Thermo-Tutor: An Intelligent Tutoring System for Thermodynamics

Antonija Mitrovic, Chris Williamson, Aidan Bebbington, Moffat Mathews, Pramuditha Suraweera, Brent Martin, David Thomson, Jay Holland

Intelligent Computer Tutoring Group
University of Canterbury
Christchurch, New Zealand

Abstract— **We present the design and an evaluation of Thermo-Tutor, an Intelligent Tutoring System (ITS) that teaches thermodynamic cycles in closed systems. Thermo-Tutor provides opportunities for students to practice their skills by solving problems. When a student submits a solution, Thermo-Tutor analyzes it and provides appropriate feedback. We discuss the support for problem solving, and the student model the ITS maintains. An initial evaluation of Thermo-Tutor was performed at the University of Canterbury. The findings show that the ITS supports student learning effectively.**

Keywords- ***teaching thermodynamics; intelligent tutoring systems***

I. Introduction

Thermodynamics, like many other engineering topics, is hard to master for novices. In addition to having to understand the theory, the student also needs a lot of practice. Education research has shown that the best learning environment is one-to-one with an expert human teacher [1]. However, the current situation in education falls a long way short of this ideal. Even in the best funded institutions, the student-to-teacher ratio is not sufficient to make this feasible. Teachers are therefore forced to suit their teaching to the average student. High achieving students may become bored and frustrated by the slow pace and lack of challenge and so fail to achieve their best. Low achieving students find the work difficult and never receive the level of attention they require.

Intelligent Tutoring Systems (ITS) are knowledge-based systems that simulate the behaviour of human teachers [2, 3, 4]. ITSs aim to provide an experience similar to personal tutoring without the need for human intervention. An ITS typically provides a problem-solving environment, in which students are given many opportunities to practice their skills. The ITS collects information about the student's actions, analyzes them and maintains the model of the student's knowledge. This student model then allows the ITS to adapt the instructional actions to the skills and abilities of each individual student. Adaptation is mostly done in terms of providing feedback, selecting or generating problems at the right level of complexity, or deciding which topic to teach.

In this paper, we present Thermo-Tutor, an ITS that teaches thermodynamic cycles in closed systems (the first law of thermodynamics). Thermo-Tutor is designed as a complement to traditional courses: it assumes that the students have already learned the basic concepts in lectures, and therefore it provides lots of problem-solving opportunities. Students can select problems to work on, and submit their solutions to be checked whenever they want.

We start by presenting the related work in Section 2, and then discuss the features of Thermo-Tutor in Section 3. We evaluated the ITS in a real classroom, and present the results of that study in Section 4. The last section presents the discussion and future work.

II. Related Work

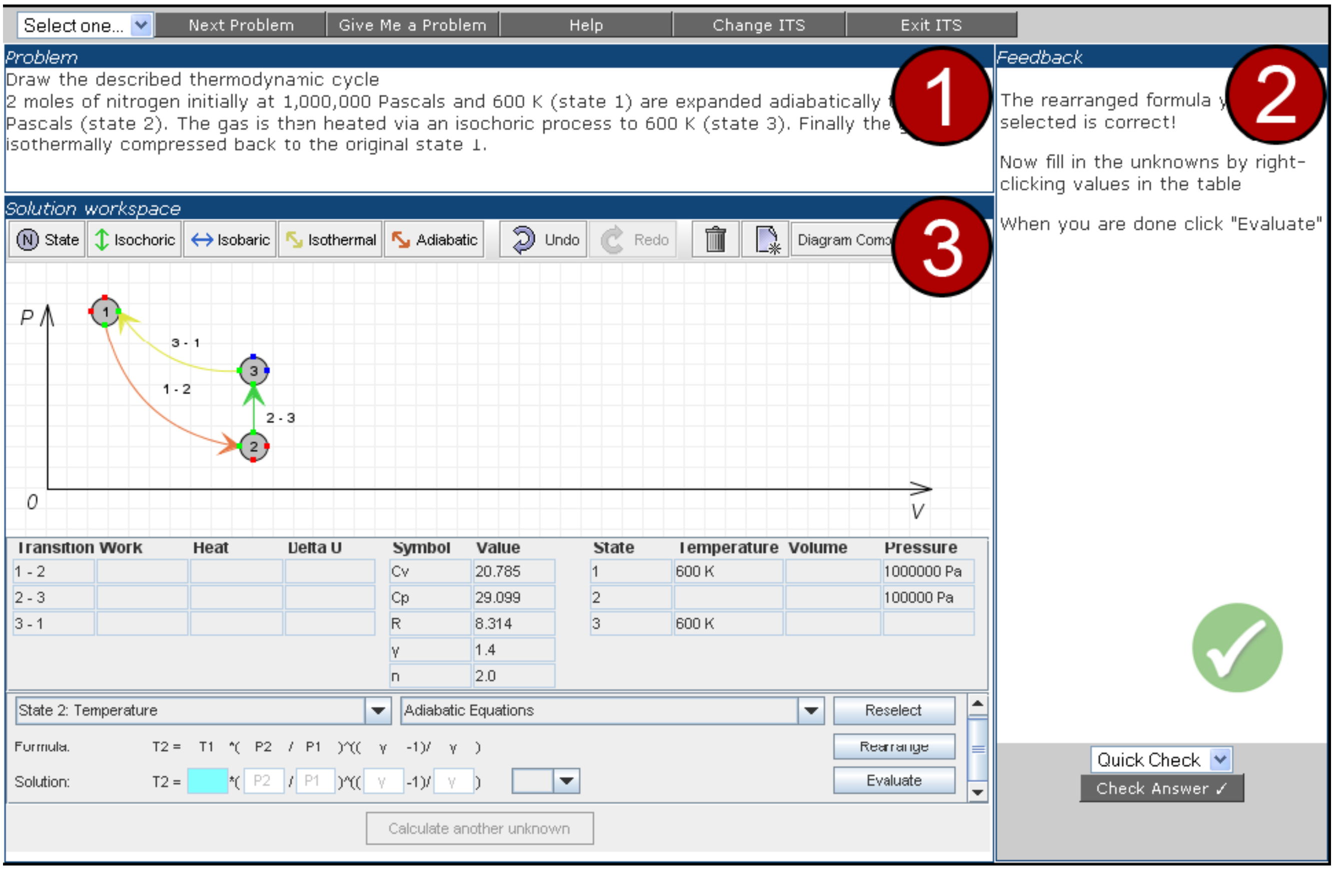
ITSs have proven their effectiveness not only in controlled lab studies, but also in real classrooms [2, 3, 5, 6, 7]. These computer-based tutors achieve significant improvements in comparison to classroom learning due to their fine-grained knowledge of the instructional domain, their ability to employ various pedagogical strategies, and their student modelling capabilities which enables individualized instruction. However, ITSs still have not achieved widespread effect on education due to their high complexity and difficulty of development. Composing the domain knowledge required for ITSs consumes the majority of the total development time [8]. The task requires multi-faceted expertise in knowledge engineering, AI programming, and the domain itself.

The Intelligent Computer Tutoring Group (ICTG)[1] has developed a number of successful constraint-based tutors over the years in various design tasks such as SQL queries [9], database design [10, 11, 12], software analysis and design using UML [13, 14], programming in Java [15], English/vocabulary skills [16, 17] and others. We have also developed several ITSs for procedural skills, such as data normalization within relational database design [18], logical database design [19] and capital investment [20]. Many of these ITSs are used in courses and individually by students around the world.

Although constraint-based tutors are easier to develop in comparison to some other existing types of ITSs [21], their development is still a labour-intensive process that requires expertise in Constraint-Based Modelling (CBM) [22, 23] and programming. Constraint-Based Modelling is a methodology for creating a domain model (the knowledge base) and consequently, modelling each student's knowledge within that domain. When developing a constraint-based tutor, the single biggest task is creating the constraints that make up the

[1] www.ictg.canterbury.ac.nz

knowledge base. Each constraint specifies a domain principle and is written in such a manner that it can be used by the tutor to evaluate a student's solution. The constraints can relate to both semantic and syntactic principles of the domain. The difficulty of creating this domain model can seem even greater to a domain expert who has little programming expertise.



Draw the described thermodynamic cycle
2 moles of nitrogen initially at 1,000,000 Pascals and 600 K (state 1) are expanded adiabatically t
Pascals (state 2). The gas is then heated via an isochoric process to 600 K (state 3). Finally the g
isothermally compressed back to the original state 1.

The rearranged formula y
selected is correct!

Now fill in the unknowns by right-clicking values in the table

When you are done click "Evaluate"

Transition	Work	Heat	Delta U
1 - 2			
2 - 3			
3 - 1			

Symbol	Value
Cv	20.785
Cp	29.099
R	8.314
γ	1.4
n	2.0

State	Temperature	Volume	Pressure
1	600 K		1000000 Pa
2			100000 Pa
3	600 K		

State 2: Temperature
Adiabatic Equations

Formula: T2 = T1 *(P2 / P1)^((γ -1)/ γ)

Solution: T2 = *(P2 / P1)^((γ -1)/ γ)

Figure 1. Thermo-Tutor Interface Layout

In order to reduce the time and effort required for producing constraint-based tutors and to make this process more accessible to and inclusive of domain experts, we developed ASPIRE, an authoring system that can generate the domain model with the assistance of a domain expert and produce a fully functional ITS. ASPIRE guides the author through a semi-automated process to develop the domain model, and allows the author to deploy their tutor on the web, with all necessary modules included automatically. ASPIRE allows a domain expert (i.e. a teacher) with little or no programming background to create an ITS for their domain. We previously evaluated ASPIRE by comparing the constraints it automatically generated to those manually created by domain experts in two well-established ITSs namely, KERMIT [9, 10] and NORMIT [9, 11]. Evaluation showed that ASPIRE can generate all 35 syntax constraints that were developed manually for KERMIT, and all but two of the 21 syntax constraints that were developed for NORMIT [24]. It was also found that ASPIRE could generate 90% of the 125 entity-relationship modelling semantic constraints found in KERMIT, and the complete set of 45 semantic constraints that exist for NORMIT. Using an authoring system such as ASPIRE helps alleviate many common issues encountered in ITS development such as the author failing to properly understand the domain, and communication failures between the domain expert and the author.

III. Thermo-Tutor

Thermo-Tutor teaches thermodynamic cycles in closed systems. It does this by providing students with many opportunities to solve problems while receiving customised feedback on their solution. It is not intended to cover the entire course, nor the homework component of an entire course. Instead, Thermo-Tutor complements a particularly troublesome section of the course and helps students to get started in thermodynamics.

When designing the user interface for Thermo-Tutor, one of the goals was to make the students' experience correspond as closely as possible to how they would solve such problems on paper. We felt that this was important as it would not only make the interface more natural and easy to use, but it would also support transferability of the problem-solving skills students gain from the ITS to solving similar problems outside the tutor. The Thermo-Tutor user interface is shown in Figure 1. At the top of the interface, students are provided with the controls for selecting problems, asking for help, or exiting the tutor. The rest of the interface focuses on the current problem with dedicated space for the statement of the problem (1), feedback (2), and solution workspace (3).

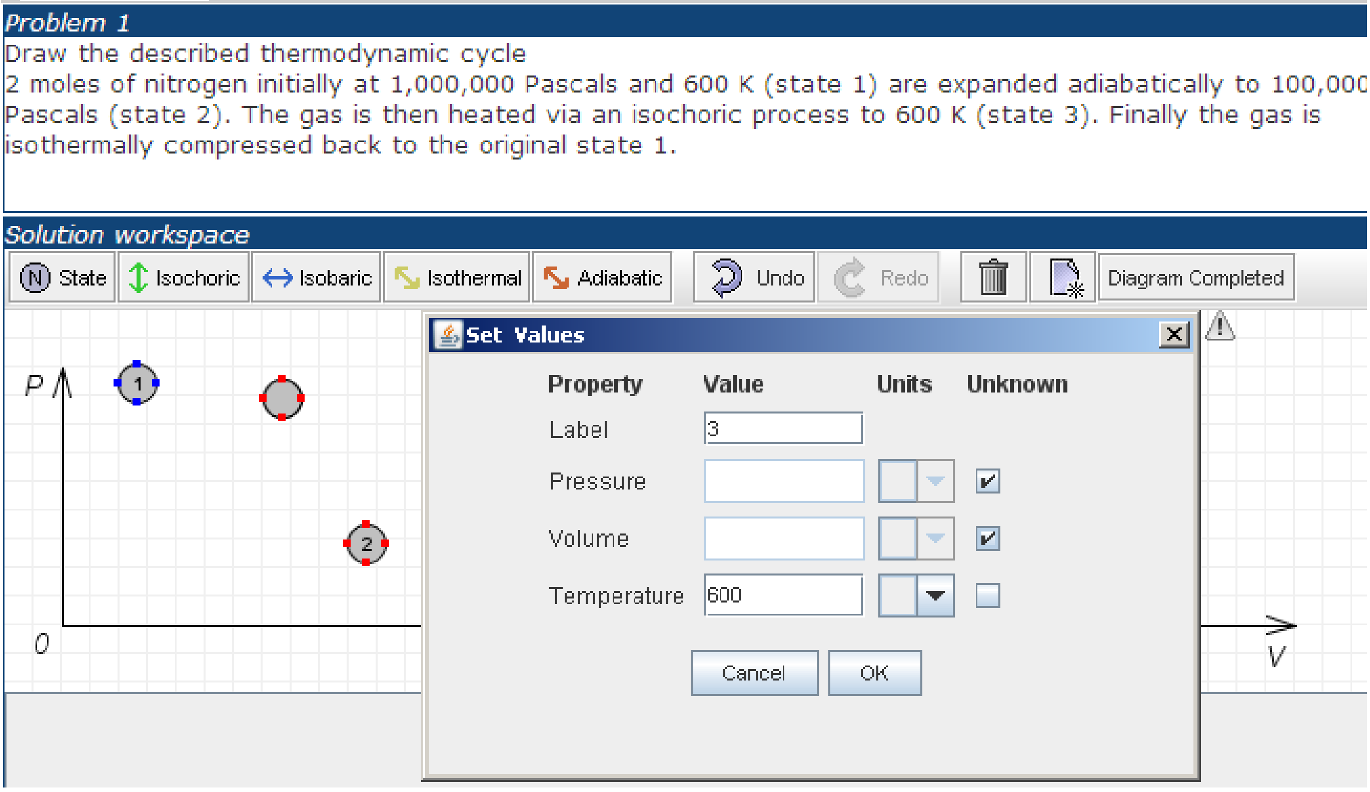
The student starts by selecting a problem, and after reading the problem text progresses to creating their solution within the solution workspace. A solution can be submitted at any stage for evaluation; the system provides feedback on the submitted solution in the feedback pane. We aimed to make Thermo-Tutor suitable for solving problems related to cycles in closed systems. Solving such problems involves two main phases: drawing the diagram, and later calculating unknowns.

When drawing the diagram, the student needs to examine the problem statement in order to identify all states and transitions between states, as well as to specify the properties of states and transitions. A state is a point where the system is at equilibrium under a given set of conditions. Each state is described by three properties; pressure, volume and temperature. The relationship between these properties is defined by the ideal gas law such that if two properties are known then the third can be calculated without any additional information.

A diagram of a thermodynamic cycle shows the states and transitions with axis of pressure versus volume. In Thermo-Tutor, diagrams do not include scales on the axis as they are used primarily to show relative positions of states. The tutor allows the student to create a diagram of a thermodynamic cycle with a simple point-and click style interface. To draw a state, the student needs to select the state icon in the tool bar, and click on the position within the diagram area where the state should appear. Thermo-Tutor opens the *Set Values* window, as illustrated in Figure 2, which allows the student to specify the properties of the state. In the situation illustrated in Figure 2, the student has already specified states 1 and 2, and is in the process of specifying the properties of state 3. The student first specifies the label of the state. The temperature for state 3 is given in the problem statement (600K). The pressure and volume are unknown, and the student needs to tick the *Unknown* boxes, to acknowledge that these properties are to be calculated later.

Some properties of states may not be explicitly given by the problem statement, but instead must be inferred by the student using their knowledge of thermodynamics. For example, an isobaric transition is one in which the pressure remains constant. In this case the student may only be given the pressure of the start state; the student would then be required to infer that the pressure of the end state is the same as the start state. During this first phase the student will not be required to carry out any calculations. All the information required to complete this phase comes directly from the problem statement.

A transition (i.e. a process) describes the way the system changes from one state to another. Transitions are either distinguished by what remains constant throughout the transition (pressure, volume or temperature) or are adiabatic. To draw a transition, the student needs to select the icon that corresponds to a particular transition type, click on the starting state and drag to the destination state. Thermo-Tutor then draws the transition between (and connecting) the two states, unless there are errors. For example, in the situation shown in Figure 2, if the student selects the isochoric transition and tries to draw it between states 2 and 3 positioned as shown in this figure, Thermo-Tutor will not draw the transition. It will



Problem 1

Draw the described thermodynamic cycle

2 moles of nitrogen initially at 1,000,000 Pascals and 600 K (state 1) are expanded adiabatically to 100,000 Pascals (state 2). The gas is then heated via an isochoric process to 600 K (state 3). Finally the gas is isothermally compressed back to the original state 1.

Solution workspace

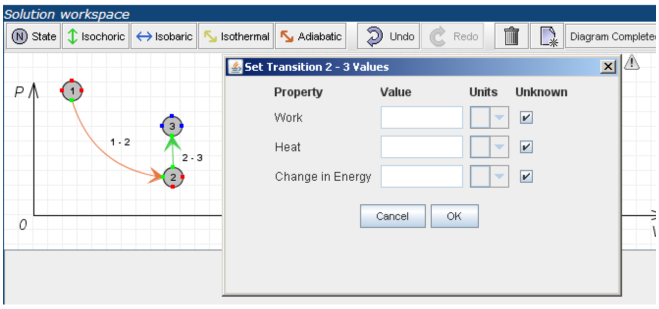
Set Values

Property	Value	Units	Unknown
Label	3		
Pressure			☑
Volume			☑
Temperature	600		□

Figure 2. Specifying the properties of a state

instead remind the student that an isochoric transition is a process under constant volume, and therefore such transition needs to be parallel with the p axis. Thermo-Tutor achieves this by comparing the student's action (drawing an isochoric transition) to constraints, which specify that isochoric transitions mean constant volume.

If the student draws the transition correctly, he/she can then specify its properties: work, heat, and the change in energy (Figure 3). The student can reposition states and transitions, as well as change their properties at any time.

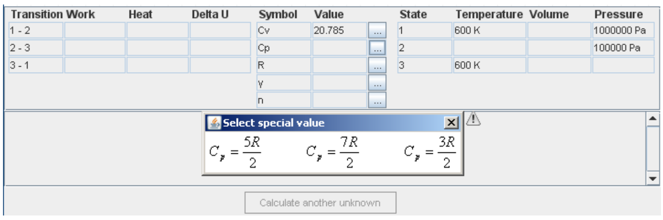


Set Transition 2 - 3 Values

Property	Value	Units	Unknown
Work			☑
Heat			☑
Change in Energy			☑

Figure 3. Specifying the properties of a transition

Students can submit the diagram for checking whenever they want. The tutor evaluates the student solution against the constraints in its knowledge base. Any constraint that is violated represents an error in the student's solution. Feedback of varying levels is provided, ranging from Error Flag, to Full Solution. On the first submission, the feedback is general and only informs students whether there are any errors in the solution. The feedback level increases on each incorrect or incomplete submission. After the second submission, the student is informed of the location of the error within the solution (*Error Flag*); for example, the student might be told that one of the states is specified incorrectly. On the third submission, Thermo-Tutor provides more detail about the error itself (the *Hint* level). Automatic progression of feedback levels stops at this point. The additional two levels provide hints about all mistakes the student made (*All Errors*) or provide the solution. Please note that the student might explicitly request a specific level of feedback by selecting it from the drop-down box.



Transition	Work	Heat	Delta U
1 - 2			
2 - 3			
3 - 1			

Symbol	Value
Cv	20.785
Cp	
R	
γ	
n	

State	Temperature	Volume	Pressure
1	600 K		1000000 Pa
2			100000 Pa
3	600 K		

Select special value

$C_p = \frac{5R}{2}$ $C_p = \frac{7R}{2}$ $C_p = \frac{3R}{2}$

Figure 4. Specifying the formula for the Cp constant

Once the diagram is correct, the student is allowed to move to the next phase to calculate the unknowns. The tutor shows the summary of all the properties the student specified for states and transitions in tables below the diagram, as illustrated in Figure 4. The tutor then asks the student to specify the values for the constants needed for various formulae. In Figure 4, the student has specified the value for C_V (heat capacity at constant volume) and is currently working on the values for Cp (heat capacity at constant pressure). The student can simply select one of the formulae, and the corresponding value is entered automatically into the table. This step is included so that the data summary displays all information that the student could possibly need for subsequent calculations in the next step.

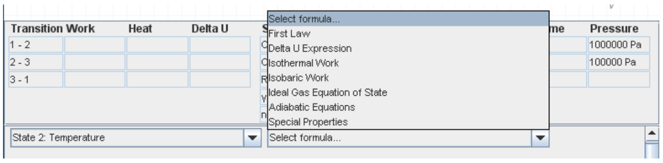
The student now needs to build upon what he/she knows about the cycle until all properties of all transitions and states are known. There are many formulae used during this phase and there are usually many ways of determining each property. It is often necessary to calculate certain properties in some order, but there are also usually several properties which can be calculated at any one point in the problem solving process. The set of properties which can be calculated is dependent on the properties that are currently known, and the types of transitions present in the cycle. For example, the change in internal energy of a transition can often not be calculated until the temperatures at both the initial and final states are known, and some formulae are only valid for isothermal transitions. In general, it is necessary to determine at least two properties of each state, and from them it is possible to determine all properties of transitions. However, it is sometimes sufficient to determine just one property of some states. This is because some formulae which are related to specific types of transitions implicitly contain information about these types of transitions.

Calculating unknowns on paper involves using formulae to gradually build upon what is known about the cycle. Some of the formulae involve exponents and logarithms, which make it difficult to design an interface which is easy to use with a mouse and keyboard. Many people find entering mathematical expressions with a keyboard cumbersome, and for good reason. The method of entering expressions via a keyboard differs significantly from the usual method of using a calculator.

In order to facilitate this process in Thermo-Tutor, we have broken it into the following steps:

1. Select the unknown to calculate
2. Select the formula group to use
3. Select the formula to use from this group
4. If necessary, rearrange formula by selecting from alternatives
5. Fill in all variables in the equation and specify the units of the result
6. Evaluate the equation to get the value for this unknown

The interface firstly requires the student to select the unknown they are going to calculate; the student achieves this by clicking the *Calculate another unknown* button. Following



Transition	Work	Heat	Delta U
1 - 2			
2 - 3			
3 - 1			

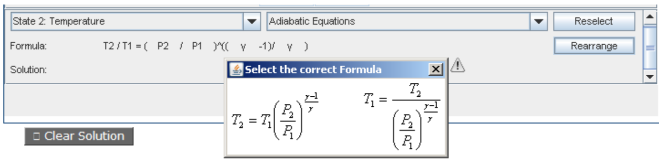
me	Pressure
	1000000 Pa
	100000 Pa

Figure 5. Selecting the formula group

this, the student selects an unknown from the list of all unknown properties for the given cycle. The student then needs to select a formula group, as shown in Figure 5. Thermo-Tutor shows different groups of formulae each time.

After selecting the formula group, a dialog window displays the formulae in that group. The student must select the correct formula from this dialog. Each group contains a number of formulae, some of which are valid under some circumstances, and some which are invalid.

The student may use the formula as is, but sometimes it is necessary to re-arrange the formula. Instead of asking the student to type the correct statement of the formula, Thermo-Tutor provides the *Rearrange* button, which shows all possible re-arrangements of the formula and allows the student to simply select the one that is appropriate. For example, Figure 6 shows a situation in which the student has selected an adiabatic equation which shows the ratio of the start and end temperatures. Because the student wants to calculate the temperature for state 1, he/she needs the version of this formula which has T_2 on the left-hand side.



State 2: Temperature
Adiabatic Equations

Formula: T2 / T1 = (P2 / P1)^((γ -1)/ γ)

Solution:

Select the correct Formula

$$T_2 = T_1\left(\frac{P_2}{P_1}\right)^{\frac{\gamma-1}{\gamma}} \qquad T_1 = \frac{T_2}{\left(\frac{P_2}{P_1}\right)^{\frac{\gamma-1}{\gamma}}}$$

Figure 6. Rearranging the formula

After selecting the correct version of the formula, the student is shown an equation editor, as illustrated in the lower part of the solution space in Figure 1. The student must provide values for all the variables on the right hand side of the equation, and select a unit for the result. The interface provides a textbox for each variable in the formula. The student must fill in each of these textboxes by selecting the value from the data summary. Values can be copied from the data summary into the currently selected textbox by double-clicking or right-clicking. If the student specified all the values correctly within the formula, he/she is required to enter the numerical value, which is then copied into the summary table cell for the unknown being calculated. The student continues calculating unknowns until all cells in the data summary table are correctly filled.

Thermo-Tutor was developed in ASPIRE, our authoring system and deployment environment for constraint-based tutors. ASPIRE is written in Allegro Common Lisp, runs on the AllegroServe web server. It also makes extensive use of JavaScript for the web interface it provides, and XML for transmission and temporary storage of data. ASPIRE uses the AllegroCache Object Oriented database for storing the majority of its operational data. The interface of Thermo-Tutor is provided by a Java applet which communicates with ASPIRE via XML over HTTP. Due to space restriction, we do not provide details of ITS development using ASPIRE, but instead refer the interested reader to [20, 25].

IV. Evaluation Study

In May 2010, we performed a small evaluation study of Thermo-Tutor with students enrolled in ENCH291, a second-year course on Chemical Process Technology within the Chemical and Process Engineering degree at the University of Canterbury. Participation was voluntary, and we had 22 volunteers. The students were given a short explanation on how to use Thermo-Tutor, and were then left to use it by themselves. The study was done during a 50-minute lab for the course.

The students used the tutor for an average of 37 minutes. Thermo-Tutor collected information about all actions students performed, and stored them in logs. The students were also asked to complete a questionnaire, providing subjective views on Thermo-Tutor.

On average the students solved 1.5 problems, and attempted 1.8 problems. As this was the first time using the ITS, and due to the relatively large problem size and short length of the session, this was deemed an acceptable rate. Table 1 summarizes the data obtained from the logs.

From Table 1 we can see that although on average only 1.8 problems are attempted, it can take many attempts to solve a problem. However, because each problem is divided into multiple steps, and an attempt is required at each step, this figure does not represent 71 whole incorrect solutions. It is encouraging that the number of solved (completed) problems is close in value to the number of attempted problems. This means that students are eventually solving the problem, rather than giving up and moving on. Out of 233 constraints in the tutor, the students used 154 on average. The number of constraints used ranges from 57 (for a student who only attempted one problem and did not complete it), to 186 (in the case of a student who completed two problems).

TABLE I. Log Information

Thermo-Tutor Log Information – mean (SD)	
Solved problems	1.45 (0.67)
Attempted problems	1.77 (0.81)
Total attempts	71.23 (26.25)
Total time (min)	36.95 (7.88)
Constraints used	154.64 (28.79)

We also analyzed students' learning in terms of constraints. This information was obtained from the student models. In Thermo-Tutor (as in other constraint-based tutors), the student model contains a history of usage for every constraint the student used. The history states whether the constraint was satisfied (i.e. used correctly) or violated on each attempt. This allows analysis of students' learning over time. We plot the probability of violating a constraint on each occasion when it was used, averaged over all constraints and all participants. If over time the probability of constraint violation decreases, we can say the students have learned the constraints [9, 22].

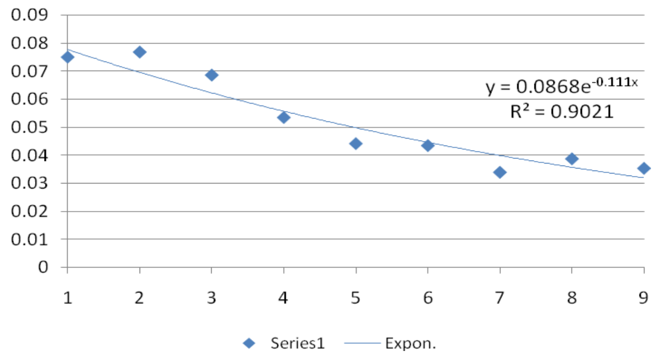


Figure 7. Learning curve for all constraints

Fig. 7 shows the learning curve for all constraints and all participants. Please note that the cut-off point (the ninth occasion) was chosen to correspond to the 50% of the number of constraints at occasion 1. As students used the tutor, there is a large and consistent decrease in the overall error rate (corresponding to the increase in learning), from 7.5% to 3.5% on the ninth opportunity of using constraints; i.e. students learned the domain principles they used. Furthermore, the fit of the learning curve illustrates the variability of the data. In other words, it shows how well students used what they learned previously. Over time (i.e. over multiple learning opportunities that are different to each other but use similar concepts), this allows us to gauge the transferability of the skills they learn; here we mean transferability of skills within the same domain but to different contexts. As time in this study was limited to 50 minutes, students did not get ample opportunity for us to gauge transferability of the skills they learned. In spite of this, the learning curve has a very good fit (R^2=0.9), suggesting that already students were not only learning the domain knowledge, but learning it well enough to use it consistently more correctly on future attempts, which could be in different steps of the same problem.

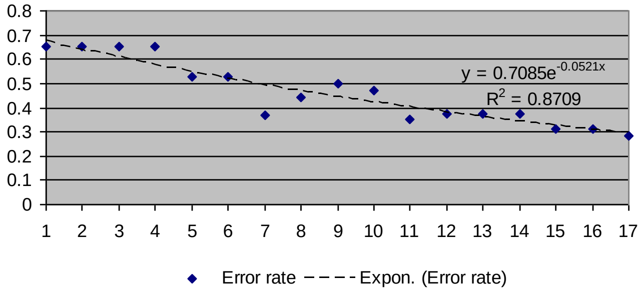


Figure 8. Learning curve for constraint 60

Deeper inspection of the student models revealed that some constraints were being satisfied immediately, i.e. the first time the student came across those constraints. When this is the case, no learning has taken place as the student has not moved from an erroneous state to a correct state in terms of the constraint. Of the 233 constraints in the tutor's knowledge base, 27 constraints were found to be in this situation. This resulted was accounted for on manual inspection , as there are constraints that check basic, yet important problem-solving concepts, such as whether the student has specified the answer for a particular step, or whether the type of value entered was consistent with what was expected. As the participants were second year university students, it is not surprising that such constraints were satisfied immediately.

Inspection of the domain principles that were difficult for students (i.e. constraints that were violated) sheds more light on the learning process at a finer detail. For example, the constraint which was violated most often on the first occurrence was constraint 60. Without delving into an explanation of Thermodynamics here, this constraint evaluates the student's knowledge when solving for unknowns in the case that pressure is not known. Constraints have varying levels of granularity, and this constraint is a broader "catch-all" within this specific context. When a student violates this constraint, they will get a broad feedback message alerting them to the fact that one or more of the values they have entered into their solution is incorrect. Other finer-grained constraints would expose more detail to the student in the feedback about specific errors, particularly at higher levels of feedback. Fig. 8 shows the learning curve for constraint 60. It can be seen that students are initially experiencing some problems with this constraint (the first four occasions of use have the same error rate), but later on the error rate decreases. There is an obvious trend downwards as time increases. The fit of the curve is good considering the low number of occurrences (n=20 for t=1).

The constraint which was violated the second most number of times on the first occurrence is constraint 86. This constraint focuses on the student's knowledge of formulae when work is unknown. Fig. 9 shows the learning curve for this constraint. Although the fit of this curve is not high, the downward trend in errors indicates that students are learning this particular constraint.

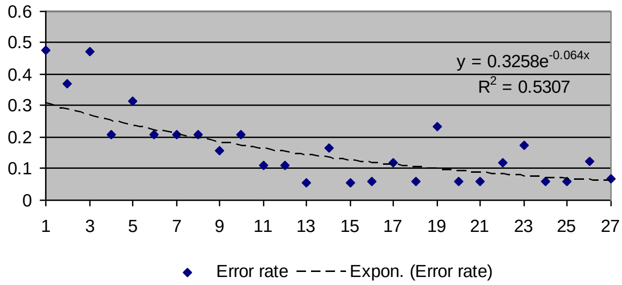


Figure 9. Learning curve for constraint 86

A constraint is considered *learned* when after some initial violations the student satisfies the constraint on future occurrences. Table 2 presents the nine most frequently learned constraints. The constraints themselves are quite complicated so the Table shows the problem-solving phase, specific step, and the knowledge evaluated to indicate the purpose of the constraint.

TABLE II. LEARNED CONSTRAINTS

Constraint	*Learned* Count	Phase, step, knowledge evaluated
60	11	Calculating unknowns. Pressure unknown. Knowledge of what values to use when solving this equation
86	8	Calculating unknowns. Work unknown. Knowledge of appropriate formulae.
74	8	Calculating unknowns. Initial energy unknown. Knowledge of appropriate formulae.
3	8	Drawing diagram. Assigning properties to states. Knowledge of what properties relate to each state and whether they are known
80	7	Calculating unknowns. Volume unknown. Knowledge of appropriate formulae.
71	5	Calculating unknowns. Heat unknown. Knowledge of appropriate formulae.
55	5	Calculating unknowns. Energy unknown. Calculating values for use within the rearranged formula.
62	5	Calculating unknowns. Volume unknown. Knowledge of units

Table 2 shows that students are learning constraints relevant to fundamental domain concepts: selecting values, selecting formulae, defining state properties, performing calculations, and selecting units. The *Learned* count shows how many students have successfully learned the constraint during the session.

Questionnaire

At the end of the session, the participants were asked to fill out a questionnaire regarding their learning experience using Thermo-Tutor. The questionnaire consisted of five questions asking to rate various aspects of the tutor, on a scale of 1 (low/poor) to 5 (high/excellent), and three open ended questions. 18 participants filled out the questionnaire. Table 3 summarizes the results from the five rating questions.

TABLE III. QUESTIONNAIRE RESULTS

Question	Average (SD)
Mental effort required	3.72 (0.96)
Problem difficulty	3.22 (0.8)
Enjoyment	3.33 (0.97)
How would you rate the overall quality of Thermo-Tutor?	3.78 (0.73)
How would you rate the quality of the feedback from Thermo-Tutor?	3.39 (0.98)

TABLE IV. QUESTIONNAIRE COMMENTS

Question	Responses
What did you like about Thermo-Tutor?	Diagram It helped in visualising the problem Equations provided Revision Helped choosing correct equations Right idea, should have been done ages ago It told me when I made a mistake More fun than on paper
What changes would you like to see in Thermo-Tutor?	Better colors. Yellow and green hard to see Better error indications Instructions on how to use it A bit more user friendly Sort out the rounding issues Fix small bugs
Any other comments?	I hope this is available for exam study It took a while to realise how to use it Great study tool

These results are generally positive, although not outstanding. Mental effort required is quite high, although this is expected when asking students to use a new piece of software. The participants found the problem about the right level of complexity, as well as the quality of Thermo-Tutor and the feedback. The answers to the open ended questions were more encouraging. The responses in Table 4 show that students seemed to like the tutor, even if it was perhaps a bit difficult to use. This is promising, as with a small amount of additional work these small issues can be fixed. The students like the concept and general structure of Thermo-Tutor and it seems very relevant to their course of study.

V. CONCLUSIONS AND FUTURE WORK

This paper describes Thermo-Tutor, an intelligent tutoring system that teaches Thermodynamics, and is aimed at university level students with some amount of prior knowledge. Thermo-Tutor provides an environment where students can practice drawing and working with thermodynamic cycles in closed systems.

We presented the tutor’s interface and the problem-solving process it supports. The interface was designed to be similar to how a student would solve a problem on paper, but is able to provide much more assistance to the student. Students receive immediate feedback, and can progress through the material at their own pace. We see Thermo-Tutor as potentially very useful for students learning thermodynamics, an opinion backed up by the responses to the questionnaire.

Evaluation results show that the ITS is effective at teaching thermodynamic cycles: after using Thermo-Tutor for less than an hour, students were learning constraints representing fundamental aspects of the domain. The ITS appeared to have a relatively steep learning curve, even for students with prior knowledge; *“It took a while to realise how to use it”*. This did not deter students from finding the additional value in it compared to their current methods of study; *“I hope this is available for exam study”*. It is expected that as students become familiar with the interface they will become more efficient in learning.

A few improvements to the tutor remain to be done; however we believe we have developed a successful tutor in which students learn Thermodynamics concepts related to closed cycles. Thermo-Tutor demonstrates the power and flexibility of the constraint-based approach in a complex engineering domain.

We plan to evaluate Thermo-Tutor at the University of Canterbury with more students, using the ITS for a longer period of time. Pending further positive evaluation results, Thermo-Tutor could be deployed at other learning institutes where Thermodynamics is taught.

REFERENCES

[1] Bloom, B.S. “The 2 Sigma Problem: The search for Methods of Group Instruction as Effective as One-to-One Tutoring,” Educational Researcher, vol. 13, no. 6, pp. 4–16, 1984.

[2] Anderson, J. R., Corbett, A. T., Koedinger, K. R., Pelletier, R. “Cognitive Tutors: Lessons Learned’. J. Learning Sciences, vol. 4, no. 2, pp. 167-207, 1995.

[3] Mitrovic, A., Martin, B., Suraweera, P. “Intelligent Tutors for All: Constraint-Based Approach”. IEEE Intelligent Systems, vol. 22, no. 4, pp. 38-45, 2007.

[4] Woolf, B.P. Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning, Morgan Kaufmann, 2008.

[5] Mitrovic, A., Suraweera, P., Martin, B., Weerasinghe, A. “DB-Suite: Experiences with Three Intelligent, Web-based Database Tutors”. Journal of Interactive Learning Research, vol. 15, pp. 409-432, 2004.

[6] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R.., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M., “The Andes Physics Tutoring System: Lessons Learned.” Int. J. Artificial Intelligence in Education, vol. 15, pp. 147-204, 2005.

[7] Koedinger, K.R., Anderson, J. R., Hadley, W. H., Mark, M. A. “Intelligent Tutoring Goes to School in the Big City”. Int. J. Artificial Intelligence in Education, vol. 8, pp. 30-43, 1997.

[8] Murray, “An Overview of Intelligent Tutoring System Authoring Tools: Updated Analysis of the State of the Art”. Authoring Tools for Advanced Technology Learning Environments., pp. 491-545, 2003.

[9] Mitrovic, A., Ohlsson, S. “Evaluation of a Constraint-Based Tutor for a Database Language”. Int. J. Artificial Intelligence in Education, vol. 10, no 3-4, pp. 238-256, 1999.

[10] Suraweera, P., Mitrovic, A. “An Intelligent Tutoring System for Entity Relationship Modelling”. Int. J. Artificial Intelligence in Education, vol. 14, pp. 375-417, 2004.

[11] Weerasinghe, A., Mitrovic, A. “Facilitating Deep Learning Through Self-Explanation in an Open-Ended Domain”. Knowledge-based and Intelligent Engineering Systems, IOS Press, vol. 10, no. 1, pp. 3-19, 2006.

[12] Zakharov, K., Mitrovic, A., Ohlsson, S. “Feedback Micro-Engineering in EER-Tutor”. Proc. Artificial Intelligence in Education, IOS Press, pp. 718-725, 2005.

[13] Baghaei, N., Mitrovic, A., Irwin, W. “Supporting Collaborative Learning and Problem-Solving in a Constraint-based CSCL Environment for UML Class Diagrams”. Computer-Supported Collaborative Learning, vol. 2, no. 2-3, pp. 159-190, 2007.

[14] Baghaei, N., Mitrovic, A. “From Modelling Domain Knowledge to Metacognitive Skills: Extending a Constraint-Based Tutoring System to Support Collaboration. Proc. User Modeling 2007, pp. 217-227, 2007.

[15] Holland, J., Mitrovic, A., Martin, B. “A Constraint-Based Tutor for Java”, In: Kong, S.C, Ogata, H., Arnseth, H.C., Chan, C.K. K., Hirashima, T., Klett, F., Lee, J.H.M., Liu, C.C., Looi, C. K., Milrad, M., Mitrovic, A., Nakabayashi, K., Wong, S.L., Yang, S.J.H (eds) Proc. 17th Int. Con. Computers in Education, Hong Kong: Asia-Pacific society for computers in education, pp. 142-146, 2009.

[16] Mayo, M., Mitrovic, A. “Optimising ITS Behaviour with Bayesian Networks and Decision Theory”. Int. J. Artificial Intelligence in Education, vol. 12, no. 2, pp. 124-153, 2001.

[17] Martin, B., Mitrovic, A. “Domain Modeling: Art or Science?” Proc. Artificial Intelligence in Education 2003, IOS Press, pp. 183-190, 2003.

[18] Mitrovic, A. “The Effect of Explaining on Learning: A Case Study with a Data Normalization Tutor”. Proc. Artificial Intelligence in Education 2005, IOS Press, pp. 499-506, 2005.

[19] Milik, N., Marshall, M., Mitrovic, A. “Responding to Free-Form Student Questions in ERM-Tutor”. Intelligent Tutoring Systems 2006, pp. 707-709, 2006.

[20] Mitrovic, A., McGuigan, N., Martin, B., Suraweera, P., Milik, N., Holland, J. Authoring Constraint-Based Tutors in ASPIRE: A Case Study of a Capital Investment Tutor”. Proc. ED-MEDIA 2008, Vienna, pp. 4607-4616, 2008.

[21] Mitrovic, A., Koedinger, K. R., and Martin, B. “A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modelling”. Proc. User Modeling 2003, Springer-Verlag, pp. 313-322, 2003.

[22] Ohlsson, S. “Constraint-Based Student Modeling”. Int. J. Artificial Intelligence and Education, vol. 3, no. 4, pp. 429-447, 1992

[23] Ohlsson, S., Mitrovic, A. “Fidelity and Efficiency of Knowledge Representations for Intelligent Tutoring Systems”. Technology, Instruction, Cognition and Learning, vol. 5, no 2, pp. 101-132, 2007.

[24] Suraweera, P., Mitrovic, A., and Martin, B., "Widening the Knowledge Acquisition Bottleneck for Constraint-based Tutors," IJAIED, vol. 20, 2010.

[25] Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., and McGuigan, N. ASPIRE: An Authoring System and Deployment Environment for Constraint-Based Tutors. Int. J. Artificial Intelligence in Education, vol. 19, no. 2, pp. 155-188, 2009.