Making Large Class Teaching More Adaptive With the Logic-ITA

Kalina Yacef

School of Information Techonologies
University of Sydney
F09 building, NSW 2006, Australia.

kalina@it.usyd.edu.au

Abstract

Adaptive teaching does not usually rhyme with large number of students in class. Given the small amount of face to face and the disproportionate ratio, teachers have to use their intuition and experience to fine tune teaching strategies and contents. We report in this paper on our experience of using an intelligent teaching assistant system in a large Computer Science undergraduate class. Whilst helping students practice with tailored and immediate feedback, it also captures data so that the teaching staff can be informed in real time of the students' problems and progress. We show how this information is used to adapt the teaching to the current class.

1 Introduction

Large number of students in class is a notorious difficult problem. Firstly, the disproportioned ratio between lecturer and students means that teachers cannot afford to spend time with each individual student. Secondly, the student population is usually very heterogenous in terms of abilities, background knowledge and motivation. This means that adaptation of the teaching material to individual needs is not possible.

Teachers are confronted to a rough dilemma: should lectures be catered for the top, the bottom or the average of the student population? This is further complicated by the fact that they do not really know what these categories are at the time they need to make a decision. It is easier in hindsight to reflect on student performance and assess those, but at the time they need to deliver lectures and tailor the teaching material, they have not much idea.

In some well-defined domains, students can benefit from using an intelligent tutoring system. Some of these systems are successfully deployed in schools to help with maths (Koedinger and Anderson 1997) or with reading skills (Beck *et al.* 2003) for example, and show a significant improvement on student learning. Whilst their educational benefit is still sometime disputed, we believe that they are useful tools when (i) their role is clearly and narrowly defined by the teacher and (ii) their purpose is to compensate the lack of one-to-one interactions with the human teacher and/or to add new forms of communications that are particular to computing (such as interactive videos, games, etc..). There is no substitute to a good human teacher however it is not possible to have one teacher for every student. But we can use computing technology to empower teachers and help them teach better and more efficiently.

We have implemented a system, the Logic-ITA, which is used as an aid for teaching logic formal proofs. On one hand, it provides students with an environment "intelligent" enough to check their exercises and give them a contextualised and immediate feedback. On another hand, it records each student's activity and mistakes and collates this information for the teacher, who can then be much better informed on their classes' progress. This paper describes our experience in using such a tool in a large class.

2 What is the Logic-ITA?

The Logic-ITA is an experiment of how AIED (Artificial Intelligence in Education) technologies and principles could be coupled with a teacher tool that would inform the lecturer in real time about the students' progress and problems. ITA stands for Intelligent Teaching Assistant. First we built the Logic Tutor, a web-based practice tool with some "intelligent" features such as personalised feedback and exercises to help students practice on formal proofs of logic (Abraham *et al.* 2001). Then we added useful tools for the teacher: the LT-Configurator, a tool with authoring facilities to configure the teaching settings and the LT-Analyser, a tool that collects all the information about each student's interactions with the Logic Tutor (stored in the student's user model), collates them in a database that the teacher can query (Lesta and Yacef 2002).

The Logic-ITA presents some analogies with a human tutor (in the australian sense, i.e. a person who works with the lecturer and manages a problem-solving session with a group of usually 20 students). The Logic Tutor, dedicated to the students, has enough knowledge both about the domain of logic and basic teaching to "monitor" that students conduct their proofs in a correct way and give them immediate feedback and hints. The LT-Analyser, dedicated to the teacher, provides feedback to the teacher about how the students are going. This feedback is for the moment manually queried by the teacher and can be done at the class level, a group level, or an individual level. For example, the teacher can query the most frequent mistakes made by the class, or by a particular tutorial class, or by the students who attempted exercise X, or by student Y. Naturally the analogy with a human tutor stops here: the Logic-ITA does not capture any information such as the student motivation and any visual or audible clues such as the sighing, the head scratching, the ease with which students "see" the proof and so on. On another hand, a human tutor cannot, like the Logic-ITA, be watching each student, keep track of every exercise made, every concept seen, every mistake made for each student.

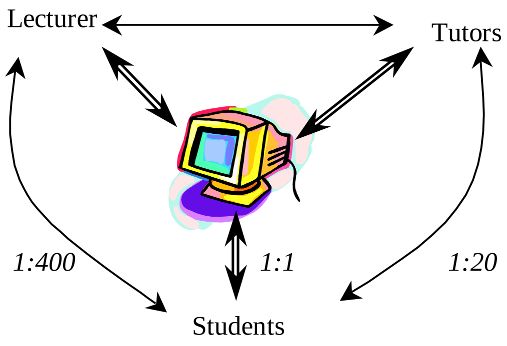

Figure 1 - Schematic representation of teaching and learning interactions

Figure 1 shows the various parties involved and the stream of their interactions. Single line arrows refer to interactions existing in a traditional undergraduate teaching setting. The lecturer delivers lectures to a large class, with for example 400 students. Tutors facilitate small problem solving sessions with around one tutor for 20 students. Finally, lecturer and tutors also have interactions, usually briefing and debriefing sessions. The new interactions brought by the Logic-ITA is shown using double line arrows. Firstly, students use the system on a one-on-one basis, from home or from the university. Then lecturer and tutors can access the information stored in the student models. As can be seen, the Logic-ITA does not *replace* any existing interactions. It *adds* new ones.

Content of student models.

Without going into too much detail about the technical side of the Logic-ITA, it is useful to understand the nature of the data that it stores about each individual student. We should stress that the Logic-ITA is not a learning management system but an intelligent tutoring system. It gives exercises to students (or they can enter their own) and it checks and assists the student in reaching a correct answer, giving them immediate and appropriate feedback and hints. An exercise is given by a list, possibly empty, of premises and a conclusion. Premises and conclusion are well-formed formulae of propositional logic. The aim of the exercise is to derive formally the conclusion from the list of given premises, justifying each step with a logic rule, the lines of references and the set of premises the step relies on. The mistakes that can be made at each step vary in nature: there might be syntactical errors (eg a bracket is missing), misuse of a rule (eg the formula derived from using rule X does not give the formula entered by the student), a missing or incorrect justification and so on. In case of a mistake, the student is given immediate feedback, possibly a relevant hint (such as "*try and use the Modus Tollens rule instead of Modus Ponens*" when the system detects that this action would cancel the mistake) and tries again.

For each exercise attempted, the Logic Tutor stores in the relevant user model: the exercise, with all its steps, mistakes made alone the way, feedback received). Incidentally, the student can browse through this information at any time for reflection purposes.

The student models then consist of all the exercises attempted by the relevant user, the path followed in reaching the answer including the erroneous steps, the mistakes made and feedback received, the performance reached for each exercise, the student's current level in the curriculum, the number of times they logged in, to cite the main attributes.

3 The Logic-ITA in the Languages and Logic course

Three years ago we introduced the Logic-ITA in our second year undergraduate class entitled "Languages and Logic". In 2000, 431 students sat the final exam. In 2001 there were 390 students. As of 2002 the number of students dropped to 245 and in 2003 to 144 (the drop in numbers is due to a restructure of our curriculum. More 2nd year courses were introduced in 2002, giving more choices to our students, and it ceased to be a core subject as of 2003 for several streams of students).

Propositional logic is taught over 2 and half weeks. Concepts are explained in lectures: laws of equivalence, rules of inference, tautologies and the process of conducting formal proofs, first using the Natural Deduction system and then using Resolution. Lectures are made in an interactive way, using lots of examples and exercises done collectively. It is one the most theoretical undergraduate course offered in the school and the fact is many of our computer science students struggle with theory. Hence practice is not only useful but also fundamental for them to grasp the concepts. The course includes programming assignments and practical exercises. Formal proofs in particular require a lot of practice to reach a stage where their philosophy is well understood. Given the small timeframe, average students can roughly assimilate the concept and understand when they see the solution, or parts of it. However, to be able to do a formal proof by themselves from the start, students need to do them a certain number of times until the concept is mastered. This time length obviously varies between subjects. In the past, we observed that the face-to-face contact time was not sufficient to provide students with good and sufficient feedback. Only self-disciplined students sufficiently prepared their homework for their tutorial and received the feedback they needed. As of 2001, the Logic-ITA was particularly useful to the students as it allowed them to practice on their own and to receive immediate feedback. It can be assumed that the same proportion of students prepared themselves for the tutorials, but those who did not still had the chance to do so before the exam. Analysis made after the exam was marked showed a significant increase in the logic question marks for students finishing more than 6 exercises.

4 How the Logic-ITA was used

The Logic-ITA comprises three tools. One, the actual tutoring tool, is destined to students (Logic Tutor) and the two others to the teacher: one allows him or her to set up the curriculum, difficulty levels and progression of students through these levels, the other provides information to the teacher about how the class is going. We will describe how each component was used.

4.1 System set up

In effect, the curriculum we designed recommended that students first practice on short exercises involving laws of equivalence or simple rules of inference (ie not Indirect Proof (IP) or Conditional Proof (CP)), without yet mixing the two categories; then on a mix of these two sets; and finally on any rule or law including IP and CP. Throughout these stages, a pacing of the difficulty is made by setting limits on the length of the exercises (ie the number of lines in a possible solution) and the number of rules they involve. The longer the proof needs to be, the more vision is required.

The reason behind the initial separation of laws and rules is that they are of different types and uses. The former come from logical equivalences, therefore can be used to substitute any subpart of a wff (for example, in the wff ((A&B)→C), one can derive ((B&A)→C) using the law of commutativity.). Rules of inference come from logical inferences. They should only be applied to a whole formula (for example, the Simplification rule says that one can derive A from (A&B). But, in the wff ((A&B)→C), it is not logically valid to derive (A→C)). Hence separating the two allows time for students to focus on one same type of rules in one exercise.

Exercises are then indexed using a combination of type of rules they involve, their numbers as well as the length of the proof. For example, level 2 involves very easy, short exercises using laws of equivalence or simple rules of inference and level 3 exercises may use the same type of rules but in longer and more complex exercises. We set up 5 levels in total.

Students then progress through these levels after "completion" of a "certain" number of exercises at the current level. The terms *completion* and *certain* are defined by the teacher. For example, to progress from level 1 to level 2, students had to complete (even if they made mistakes along the way) 3 exercises. But to progress from level 4 to level 5 (the upper level), they also had to complete 3 exercises but with the constraints of using *Indirect proof* or *Conditional proof*.

These levels are useful for two main reasons: one is that they allow students to be recommended exercises fitting their level and needs (although students always have the freedom to choose or enter any exercise they want). The other is that it provides a mechanism to classify exercises and student levels in a way that is meaningful to the teacher and used in the LT-Analyser.

4.2 Logic Tutor

A live demonstration of the Logic Tutor, i.e. the student's side, was given in lecture. Students, due to their knowledge in computer science, did not experience any usability problem with the tool. We asked students to hand in their homework using the Logic Tutor. This had two beneficial effects:

- firstly, students had to use the tool, at least once (one individual assignment and two small sets of weekly exercises which could be done in pairs). This enforcement was mainly to force the student to use the tool at least once. As marketers know well, the first time use is the biggest barrier. Once a person is logged in, he or she is more likely to do more exercises.
- secondly, marking time was dramatically reduced for these exercises compared to previous years. Formal proofs are very tedious to check for a human. There is not one single solution, so the validity of each step must be thoroughly checked by the marker. The Logic Tutor checking the validity of each step on the fly, the marker only needs to check whether the solution was reached or not and rely on the Logic Tutor for the validity checking of all the steps. This allowed us to add a very small marked assignment on the Logic Tutor in 2002 without worrying about marking budget.

Some students used the tool only the strict number of times they had to, but many played with the tool and fully used its functionalities.

4.3 Analysing student data with LT-Analyser

During the two weeks and then again in the last week of the semester, taken by revisions, we queried the database containing the latest update of student models to find out simple information such as:

- the logic rules that were the most commonly misused,
- the number and type of mistakes made globally (across all rules) and for each rule, sorted by descending number of occurrences,
- the performance on a particular exercise,
- the number of times students logged in on average.

Additional queries were made after the final exam was marked. We investigated, for example, the correlation between the usage of the Logic Tutor (frequency of usage, level reached, number of exercises attempted and their performance) with the mark obtained by the student in the logic formal proof question at the final exam.

5 Making large class teaching more adaptive

Re-focus the content of lectures

The results of the LT-Analyser queries was primarily used to focus the contents of the revision lecture on logic and to draw the attention of the students on the concepts they seemed to be most struggling with. In 2002 for example, the manipulation of premises in *Indirect* and *Conditional Proofs* generated a lot of mistakes. In lectures, we re-explained these concepts, with relevant and concrete examples of mistakes made by the students in the past weeks. This occurred in the context of exercises in where students had to participate. The students' response (an even greater attention than usual!) was a good indicator of the accuracy of the focus.

Increased opportunities for student to practise with feedback

The Logic Tutor is knowledgeable enough to allow students to practise as much as they need. The feedback they receive is often as enlightening as that of a human tutor. The survey conducted with students who used to the Logic Tutor for the course revealed that the tool was considered useful (74%) to learn formal proofs.

This in turn led to a **significant increase in student performance**. We compared the results of the 2001 class, who used a rather constraining version of the Logic Tutor (students had to use it on Unix undergraduate machines) and the 2002 class, who used the web-based Logic Tutor. We used, as our control group, the 2000 class who did not use the tool at all and compared the results obtained in homework and exam question. We gave similar homework in 2000, 2001 and 2002 as well as a similar exam question on formal proofs. The homework consisted of logic proofs, given the premises and the conclusion. The exam question was a logic proof to complete with some parts of the proof provided, and the student had to fill in the missing parts. In 2000, all assessments were paper-pen. In 2001 and 2002, the homework had to be done through the Logic Tutor whilst of course the exam question remained paper pen. Hence the exam question is our most important and unbiased indicator, as homework answers were "filtered" by the Logic Tutor, whereas the exam question was not.

The homework average mark in 2002, not surprisingly, is close to the maximum mark, strongly due to the fact that the Logic Tutor takes care of the mistakes before the submission. The interesting result lies with the exam question average result, which steadily grew up over the 2 years to reach an effect size in 2002 of 0.9 standard deviation, or sigma[2], meaning that the average mark obtained by the students in 2002 increased by 0.9 times the standard deviation of the control group (year 2000). This means that around 65% of students in 2002 exceeded the levels of achievement attained by only 10% of students in 2000. This indicates to us that students were better prepared. At least the Logic-ITA gave students the opportunity to practice more often with feedback and to be more familiarised with the process prior to sitting the exam. This is also confirmed by the correlation we found between the number of exercises practised, the level reached in the Logic Tutor and the exam mark. ANOVA tests confirmed a statistical difference with the control group in regards to the logic exam question ($F=69.9$, $p<0.001$), whereas the results obtained at another question (on Regular languages and finite automata) could not be differentiated between 2002 and 2000. This was important to verify this as the reduction in size of the class (from 431 in year 2000 to 245 in year 2002) also meant that 2002 students were more likely to be motivated for the course than 2000 students, hence potentially affecting the results. More details of the statistical analysis can be found at (Yacef 2003). It is also important to note that the evaluation reflects on the whole usage of the Logic-ITA, without highlighting which proportion is due to the Logic Tutor itself, and which proportion is due to the monitoring process.

Better awareness of ingrained misconceptions

During the time the students used the tool, a handful of students complained that the system was not working properly. The reason was that the system would not let them enter a particular step, which was in fact invalid. Most students, most of the time, read the feedback given by the system and understood their mistake. But some students kept ignoring the feedback. From a computer design point of view, this obviously raised the question that the interface was not appropriate for everyone and could be improved. But more interestingly, from a teaching point of view, it highlighted the fact that students can have deep misconceptions and can be totally oblivious of them, even when they are confronted with evidence of the contrary. Thanks to the Logic Tutor and to the fact that they had to hand in homework using the system, these cases were brought up to a tutor or to the lecturer (either because the student would "complain" or because his/her pen rectification would stand out on the print out). A remedial discussion then took place, with the lecturer being aware of this ingrained misconception.

Guidelines to improve following courses

Subsequent analysis was also made after the end of the semester, using other techniques to extract more information. In particular we used association rule to find out the mistakes that often occur together (Merceron and Yacef 2003). These exposed that the concept itself of formal proofs (especially its "formal" side, ie that each column must contain specific information and the way that information is calculated) caused difficulties. We also looked at the correlation between the use of the Logic Tutor and the exam results. There is a direct link between the number of exercises, the level reached by the students and the result at the exam question on formal proofs.

All these analysis of student data is also extremely useful for improving subsequent teaching semesters. The previous years' data suggested some changes for 2003.

[2] Whilst Bloom reports studies showing that human tutoring can yield an effect size of 2 sigmas (Bloom 1984), Kulik's meta-analytic study (Kulik 1994) of computer assisted instruction reports that, over a wide range of instructional areas and student levels, a gain of approximately 0.35 sigma is achieved. Koedinger, Anderson and colleagues have also shown in various evaluation studies that their cognitive tutors reached a 1 sigma effect (Koedinger and Anderson 1993; Koedinger and Anderson 1997).

a) The concept itself of formal proofs, which seems to be often misunderstood, will be introduced earlier. Also, the fundamental difference between laws of equivalence and rules of inference will be much more emphasized. In previous years, students saw laws of equivalence, simple rules of inference, and then the two complex rules of inference (Conditional and Indirect proofs) before attacking the concept of formal proofs. In 2003, the concept of formal proofs was introduced just after the laws of equivalence, a second time after the rules of inference, a third time after Conditional and Indirect proofs, and again a fourth time for proving tautologies.

b) Students were given *counter-examples* based on common mistakes of previous years, not only in lectures but also in tutorial exercises, to engage them actively in finding the mistakes. This follows the approach advocated by (Fekete 2003). They were given proof fragments with invalid steps (students were aware that they are invalid) and the aim of the exercise will consist in finding the mistakes and explaining why. In some cases where it was appropriate, they were asked to amend the step to make it valid.

6 Conclusion

We are concerned with finding ways to alleviate the lack of teaching adaptation in large classes. The Logic ITA is an example of how technology can be used to complement the teacher. It acts as an additional intermediary between teacher and students: on one hand it provides students with an environment to practice formal proofs and on another hand it allows teachers to monitor how the class is progressing. We conducted an experiment over the past 3 years of using this tool in an undergraduate course. Since the introduction of the tool students were able to practice formal proofs and receive immediate feedback, and the teacher was able to tailor the content of lectures to the current class and also to redesign or fine-tune the following teaching semesters.

The tool as it stands can only be used for Logic. However, the concept of class monitoring can be applied to any domain where a tutoring system can be built. The minimum that the system would need to provide is a record all the data for each student with their mistakes as well as correct answers.

7 References

Abraham, D., L. Crawford, L. Lesta, A. Merceron and K. Yacef (2001). "The Logic Tutor: a Multimedia Presentation." *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning* 3(2).

Beck, J. E., Mostow J., A. Cuneo and J. Bey (2003). "Can automated questioning help children's reading comprehension?" *Proceedings of International Conference on Artificial Intelligence in Education (AIED2003)* (eds) Sydney, Australia,

Bloom, B. S. (1984). "The 2 Sigma problem: The search for methods of group instruction as effective as one-to-one tutoring." *Educational Researcher* 13(6): 4-16.

Fekete, A. (2003). "Using Counter-Examples in the Data Structures Course". *Proceedings of Australasian Computing Education Conference (ACE2003)* T. Greening and R. Lister (eds) Adelaide, Australia,

Koedinger, K. R. and J. R. Anderson (1993). "Effective use of intelligent software in high school math classrooms". *Proceedings of Conference on Artificial Intelligence in Education (AIED'93)* (eds) Charlottesville: VA,

Koedinger, K. R. and J. R. Anderson (1997). "Intelligent Tutoring Goes to School in the Big City." *International Journal of Artificial Intelligence in Education* 8: 30-43.

Kulik, J. A. (1994). "Meta-analytic studies of findings on computer-based instruction". *Technology assessment in education and training*, E. B. a. H. O'Neil (eds), Hillsdale, NJ, Lawrence Erlbaum Associates.

Lesta, L. and K. Yacef (2002). "An Intelligent Teaching-Assistant System for Logic". *Proceedings of International Conference on Intelligent Tutoring Systems (ITS'02)* S. Cerri and F. Paraguo (eds) Biarritz, France, Springer-Verlag,

Merceron, A. and K. Yacef (2003). "A Web-based Tutoring Tool with Mining Facilities to Improve Learning and Teaching". *Proceedings of 11th International Conference on Artificial Intelligence in Education.* F. Verdejo and U. Hoppe (eds) Sydney, IOS Press,

Yacef, K. (2003). Experiment and Evaluation Results of the Logic-ITA. *Technical report 542*, School of Information Technologies, University of Sydney.