**Creating an Intelligent Tutoring Agent for**

**Thermodynamics Equations**

**Anna Baglione1, David Strength2, José Virella Velázquez3,**

**Mostafa Amin-Naseri4, Enruo Guo4, Stephen Gilbert4, PhD**

*Iowa State University, Virtual Reality Applications Center, Ames, IA, 50011*

**Abstract.** Intelligent Tutoring Systems (ITSs) have been proven effective in enhancing students’ problem solving skills by generating individualized feedback and simulating one-on-one tutoring between a student and a domain expert. However, most ITSs do not emphasize the initial and most important step of problem solving: framing the problem. To address this issue, Jackman et al. (2011) have designed ITSs for two specific engineering domains: thermodynamics and statics. In this paper, we explain how we designed and implemented an intelligent equation tutor that develops problem framing skills and identifies conceptual errors by leveraging the existing thermodynamics tutor’s framework and the MathFlow equation editor. Our tutor parses a student’s equation, analyzes it to identify misconceptions the student may have, and delivers individualized feedback to the student based on this analysis. Based on an 82% agreement rate between our tutor’s feedback and an expert instructor’s feedback, we have demonstrated our tutor’s potential to help students build problem framing skills.

1. **Introduction**

Problem solving is a critical skill with which many engineering students struggle. Students have a tendency to memorize and reuse equations instead of fully understanding concepts because they may lack key problem framing skills. Framing a problem involves three important steps: drawing a diagram, writing equations, and updating the diagram based on the equations. A student who cannot frame a problem may have difficulty completing any or all of these steps. Furthermore, the student may be too embarrassed to ask for help from an instructor, who does not have adequate time to dedicate individual attention to each student in the classroom.

1The Ohio State University

2Belmont University

3University of Puerto Rico at Mayagüez

4Iowa State University

An intelligent tutoring system can meet both the student’s and the instructor’s needs by targeting misconceptions during the problem framing process and tailoring feedback to address these misconceptions.

Many tutors simply generate practice problems until the student recognizes and corrects the initial error. This approach not only prolongs the learning process for the student, but has also been proven ineffective for students who exceed approximately eight failed attempts. Once students pass this threshold, there is very little chance that they will solve the problem regardless of the number of additional practice problems provided (Beck & Gong, 2013). Our approach is to identify the source of a

student’s error and address this error with specific remedial exercises, such as a follow-up question or video. In this way, our tutoring agent helps students resolve misconceptions and develop strong problem framing skills.

1. **Related Work**

Existing tutors differ in three key areas: student input, tutor feedback, and system architecture. We examine these differences and discuss their merits for the purpose of informing the development of our own equation tutor.

Student and authoring components work cooperatively in an ITS to foster a flexible and individualized learning environment.  For example, the Andes physics tutor (Gertner & VanLehn, 2000) is modularly divided into two key systems: an authoring environment and student environment. A Bayesian network informs the student model, and a problem solver automatically generates a solution graph after the problem is defined. However, the student must install Andes as a standalone application; this requirement limits the locations where a student can access the tutor’s student model. Our equation tutor will be modularly designed to enable future use within other engineering domains, such as statics.  However, its distribution model will avoid the disadvantages of standalone applications because each student will always have access to his or her student model on a server. This distributed client-server architecture is based on Alpert et al.’s (1999) evaluations of previous web-enabled architectures for the purpose of converting an existing standalone ITS to the World Wide Web. Like Alpert et. al, we chose this architecture for its responsive user interface, centralized student model, and flexible communication.

Many tutoring interfaces favor prompt-based input. For example, Koedinger et. al’s (1997) PAT equation editor, like most equation tutors, is text-based and allows the student to type in their equations, then specify what operations will be applied to it. Albert et. al’s (1999) AlgeBrain provides students with pre-defined terms which they may select to include in their solutions. Mitrovic’s (2011) Thermo-Tutor asks the user to fill multiple blank text fields, each of which corresponds to a key term in the equation (e.g. pressure or work). Our tutor differs from others by requiring the student to write the entire equation in the input box, with no access to pre-defined terms. In this way, we encourage the student to develop effective problem framing skills, such as defining known and needed quantities, rather than relying on the tutor.

The vast majority of ITSs uphold individualized, immediate feedback as the most effective way to interact with and engage the student. A notable example of this kind of feedback is dialog-based tutoring.   Razzaq and Heffernan’s (2004) E-Tutor, a dialog-based system, went beyond simply accepting students’ steps passively to instead posing step by step questions in the form of dialog boxes.  Our tutor will adopt a similar approach by generating feedback through dialog windows. These windows will host a required sub-activity, such as a follow-up question or a video, to further gauge the student’s understanding of the course material in relation to the error. Our goal is to avoid the pitfall of delivering simplified, general feedback by instead drawing upon a strong expert model which guides the student on a structured error resolution path.

1. **Research Question**

The question we wanted to answer is: How closely could the feedback from an equation tutor agree with the one-on-one feedback of an expert instructor for undergraduate students in thermodynamics?

1. **Methods**

Our equation tutor is a distributed client server, which facilitates communication between the client and one or more servers over a network. The student submits an equation in a web application, shown in Figure 1 below, which passes the equation over a network via a TCP socket connection to the tutor server. In the tutor, a handler class reads the equation as a stream of bytes. The handler then converts the byte stream into a string, which contains MathML markup, and parses this string using the Document-Object-Model (DOM). Figure 2 below shows the overall architecture.

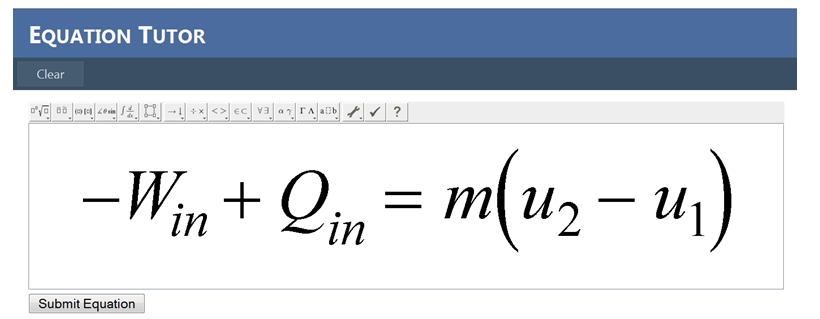


Figure 1: Equation Editor Interface

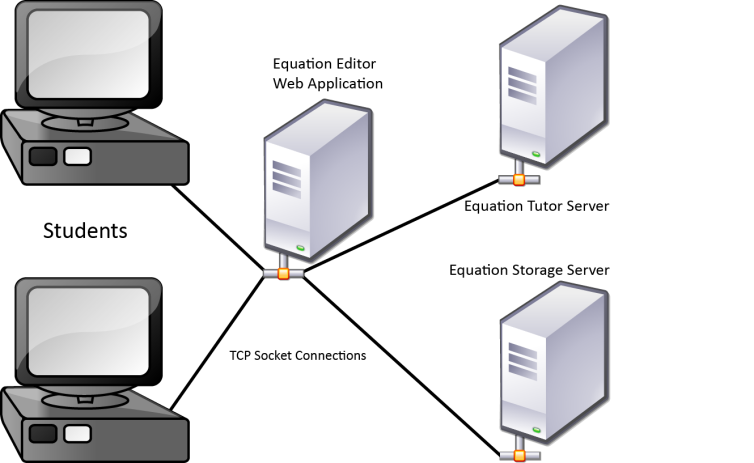


Figure 2: Tutor Architecture

MathML is a specialized version of XML composed of tags for mathematical operators, such as multiplication, division, and derivatives. It has two kinds of markup: presentation and content markup. We chose presentation markup because, using its linear structure in combination with the DOM, we were able to represent the tree hierarchy of MathML in XML on the receiving server, then traverse the tree to reconstruct the student’s equation using string concatenation based on MathML tag names.

Once our system parses the student’s equation to a string, we run the string through a series of methods which analyze the string’s components and compare them with the expert model. The tutor generates an error tag when it identifies a discrepancy between the expert model and the student’s submission. The handler then passes this error tag to another server which triggers the remedial activity.

To develop the expert model, we worked closely with an expert instructor to identify and categorize common student errors. We then established a priority order in which our tutor checks the student’s equation for these errors. Figure 3 below shows the structure of the expert model:

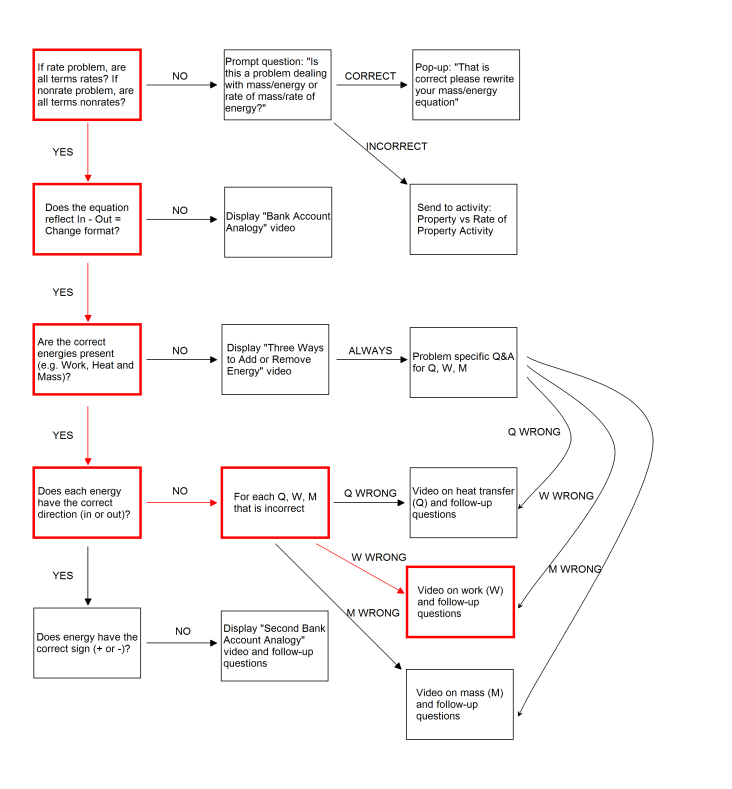


Figure 3: Expert Model

One example of a common student error involves change in energy, an important concept in thermodynamics. Our expert instructor’s pedagogical approach is to teach students that the difference between the energy entering a system and the energy leaving a system is the change in energy. The instructor maintains that students whose equations do not clearly communicate this concept should be redirected. Thus, our tutor uses change in energy as a high priority check when evaluating a student’s equation.

1. **Testing/Results**

For our preliminary analysis, we tested 92 student equations from previous thermodynamics exams and recorded the feedback our tutor generated for each. We then asked our expert instructor to provide feedback for the same equations and rank the agreement of our tutor’s feedback with his own. Figure 4 below shows that our tutor agreed or strongly agreed with the instructor’s feedback approximately 86% of the time.

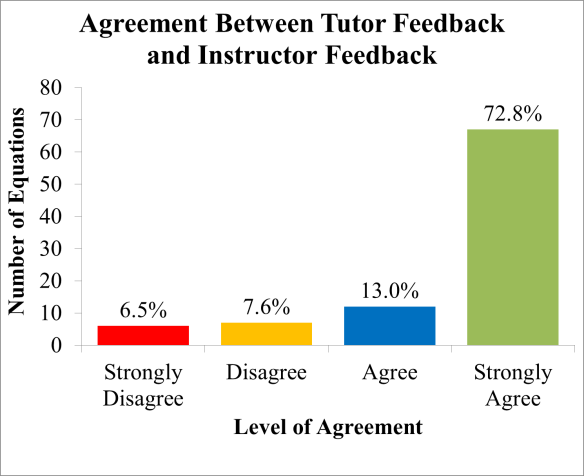


Figure 4: Preliminary Analysis of Feedback Agreement

After gathering these preliminary results, we conducted an objective analysis in which the instructor indicated the level on our state diagram representation of the expert model on which he would flag the student for an error. Figure 5 below shows that tutor agreed with the instructor’s feedback approximately 82% of the time.

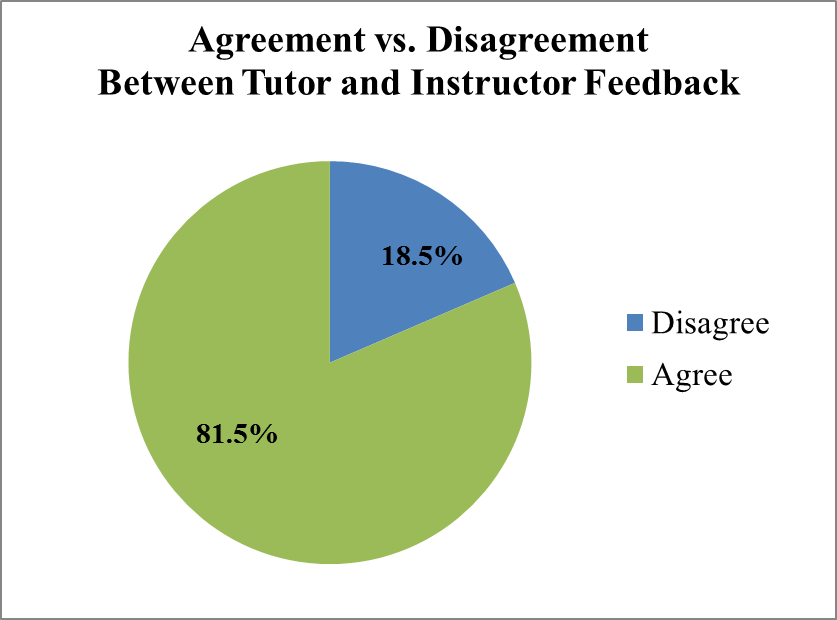


Figure 5: Objective Analysis of Feedback Agreement

Sources of disagreement in both analyses included feedback which was too general and logical errors in our code. Additionally, instructor modifications to the structure of the expert model rendered a portion of our feedback data invalid and thus reduced our sample size to 54 equations.

1. **Conclusions / Future Work**

We have discussed how we built and implemented an intelligent tutoring agent for thermodynamics equations. Future work will include revisions to existing methods and to the structure of our expert model. One of the major challenges we encountered in the development process was dealing with logic errors in our code that misinterpreted the student’s original equation. Revising our methods so that they are more robust will allow us to correct these errors, restructure the expert model based on our expert instructor’s recommendations, and deliver more granular feedback.

Another important feature we would like to incorporate is an authoring tool. By allowing instructors to specify which checks the expert model should contain, the order in which the tutor should execute them, and the remedial branching each check would deliver, this tool would accommodate different pedagogical approaches.

Finally, we plan to develop comprehensive student model that tracks a student’s knowledge and progress over time. This model would enable our system to tailor feedback to a student’s individual needs, thus better equipping the student with the problem framing skills necessary for addressing real world problems.

**References**

1. Alpert, S.R. et. al., (1999). Deploying Intelligent Tutors on the Web: An Architecture and an Example. *International Journal of Artificial Intelligence in Education*, 10, 183-197.
2. Beck, J.E. & Gong, Y. (2013). *Wheel-Spinning: Students Who Fail to Master a Skill. In* K. Yacef et al. (Eds.)., *Artificial Intelligence in Education: Proceedings of the 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013* (pp 431-440). Berlin Heidelberg: Springer.
3. Gertner, A.S. & VanLehn, K. (2000). Andes: A Coached Problem Solving Environment for Physics. In Gauthier et. al, *Intelligent Tutoring Systems:* *Proceedings of the 5th International Conference, ITS 2000 Montréal, Canada, June 19–23, 2000 (pp 133-142).* Berlin Heidelburg: Springer.
4. Jackman, J. et. al (2011), Problem Framing Skills in Engineering Problem Solving, 2011 NSF Engineering Education Awardees Conference, Reston, VA, March 13-15, 2011.
5. Koedinger, K.R. & Anderson, J.R. (1997). Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education, 8, 30-43.*
6. Mitrovic, et. al (2011). Thermo-Tutor: An Intelligent Tutoring System for Thermodynamics. In *Learning Environments and Ecosystems in Engineering Education: Proceedings of the* IEEE Global Engineering Education Conference (EDUCON), Amman, Jordan, April 4-6, 2011 (pp. 378-385). IEEE.
7. Razzaq, L. M. & Heffernan, N. T. (2004). Tutorial Dialog in an Equation Solving Intelligent Tutoring System. In J.C. Lester, et. al, *Intelligent Tutoring Systems: Proceedings of the 7th International Conference, ITS 2004, Maceió, Alagoas, Brazil, August 30 - September 3, 2004* (pp 851-853). Berlin Heidelburg: Springer.
8. Zhou, Y. et. al. (1999). Delivering Hints in a Dialogue-Based Intelligent Tutoring System*.* In *AAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence, Orlando, FL, July 18-22, 1999* (pp 128-134). Menlo Park: American Association for Artificial Intelligence (AAAI).