of the experiment was to compare students' learning results in three different conditions in which they used the CyclePad simulator guided by, respectively, a script (on paper), Pseudo Tutor plug ins, and a human tutor in a Wizard of Oz scenario. There were reliable learning gains in all conditions including the Pseudo Tutor condition. The human tutor was reliably better than the other two conditions; the difference between the other two conditions was not reliable (t-test, $p > 0.2$). The Pseudo Tutor condition was plagued by a number of technical problems, none of which had to do with CTAT. Also, due to the severe time pressure under which we had to prepare for the experiment, we managed to provide Pseudo Tutors for only 2 of the 3 scenarios that the students explored with CyclePad. For these reasons, we do not consider the experiment to have been a fair test of the Pseudo Tutor technology. It did however constitute the first use of Pseudo Tutors in a university classroom and did show learning gains associated with the use of Pseudo Tutors.

Discussion and conclusion

By combining two approaches to rapid tutor development, plug-in tutor agents and authoring tools, we were able, in a short period of time, to add tutoring to CyclePad, a sophisticated simulation environment. Establishing the required communication between CyclePad and CTAT was not trivial, but neither did it require very extensive development effort. Most of the effort went into implementing an undo facility in CyclePad. The communication between CyclePad and CTAT was based roughly on the protocol presented in [6], which, originally developed to support tutoring, was by and large capable of supporting authoring as well.

Once CyclePad was hooked up, we developed tutoring capabilities without programming. A classroom experiment showed that Pseudo Tutors are a viable way to provide tutoring in a domain as complex as thermodynamics, even though, due to a number of technical problems, the gains were not as large as we had hoped. Given the time constraints under which the Pseudo Tutors were developed, we feel confident that we will be able to improve the Pseudo Tutors significantly. The experiment was to the best of our knowledge the first foray into a college classroom of a system based on the plug-in tutor principle. Further, while Pseudo Tutors have before been used before for high-school math, in a genetics college course, and in a number of on-line courses at Carnegie Mellon, the Pseudo Tutors for thermodynamics were the first controlled evaluation this technology in a college classroom. For those wanting to try out the development approach described here, the Cognitive Tutor Authoring Tools are available free of charge, for research and educational purposes, at http://ctat.pact.cs.cmu.edu.

Acknowledgments

This research is sponsored by ONR Award N000140410107 and an NSF Award SBE0354420 to the Pittsburgh Sciences of Learning Center. The contents of the paper are solely the responsibility of the authors and do not necessarily represent the official views of the ONR or the NSF. We thank Chang-Hsin Chang, Jonathan Sewall. Zhenhua Zhang, Chih-Yuh Chao, Christine Wilson, and Sandy Katz for their contributions.

References

[1] Forbus, K. D., Whalley, P. B., Evrett, J. O., Ureel, L., Brokowski, M., Baher, J., Kuehne, S. E. (1999). CyclePad: An Articulate Virtual Laboratory for Engineering Thermodynamics. *Artificial Intelligence 114*(1-2): 297-347.

[2] Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B., & Hockenberry, M. (2004). Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.), *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 162-174). Berlin: Springer Verlag.

[3] Koedinger, K. R., Suthers, D. D., & Forbus, K. D. (1999). Component-based construction of a science learning space. *International Journal of Artificial Intelligence in Education, 10*.

[4] Mathan, S., & Koedinger, K. R. (2003). Recasting the feedback debate: Benefits of tutoring error detecting and correction skill. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 11th International Conference on Artificial Intelligence in Education, AI-ED 2003.* Amsterdam: IOS Press.

[5] Murray, T., Blessing, S., & Ainsworth, S. (Eds.) (2003). *Authoring Tools for Advanced Technology Learning Environments: Towards cost-effective adaptive, interactive and intelligent educational software.* Dordrecht, The Netherlands: Kluwer.

[6] Ritter, S., & Koedinger, K. R. (1996). An architecture for plug-in tutor agents. *Journal of Artificial Intelligence and Education* 7(3/4), 315-347.

[7] Rosé, C. P., Aleven, V., Carey, R., & Robinson, A. (in press). A First Evaluation of the Instructional Value of Negotiable Problem-Solving Goals on the Exploratory Learning Continuum. *Proceedings of 12th International Conference on AI in Education, AIED 2005.*